



Open Source Used In AppDynamics_Cloud_Metric_Reader _Service 22.11.0-749

Cisco Systems, Inc.

www.cisco.com

Cisco has more than 200 offices worldwide.
Addresses, phone numbers, and fax numbers
are listed on the Cisco website at
www.cisco.com/go/offices.

Text Part Number: 78EE117C99-1473455589

This document contains licenses and notices for open source software used in this product. With respect to the free/open source software listed in this document, if you have any questions or wish to receive a copy of any source code to which you may be entitled under the applicable free/open source license(s) (such as the GNU Lesser/General Public License), please submit this [form](#).

In your requests please include the following reference number **78EE117C99-1473455589**

Contents

1.1 reflections 0.9.10

1.1.1 Available under license

1.2 hamcrest v2.2

1.2.1 Available under license

1.3 findbugs-annotations 3.0.1

1.3.1 Available under license

1.4 snake-yaml 1.33

1.4.1 Available under license

1.5 scala-collection-compat 2.6.0

1.5.1 Available under license

1.6 bom 4.1.82.Final

1.6.1 Available under license

1.7 micronaut-servlet 3.3.1

1.7.1 Available under license

1.8 java-architecture-for-xml-binding 2.3.3

1.8.1 Available under license

1.9 micronaut-kubernetes 3.4.0

1.9.1 Available under license

1.10 checker-qual 3.12.0

1.10.1 Available under license

1.11 swagger-models 2.2.3

1.11.1 Available under license

1.12 apache-httpmime 4.5.13

1.12.1 Available under license

1.13 commons-logging 1.2

1.13.1 Available under license

1.14 okio 2.8.0

1.14.1 Available under license

1.15 javabeans-activation-framework-api 1.2.2

1.15.1 Available under license

1.16 logback-core 1.2.10

1.16.1 Available under license

1.17 apache-kafka 3.3.1

1.17.1 Available under license

1.18 feign-apache-httpclient 10.12

1.18.1 Available under license

1.19 jackson-databind 2.14.0

1.19.1 Available under license

1.20 fabric8-:::kubernetes-model-::-discovery 4.13.3

1.20.1 Available under license

1.21 junit-5-bill-of-materials 5.9.1

1.21.1 Available under license

1.22 jakarta-dependency-injection 2.0.1

1.22.1 Available under license

1.23 jackson-jaxrs 2.14.0

1.23.1 Available under license

1.24 apache-commons-lang 3.9

1.24.1 Available under license

1.25 guava-internalfuturefailureaccess-and-internalfutures 1.0.1

1.25.1 Available under license

1.26 fabric8-:::kubernetes-model-::-settings 4.13.3

1.26.1 Available under license

1.27 micronaut-security 3.8.0

1.27.1 Available under license

1.28 opentelemetry-java---io.opentelemetry:opentelemetry-sdk-metrics 1.9.1-alpha

1.28.1 Available under license

1.29 jetbrains-kotlin-kotlin-stdlib-jdk8 1.6.21

1.29.1 Available under license

1.30 stringtemplate4 4.3

1.30.1 Available under license

1.31 commons-csv commons-csv-1.8

1.31.1 Available under license

1.32 jetty-java-based-http-1-x-http-2-servlet-websocket-server 9.4.49.v20220914

1.32.1 Available under license

1.33 jackson-dataformat-yaml 2.14.0

- 1.33.1 Available under license
- 1.34 apiguardian-apiguardian-api 1.1.0**
 - 1.34.1 Available under license
- 1.35 okhttp 4.9.3**
 - 1.35.1 Available under license
- 1.36 google-gson 2.9.0**
 - 1.36.1 Available under license
- 1.37 junit-jupiter-junit-jupiter-engine 5.7.1**
 - 1.37.1 Available under license
- 1.38 prometheus-java-simpleclient-common 0.15.0**
 - 1.38.1 Available under license
- 1.39 opentelemetry-instrumentation-for-java 1.15.0-alpha**
 - 1.39.1 Available under license
- 1.40 jcommander-library 1.72**
 - 1.40.1 Available under license
- 1.41 curator-recipes 5.2.1**
 - 1.41.1 Available under license
- 1.42 micronaut-object-storage 1.0.0**
 - 1.42.1 Available under license
- 1.43 jvm-integration-for-metrics 4.0.5**
 - 1.43.1 Available under license
- 1.44 metrics---dropwizard v4.0.5**
 - 1.44.1 Available under license
- 1.45 protobuf-java 3.19.4**
 - 1.45.1 Available under license
- 1.46 micronaut-problem-json 2.5.1**
 - 1.46.1 Available under license
- 1.47 cloudevents---core 2.2.0**
 - 1.47.1 Available under license
- 1.48 micronaut-kafka 4.4.0**
 - 1.48.1 Available under license
- 1.49 fabric8--:-kubernetes-model--:-networking 4.13.3**
 - 1.49.1 Available under license
- 1.50 guava-listenablefuture-only 9999.0-empty-to-avoid-conflict-with-guava**
 - 1.50.1 Available under license
- 1.51 jackson-module-afterburner 2.14.0**
 - 1.51.1 Available under license
- 1.52 open-telemetry/opentelemetry-java 1.9.1**
 - 1.52.1 Available under license

- 1.53 junit-jupiter-junit-jupiter-engine 5.9.1**
 - 1.53.1 Available under license
- 1.54 byte-buddy byte-buddy-1.9.10**
 - 1.54.1 Available under license
- 1.55 coherence-bom 22.06.1**
 - 1.55.1 Available under license
- 1.56 feign-slf4j 10.12**
 - 1.56.1 Available under license
- 1.57 t-digest 3.2**
 - 1.57.1 Available under license
- 1.58 junit-platform-junit-platform-engine 1.9.1**
 - 1.58.1 Available under license
- 1.59 joda-time 2.10.5**
 - 1.59.1 Available under license
- 1.60 unboundid-scim2-sdk-common 2.3.6**
 - 1.60.1 Available under license
- 1.61 jul-to-slf4j-bridge 1.7.32**
 - 1.61.1 Available under license
- 1.62 feign-core 10.12**
 - 1.62.1 Available under license
- 1.63 micronaut-azure 3.5.0**
 - 1.63.1 Available under license
- 1.64 apache-commons-collections 3.2.2**
 - 1.64.1 Available under license
- 1.65 log4j-compatibility-api 2.17.1**
 - 1.65.1 Available under license
- 1.66 metrics-integration-for-apache-httpasyncclient 4.0.5**
 - 1.66.1 Available under license
- 1.67 zookeeper 3.8.0**
 - 1.67.1 Available under license
- 1.68 metrics-integration-with-jmx 4.0.5**
 - 1.68.1 Available under license
- 1.69 jose4j 0.6.3**
 - 1.69.1 Available under license
- 1.70 ktor-bom 1.6.8**
 - 1.70.1 Available under license
- 1.71 jetty-java-based-http-1-x-http-2-servlet-websocket-server 11.0.12**
 - 1.71.1 Available under license
- 1.72 fabric8-::-kubernetes-model-::-scheduling 4.13.3**

- 1.72.1 Available under license
- 1.73 feign-jackson 10.12**
 - 1.73.1 Available under license
- 1.74 prometheus-java-span-context-supplier-common 0.15.0**
 - 1.74.1 Available under license
- 1.75 feign-okhttp 10.12**
 - 1.75.1 Available under license
- 1.76 jakarta-expression-language-3.0-api 5.0.0**
 - 1.76.1 Available under license
- 1.77 latencyutils 2.0.3**
 - 1.77.1 Available under license
- 1.78 micronaut-test 3.1.1**
 - 1.78.1 Available under license
- 1.79 curator-client 5.2.1**
 - 1.79.1 Available under license
- 1.80 google-guice 4.2.3**
 - 1.80.1 Available under license
- 1.81 cloudevents---kafka-transport-binding 2.2.0**
 - 1.81.1 Available under license
- 1.82 micronaut-reactor 2.4.1**
 - 1.82.1 Available under license
- 1.83 elasticsearch-kubernetes-cloud-plugin 4.13.3**
 - 1.83.1 Available under license
- 1.84 netty-tomcatnative-openssl-dynamic 2.0.54.Final**
 - 1.84.1 Available under license
- 1.85 apache-yetus-audience-annotations 0.12.0**
 - 1.85.1 Available under license
- 1.86 micronaut-for-spring 4.3.1**
 - 1.86.1 Available under license
- 1.87 aws-java-sdk::-bill-of-materials 2.17.271**
 - 1.87.1 Available under license
- 1.88 fabric8::-kubernetes-model::-storage-class 4.13.3**
 - 1.88.1 Available under license
- 1.89 kafka-streams-avro-serde 5.5.5**
 - 1.89.1 Available under license
- 1.90 netty-transport-native-unix-common 4.1.84.Final**
 - 1.90.1 Available under license
- 1.91 testng 7.0.0**
 - 1.91.1 Available under license

- 1.92 asm-tree 7.2**
 - 1.92.1 Available under license
- 1.93 apache-kafka 3.3.1**
 - 1.93.1 Available under license
- 1.94 guava 31.0.1-jre**
 - 1.94.1 Available under license
- 1.95 micronaut-rxjava-2 1.3.0**
 - 1.95.1 Available under license
- 1.96 io-grpc-grpc-core 1.47.0**
 - 1.96.1 Available under license
- 1.97 google-guice 4.1.0**
 - 1.97.1 Available under license
- 1.98 commons-compress 1.21**
 - 1.98.1 Available under license
- 1.99 jersey's-jersey 2.34**
 - 1.99.1 Available under license
- 1.100 assertj-fluent-assertions 3.21.0**
 - 1.100.1 Available under license
- 1.101 kafka-protobuf-serializer 5.5.5**
 - 1.101.1 Available under license
- 1.102 apache-commons-cli 1.4**
 - 1.102.1 Available under license
- 1.103 utils 5.5.5**
 - 1.103.1 Available under license
- 1.104 appdynamics-java-agent-api 4.5.13.27526**
 - 1.104.1 Available under license
- 1.105 jackson-xc 2.14.0**
 - 1.105.1 Available under license
- 1.106 jetbrains-annotations 13.0**
 - 1.106.1 Available under license
- 1.107 io-grpc-grpc-context 1.47.0**
 - 1.107.1 Available under license
- 1.108 io-swagger-swagger-annotations 2.2.2**
 - 1.108.1 Available under license
- 1.109 cloudevents---api 2.2.0**
 - 1.109.1 Available under license
- 1.110 slf4j-api-module 1.7.26**
 - 1.110.1 Available under license
- 1.111 asm 9.1**

- 1.111.1 Available under license
- 1.112 metrics-utility-servlets 4.0.5**
 - 1.112.1 Available under license
- 1.113 jackson-datatype-joda 2.14.0**
 - 1.113.1 Available under license
- 1.114 tdigestsketch 0.17.0**
 - 1.114.1 Available under license
- 1.115 feign-jackson 8.18.0**
 - 1.115.1 Available under license
- 1.116 apache-groovy 4.0.1**
 - 1.116.1 Available under license
- 1.117 micronaut-kotlin-integrations 3.2.2**
 - 1.117.1 Available under license
- 1.118 scala 2.13.10**
 - 1.118.1 Available under license
- 1.119 micronaut-grpc 3.3.1**
 - 1.119.1 Available under license
- 1.120 micronaut-data 3.8.1**
 - 1.120.1 Available under license
- 1.121 jackson-jaxrs-base 2.14.0**
 - 1.121.1 Available under license
- 1.122 open-feign-forms-core 3.8.0**
 - 1.122.1 Available under license
- 1.123 wire-schema 3.6.0**
 - 1.123.1 Available under license
- 1.124 swagger-core v2.2.3**
 - 1.124.1 Available under license
- 1.125 reactive-streams v1.0.3**
 - 1.125.1 Available under license
- 1.126 kotlin-libraries-bill-of-materials 1.6.21**
 - 1.126.1 Available under license
- 1.127 rest-assured 5.1.0**
 - 1.127.1 Available under license
- 1.128 jul-to-slf4j-bridge 1.7.26**
 - 1.128.1 Available under license
- 1.129 apache-httpcomponents-core 4.4.11**
 - 1.129.1 Available under license
- 1.130 jackson-annotations 2.14.0**
 - 1.130.1 Available under license

1.131 wire-multiplatform-runtime-(experimental) 3.6.0

1.131.1 Available under license

1.132 json-p 1.0.4

1.132.1 Available under license

1.133 apache-httpcomponents-asyncclient 4.1.4

1.133.1 Available under license

1.134 kafka-schema-serializer 5.5.5

1.134.1 Available under license

1.135 antlr-4-tool 4.8-1

1.135.1 Available under license

1.136 fabric8-::-kubernetes-model-::-batch 4.13.3

1.136.1 Available under license

1.137 apache-groovy 3.0.13

1.137.1 Available under license

1.138 asm-analysis 7.2

1.138.1 Available under license

1.139 micronaut-mqtt 2.3.0

1.139.1 Available under license

1.140 jackson-datatype-jsr310 2.14.0

1.140.1 Available under license

1.141 jopt-simple 5.0.4

1.141.1 Available under license

1.142 feign-core 8.18.0

1.142.1 Available under license

1.143 micronaut 3.7.2

1.143.1 Available under license

1.144 apache-kafka 3.3.1

1.144.1 Available under license

1.145 apache-avro 1.11.1

1.145.1 Available under license

1.146 asm-commons 7.2

1.146.1 Available under license

1.147 micronaut-gcp 4.6.0

1.147.1 Available under license

1.148 bean-validation-api 2.0.1

1.148.1 Available under license

1.149 error_prone_annotations 2.7.1

1.149.1 Available under license

1.150 feign-hystrix 10.12

- 1.150.1 Available under license
- 1.151 javaserver(tm)-faces-2.3-api 3.1.0.SP01**
 - 1.151.1 Available under license
- 1.152 perfmark-perfmark-api 0.25.0**
 - 1.152.1 Available under license
- 1.153 snake-yaml 1.32**
 - 1.153.1 Available under license
- 1.154 prometheus-java-simpleclient 0.15.0**
 - 1.154.1 Available under license
- 1.155 byte-buddy-agent 1.9.10**
 - 1.155.1 Available under license
- 1.156 icu4j 67.1**
 - 1.156.1 Available under license
- 1.157 generex 1.0.2**
 - 1.157.1 Available under license
- 1.158 apache-http-client 4.5.13**
 - 1.158.1 Available under license
- 1.159 jetbrains-kotlin-kotlin-stdlib-jdk7 1.6.21**
 - 1.159.1 Available under license
- 1.160 commons-codec 1.14**
 - 1.160.1 Available under license
- 1.161 fabric8-::-kubernetes-model-::-admission-registration,-authentication-and-authorization 4.13.3**
 - 1.161.1 Available under license
- 1.162 wire-protocol-buffer-runtime 3.6.0**
 - 1.162.1 Available under license
- 1.163 bean-validation-api 2.0.2**
 - 1.163.1 Available under license
- 1.164 apache-commons-lang 2.6**
 - 1.164.1 Available under license
- 1.165 org.apache.groovy:groovy-xml 4.0.1**
 - 1.165.1 Available under license
- 1.166 vavr-match 0.9.2**
 - 1.166.1 Available under license
- 1.167 fabric8-::-kubernetes-model-::-extensions 4.13.3**
 - 1.167.1 Available under license
- 1.168 jsr311-api 1.1.1**
 - 1.168.1 Available under license
- 1.169 antlr 4.8-1**

- 1.169.1 Available under license
- 1.170 jdbc3-bom 3.30.0**
 - 1.170.1 Available under license
- 1.171 micrometer-bom 1.9.4**
 - 1.171.1 Available under license
- 1.172 micronaut-oracle-cloud 2.2.1**
 - 1.172.1 Available under license
- 1.173 kafka-avro-serializer 5.5.5**
 - 1.173.1 Available under license
- 1.174 micronaut-cache 3.5.0**
 - 1.174.1 Available under license
- 1.175 metrics-core 4.2.12**
 - 1.175.1 Available under license
- 1.176 javax-annotation-api 1.3.2**
 - 1.176.1 Available under license
- 1.177 fabric8-::-kubernetes-model-::-apps 4.13.3**
 - 1.177.1 Available under license
- 1.178 caffeine-cache 3.0.3**
 - 1.178.1 Available under license
- 1.179 metrics-integration-with-jmx 4.1.29**
 - 1.179.1 Available under license
- 1.180 openapi/swagger-support 4.5.2**
 - 1.180.1 Available under license
- 1.181 micronaut-views 3.6.0**
 - 1.181.1 Available under license
- 1.182 apache-groovy 4.0.1**
 - 1.182.1 Available under license
- 1.183 fabric8-::-kubernetes-model-::-core 4.13.3**
 - 1.183.1 Available under license
- 1.184 jackson-datatype-guava 2.14.0**
 - 1.184.1 Available under license
- 1.185 json-smart 2.4.7**
 - 1.185.1 Available under license
- 1.186 micronaut-flyway 5.4.1**
 - 1.186.1 Available under license
- 1.187 kotlin-stdlib-common 1.6.21**
 - 1.187.1 Available under license
- 1.188 jackson-module-scala 2.14.0**
 - 1.188.1 Available under license

- 1.189 netty-project 4.1.84.Final**
 - 1.189.1 Available under license
- 1.190 scala-logging_2.12 3.9.4**
 - 1.190.1 Available under license
- 1.191 micronaut-micrometer 4.6.1**
 - 1.191.1 Available under license
- 1.192 jackson-datatype-jdk8 2.14.0**
 - 1.192.1 Available under license
- 1.193 junit-jupiter-junit-jupiter-api 5.7.1**
 - 1.193.1 Available under license
- 1.194 project-lombok 1.18.8**
 - 1.194.1 Available under license
- 1.195 antlr 3.5.2**
 - 1.195.1 Available under license
- 1.196 apache-log4j-api 2.17.1**
 - 1.196.1 Available under license
- 1.197 junit 4.13.1**
 - 1.197.1 Available under license
- 1.198 feign-httpclient 8.18.0**
 - 1.198.1 Available under license
- 1.199 micronaut-liquibase 5.4.1**
 - 1.199.1 Available under license
- 1.200 fabric8-::-kubernetes-model-::-certificates 4.13.3**
 - 1.200.1 Available under license
- 1.201 javatm-ee-6-specification-apis 8.0**
 - 1.201.1 Available under license
- 1.202 opentelemetry 1.9.1**
 - 1.202.1 Available under license
- 1.203 kotlin 1.6.21**
 - 1.203.1 Available under license
- 1.204 j2objc-annotations 1.3**
 - 1.204.1 Available under license
- 1.205 rxjava 2.2.14**
 - 1.205.1 Available under license
- 1.206 jacoco 0.8.5**
 - 1.206.1 Available under license
- 1.207 asm-based-accessors-helper-used-by-json-smart 2.4.7**
 - 1.207.1 Available under license
- 1.208 okio 2.8.0**

- 1.208.1 Available under license
- 1.209 micronaut-serialization 1.3.2**
 - 1.209.1 Available under license
- 1.210 kafka-protobuf-provider 5.5.5**
 - 1.210.1 Available under license
- 1.211 io-grpc-grpc-api 1.47.0**
 - 1.211.1 Available under license
- 1.212 automation 1.11-8**
 - 1.212.1 Available under license
- 1.213 objenesis 2.6**
 - 1.213.1 Available under license
- 1.214 javabeans-activation-framework-api 1.2.1**
 - 1.214.1 Available under license
- 1.215 micronaut-jax-rs 3.4.0**
 - 1.215.1 Available under license
- 1.216 commons-io 2.11.0**
 - 1.216.1 Available under license
- 1.217 fabric8-::-kubernetes-model-::-events 4.13.3**
 - 1.217.1 Available under license
- 1.218 findbugs-jsr305 3.0.2**
 - 1.218.1 Available under license
- 1.219 spring-framework 5.3.23**
 - 1.219.1 Available under license
- 1.220 micronaut-groovy 3.3.1**
 - 1.220.1 Available under license
- 1.221 json-path 5.2.0**
 - 1.221.1 Available under license
- 1.222 azure-java-sdk-bom-(bill-of-materials) 1.2.4**
 - 1.222.1 Available under license
- 1.223 javax.inject:javax.inject 1**
 - 1.223.1 Available under license
- 1.224 jackson-integration-for-metrics 4.0.5**
 - 1.224.1 Available under license
- 1.225 fastutil 8.2.3**
 - 1.225.1 Available under license
- 1.226 org.apache.groovy:groovy-json 4.0.1**
 - 1.226.1 Available under license
- 1.227 aop-alliance 1.0**
 - 1.227.1 Available under license

- 1.228 java-architecture-for-xml-binding 2.3.2**
 - 1.228.1 Available under license
- 1.229 jackson-module--guice 2.14.0**
 - 1.229.1 Available under license
- 1.230 micronaut-test-resources 1.1.2**
 - 1.230.1 Available under license
- 1.231 maven-artifact 3.8.3**
 - 1.231.1 Available under license
- 1.232 resilience4j 0.13.1**
 - 1.232.1 Available under license
- 1.233 micronaut-microstream 1.2.0**
 - 1.233.1 Available under license
- 1.234 javax-inject1-as-osgi-bundle 2.6.1**
 - 1.234.1 Available under license
- 1.235 fabric8--kubernetes-model--api-extensions 4.13.3**
 - 1.235.1 Available under license
- 1.236 io-grpc-grpc-okhttp 1.47.0**
 - 1.236.1 Available under license
- 1.237 libplexus-utils 3.3.0**
 - 1.237.1 Available under license
- 1.238 feign-jaxrs 8.18.0**
 - 1.238.1 Available under license
- 1.239 rest-assured-common 5.2.0**
 - 1.239.1 Available under license
- 1.240 micronaut-rxjava-3 2.3.0**
 - 1.240.1 Available under license
- 1.241 abego-core 1.0.3**
 - 1.241.1 Available under license
- 1.242 spock-framework--bill-of-materials 2.1-groovy-3.0**
 - 1.242.1 Available under license
- 1.243 hamcrest 1.3**
 - 1.243.1 Available under license
- 1.244 micronaut-coherence 3.5.1**
 - 1.244.1 Available under license
- 1.245 prometheus-java-span-context-supplier-opentelemetry-agent 0.15.0**
 - 1.245.1 Available under license
- 1.246 jackson-dataformat-csv 2.14.0**
 - 1.246.1 Available under license
- 1.247 jackson-bom 2.14.0**

- 1.247.1 Available under license
- 1.248 opentelemetry-java 1.9.1**
 - 1.248.1 Available under license
- 1.249 vavr 0.9.2**
 - 1.249.1 Available under license
- 1.250 jakarta-annotations-api 2.1.1**
 - 1.250.1 Available under license
- 1.251 fabric8-::-kubernetes-model-::-metrics 4.13.3**
 - 1.251.1 Available under license
- 1.252 fabric8-::-kubernetes-model-::-autoscaling 4.13.3**
 - 1.252.1 Available under license
- 1.253 feign-slf4j 8.18.0**
 - 1.253.1 Available under license
- 1.254 jakarta-expression-language-3.0-implementation 5.0.0-M1**
 - 1.254.1 Available under license
- 1.255 kafka-schema-registry-client 5.5.5**
 - 1.255.1 Available under license
- 1.256 micrometer-registry-prometheus 1.9.4**
 - 1.256.1 Available under license
- 1.257 io-swagger-swagger-annotations 1.6.2**
 - 1.257.1 Available under license
- 1.258 curator-framework 5.2.1**
 - 1.258.1 Available under license
- 1.259 kotlinox-coroutines-bom 1.5.1**
 - 1.259.1 Available under license
- 1.260 micronaut-tracing 4.4.0**
 - 1.260.1 Available under license
- 1.261 jakarta-ws-rs-api 2.1.6**
 - 1.261.1 Available under license
- 1.262 jetty-::-jakarta-servlet-api-and-schemas-for-jpms-and-osgi 5.0.2**
 - 1.262.1 Available under license
- 1.263 jackson-bom 2.13.4**
 - 1.263.1 Available under license
- 1.264 apache-zookeeper-jute 3.8.0**
 - 1.264.1 Available under license
- 1.265 opentest4j-opentest4j 1.2.0**
 - 1.265.1 Available under license
- 1.266 feign-jax-rs 10.12**
 - 1.266.1 Available under license

- 1.267 micronaut-aws 3.9.2**
 - 1.267.1 Available under license
- 1.268 metrics-core-library 2.2.0**
 - 1.268.1 Available under license
- 1.269 zjsonpatch 0.3.0**
 - 1.269.1 Available under license
- 1.270 netty/tomcatnative-[openssl---classes] 2.0.54.Final**
 - 1.270.1 Available under license
- 1.271 fabric8-::-kubernetes-model-::-common 4.13.3**
 - 1.271.1 Available under license
- 1.272 metrics-health-checks 4.0.5**
 - 1.272.1 Available under license
- 1.273 micronaut-rxjava-2 1.2.1**
 - 1.273.1 Available under license
- 1.274 openapi/swagger-support 4.0.1**
 - 1.274.1 Available under license
- 1.275 xml-path 5.2.0**
 - 1.275.1 Available under license
- 1.276 tagsoup 1.2.1**
 - 1.276.1 Available under license
- 1.277 netty/transport/classes/epoll 4.1.84.Final**
 - 1.277.1 Available under license
- 1.278 fabric8-::-kubernetes-model-::-node 4.13.3**
 - 1.278.1 Available under license
- 1.279 protobuf-java-format 1.2**
 - 1.279.1 Available under license
- 1.280 junit-jupiter-junit-jupiter-api 5.9.1**
 - 1.280.1 Available under license
- 1.281 snappy-java 1.1.8.4**
 - 1.281.1 Available under license
- 1.282 micronaut-email 1.3.2**
 - 1.282.1 Available under license
- 1.283 micronaut-r2dbc 4.0.0**
 - 1.283.1 Available under license
- 1.284 antlr 2.7.7**
 - 1.284.1 Available under license
- 1.285 jersey's-jersey 2.35**
 - 1.285.1 Available under license
- 1.286 micronaut-mongodb 4.6.0**

- 1.286.1 Available under license
- 1.287 fabric8--:--kubernetes-model--:--rbac 4.13.3**
 - 1.287.1 Available under license
- 1.288 config 5.5.5**
 - 1.288.1 Available under license
- 1.289 prometheus-java-span-context-supplier-opentelemetry 0.15.0**
 - 1.289.1 Available under license
- 1.290 micrometer-registry-jmx 1.9.4**
 - 1.290.1 Available under license
- 1.291 testcontainers-bom 1.17.5**
 - 1.291.1 Available under license
- 1.292 argparse 0.8.1**
 - 1.292.1 Available under license
- 1.293 micronaut-sql-libraries 4.7.2**
 - 1.293.1 Available under license
- 1.294 jackson-protobuf-support 0.9.9-jackson2.9-protobuf**
 - 1.294.1 Available under license
- 1.295 micrometer-metrics/micrometer 1.9.4**
 - 1.295.1 Available under license
- 1.296 osgi-resource-locator-bundle 1.0.3**
 - 1.296.1 Available under license
- 1.297 apache-commons-beanutils 1.9.4**
 - 1.297.1 Available under license
- 1.298 fabric8--:--kubernetes-model--:--coordination 4.13.3**
 - 1.298.1 Available under license
- 1.299 fabric8--:--kubernetes-model--:--policy 4.13.3**
 - 1.299.1 Available under license
- 1.300 jackson-core 2.14.0**
 - 1.300.1 Available under license
- 1.301 jakarta-mail 1.6.2**
 - 1.301.1 Available under license
- 1.302 metrics-utility-servlets 4.1.29**
 - 1.302.1 Available under license
- 1.303 micronaut-crack 1.0.1**
 - 1.303.1 Available under license
- 1.304 asm 7.2**
 - 1.304.1 Available under license
- 1.305 jose4j 0.7.9**
 - 1.305.1 Available under license

1.306 mockito 2.27.0

1.306.1 Available under license

1.307 micronaut-micrometer 4.6.1

1.307.1 Available under license

1.308 profiler 1.0.2

1.308.1 Available under license

1.309 curator-testing 5.2.1

1.309.1 Available under license

1.310 junit-platform-junit-platform-commons 1.9.1

1.310.1 Available under license

1.311 rest-assured:-bom 5.2.0

1.311.1 Available under license

1.312 javax-ws-rs-api 2.1.1

1.312.1 Available under license

1.313 error_prone_annotations 2.10.0

1.313.1 Available under license

1.314 aws-sdk-for-java---bom 1.12.301

1.314.1 Available under license

1.315 metrics-health-checks 4.1.11

1.315.1 Available under license

1.316 micronaut-picocli-configuration 4.3.0

1.316.1 Available under license

1.317 okhttp-logging-interceptor 3.12.12

1.317.1 Available under license

1.318 micronaut-kafka 4.4.0

1.318.1 Available under license

1.319 hdrhistogram 2.1.12

1.319.1 Available under license

1.320 protocol-buffer-java-util-package 3.20.1

1.320.1 Available under license

1.321 jakarta-activation 1.1.1

1.321.1 Available under license

1.322 io-projectreactor 3.4.23

1.322.1 Available under license

1.323 scala-java8-compat_2.12 1.0.2

1.323.1 Available under license

1.324 javassist 3.18.2-GA

1.324.1 Available under license

1.325 micronaut-redis 5.3.0

- 1.325.1 Available under license
- 1.326 snappy-java 1.1.7**
 - 1.326.1 Available under license
- 1.327 micronaut-test 3.6.2**
 - 1.327.1 Available under license
- 1.328 metrics-integration-with-jmx 4.2.10**
 - 1.328.1 Available under license
- 1.329 paranamer-core 2.8**
 - 1.329.1 Available under license
- 1.330 apache-log4j-jul-adapter 2.17.1**
 - 1.330.1 Available under license
- 1.331 google-guice 4.2.2**
 - 1.331.1 Available under license

1.1 reflections 0.9.10

1.1.1 Available under license :

DO WHAT THE FUCK YOU WANT TO PUBLIC LICENSE
Version 2, December 2004

Copyright (C) 2004 Sam Hocevar <sam@hocevar.net>

Everyone is permitted to copy and distribute verbatim or modified copies of this license document, and changing it is allowed as long as the name is changed.

DO WHAT THE FUCK YOU WANT TO PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. You just DO WHAT THE FUCK YOU WANT TO.

1.2 hamcrest v2.2

1.2.1 Available under license :

BSD License

Copyright (c) 2000-2015 www.hamcrest.org
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce

the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of Hamcrest nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.3 findbugs-annotations 3.0.1

1.3.1 Available under license :

Found license 'GNU Lesser General Public License' in 'This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public'

1.4 snake-yaml 1.33

1.4.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
<name>Apache License, Version 2.0</name>  
<url>http://www.apache.org/licenses/LICENSE-2.0.txt</url>
```

Found in path(s):

```
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/META-INF/maven/org.yaml/snakeyaml/pom.xml
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (c) 2008 Google Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-
jar/org/yaml/snakeyaml/external/com/google/gdata/util/common/base/UnicodeEscaper.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-
jar/org/yaml/snakeyaml/external/com/google/gdata/util/common/base/PercentEscaper.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-
jar/org/yaml/snakeyaml/external/com/google/gdata/util/common/base/Escaper.java
No license file was found, but licenses were detected in source scan.
```

```
// This module is multi-licensed and may be used under the terms
// EPL, Eclipse Public License, V1.0 or later, http://www.eclipse.org/legal
// LGPL, GNU Lesser General Public License, V2.1 or later, http://www.gnu.org/licenses/lgpl.html
// GPL, GNU General Public License, V2 or later, http://www.gnu.org/licenses/gpl.html
// AL, Apache License, V2.0 or later, http://www.apache.org/licenses
// BSD, BSD License, http://www.opensource.org/licenses/bsd-license.php
/**
 * A Base64 encoder/decoder.
 *
 * <p>
 * This class is used to encode and decode data in Base64 format as described in RFC 1521.
 *
 * <p>
 * Project home page: <a href="http://www.source-code.biz/base64coder/java/">www.
 * source-code.biz/base64coder/java</a><br>
 * Author: Christian d'Heureuse, Inventec Informatik AG, Zurich, Switzerland<br>
 * Multi-licensed: EPL / LGPL / GPL / AL / BSD.
 */
```

Found in path(s):

```
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-
jar/org/yaml/snakeyaml/external/biz/base64Coder/Base64Coder.java
No license file was found, but licenses were detected in source scan.
```

/**

* Copyright (c) 2008, SnakeYAML

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express

* or implied. See the License for the specific language governing permissions and limitations under
* the License.

*/

Found in path(s):

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-
jar/org/yaml/snakeyaml/introspector/PropertySubstitute.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-
jar/org/yaml/snakeyaml/introspector/FieldProperty.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-
jar/org/yaml/snakeyaml/tokens/TagToken.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-
jar/org/yaml/snakeyaml/parser/Production.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-
jar/org/yaml/snakeyaml/events/MappingEndEvent.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-
jar/org/yaml/snakeyaml/extensions/compactnotation/CompactConstructor.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-
jar/org/yaml/snakeyaml/util/ArrayStack.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-
jar/org/yaml/snakeyaml/LoaderOptions.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-
jar/org/yaml/snakeyaml/error/MissingEnvironmentVariableException.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-
jar/org/yaml/snakeyaml/extensions/compactnotation/CompactData.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-
jar/org/yaml/snakeyaml/reader/ReaderException.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-
jar/org/yaml/snakeyaml/tokens/CommentToken.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-
jar/org/yaml/snakeyaml/comments/CommentLine.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-
jar/org/yaml/snakeyaml/introspector/Property.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-
jar/org/yaml/snakeyaml/events/CommentEvent.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-
jar/org/yaml/snakeyaml/tokens/StreamEndToken.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/resolver/Resolver.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/serializer/Serializer.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/tokens/ScalarToken.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/introspector/BeanAccess.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/constructor/CustomClassLoaderConstructor.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/constructor/AbstractConstruct.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/util/ArrayUtils.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/scanner/ScannerImpl.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/DumperOptions.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/resolver/ResolverTuple.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/tokens/FlowEntryToken.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/tokens/FlowMappingEndToken.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/reader/UnicodeReader.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/events/AliasEvent.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/composer/Composer.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/constructor/DuplicateKeyException.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/events/DocumentStartEvent.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/events/ImplicitTuple.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/tokens/KeyToken.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/events/StreamEndEvent.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/events/MappingStartEvent.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/nodes/ScalarNode.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/scanner/ScannerException.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/representer/Represent.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/serializer/SerializerException.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/events/DocumentEndEvent.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/error/Mark.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/tokens/DocumentEndToken.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/tokens/FlowSequenceStartToken.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/constructor/Constructor.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/extensions/compactnotation/PackageCompactConstructor.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/tokens/AnchorToken.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/events/Event.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/tokens/DirectiveToken.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/tokens/BlockEntryToken.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/Yaml.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/nodes/CollectionNode.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/error/YAMLEException.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/nodes/MappingNode.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/constructor/BaseConstructor.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/tokens/Token.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/TypeDescription.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/representer/SafeRepresenter.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/emitter/EmitterException.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/tokens/BlockEndToken.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/parser/Parser.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/scanner/Scanner.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/events/StreamStartEvent.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/events/SequenceEndEvent.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/tokens/BlockMappingStartToken.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/parser/ParserImpl.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/events/CollectionEndEvent.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/events/NodeEvent.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/tokens/ValueToken.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/constructor/SafeConstructor.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/comments/CommentType.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/nodes/Node.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/events/ScalarEvent.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/serializer/NumberAnchorGenerator.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/nodes/Tag.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/scanner/Constant.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/constructor/Construct.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/parser/VersionTagsTuple.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/parser/ParserException.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/tokens/TagTuple.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/emitter/Emitable.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/emitter/Emitter.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/tokens/FlowSequenceEndToken.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/introspector/MethodProperty.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/constructor/ConstructorException.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/reader/StreamReader.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/emitter/ScalarAnalysis.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/comments/CommentEventsCollector.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/tokens/AliasToken.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/tokens/BlockSequenceStartToken.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/events/SequenceStartEvent.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/serializer/AnchorGenerator.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/util/PlatformFeatureDetector.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/nodes/SequenceNode.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/nodes/NodeId.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/representer/BaseRepresenter.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/tokens/DocumentStartToken.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/introspector/PropertyUtils.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/introspector/MissingProperty.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/tokens/StreamStartToken.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/introspector/GenericProperty.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/util/UriEncoder.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/tokens/FlowMappingStartToken.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/util/EnumUtils.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/events/CollectionStartEvent.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/representer/Representer.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/composer/ComposerException.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/nodes/AnchorNode.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/emitter/EmitterState.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/nodes/NodeTuple.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/env/EnvScalarConstructor.java

* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/scanner/SimpleKey.java
* /opt/cola/permits/1446188159_1666171012.444366/0/snakeyaml-1-33-sources-1-jar/org/yaml/snakeyaml/error/MarkedYAMLException.java

1.5 scala-collection-compat 2.6.0

1.5.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed

with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate

comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

scala-collection-compat

Copyright (c) 2002-2021 EPFL

Copyright (c) 2011-2021 Lightbend, Inc.

Scala includes software developed at LAMP/EPFL (<https://lamp.epfl.ch/>) and Lightbend, Inc. (<https://www.lightbend.com/>).

Licensed under the Apache License, Version 2.0 (the "License"). Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.6 bom 4.1.82.Final

1.6.1 Available under license :

Copyright (c) 2003-2008 Yuta Mori All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The person or persons who have associated work with this document (the "Dedicator" or "Certifier") hereby either (a) certifies that, to the best of his knowledge, the work of authorship identified is in the public domain of the country from which the work is published, or (b) hereby dedicates whatever copyright the dedicator holds in the work of authorship identified below (the "Work") to the public domain. A certifier, moreover, dedicates any copyright interest he may have in the associated work, and for these purposes, is described as a "dedicator" below.

A certifier has taken reasonable steps to verify the copyright status of this work. Certifier recognizes that his good faith efforts may not shield him from liability if in fact the work certified is not in the public domain.

Dedicator makes this dedication for the benefit of the public at large and to the detriment of the Dedicator's heirs and successors. Dedicator intends this dedication to be an overt act of relinquishment in perpetuity of all present and future rights under copyright law, whether vested or contingent, in the Work. Dedicator understands that such relinquishment of all rights includes the relinquishment of all rights to enforce (by lawsuit or otherwise) those copyrights in the Work.

Dedicator recognizes that, once placed in the public domain, the Work may be freely reproduced, distributed, transmitted, used, modified, built upon, or otherwise exploited by anyone for any purpose, commercial or non-commercial, and in any way, including by methods that have not yet been invented or conceived.

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise

designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. **Grant of Copyright License.** Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. **Grant of Patent License.** Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. **Redistribution.** You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must

include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly

negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,

including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf

of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<https://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

The MIT License

Copyright (c) 2009 William Kinney

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,

EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object

form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

Copyright 2009-2010 Ning, Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at <https://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT

WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

/*

* Copyright (c) 2004-2006, 2008, 2009, 2011 Apple Inc. All rights reserved.

*

* @APPLE_LICENSE_HEADER_START@

*

* This file contains Original Code and/or Modifications of Original Code

* as defined in and that are subject to the Apple Public Source License

* Version 2.0 (the 'License'). You may not use this file except in

* compliance with the License. Please obtain a copy of the License at

* <https://www.opensource.apple.com/appl/> and read it before using this

* file.

*

* The Original Code and all software distributed under the License are

* distributed on an 'AS IS' basis, WITHOUT WARRANTY OF ANY KIND, EITHER

* EXPRESS OR IMPLIED, AND APPLE HEREBY DISCLAIMS ALL SUCH WARRANTIES,

* INCLUDING WITHOUT LIMITATION, ANY WARRANTIES OF MERCHANTABILITY,

* FITNESS FOR A PARTICULAR PURPOSE, QUIET ENJOYMENT OR NON-INFRINGEMENT.

* Please see the License for the specific language governing rights and

* limitations under the License.

*

* @APPLE_LICENSE_HEADER_END@

*/

The MIT License (MIT)

Copyright (c) 2014 Cory Benfield

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright 2011, Google Inc.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2010-2011 Matthew J. Francis and Contributors of the jbzip2 Project

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The person or persons who have associated work with this document (the "Dedicator" or "Certifier") hereby either (a) certifies that, to the best of his knowledge, the work of authorship identified is in the public domain of

the country from which the work is published, or (b) hereby dedicates whatever copyright the dedicators holds in the work of authorship identified below (the "Work") to the public domain. A certifier, moreover, dedicates any copyright interest he may have in the associated work, and for these purposes, is described as a "dedicator" below.

A certifier has taken reasonable steps to verify the copyright status of this work. Certifier recognizes that his good faith efforts may not shield him from liability if in fact the work certified is not in the public domain.

Dedicator makes this dedication for the benefit of the public at large and to the detriment of the Dedicator's heirs and successors. Dedicator intends this dedication to be an overt act of relinquishment in perpetuate of all present and future rights under copyright law, whether vested or contingent, in the Work. Dedicator understands that such relinquishment of all rights includes the relinquishment of all rights to enforce (by lawsuit or otherwise) those copyrights in the Work.

Dedicator recognizes that, once placed in the public domain, the Work may be freely reproduced, distributed, transmitted, used, modified, built upon, or otherwise exploited by anyone for any purpose, commercial or non-commercial, and in any way, including by methods that have not yet been invented or conceived.

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of

this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only

on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<https://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

The MIT License

Copyright (c) 2012, 2014, 2015, 2016 Tatsuhiro Tsujikawa
Copyright (c) 2012, 2014, 2015, 2016 nghttp2 contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the

editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the

same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<https://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

(BSD License: <https://www.opensource.org/licenses/bsd-license>)

Copyright (c) 2011, Joe Walnes, Aslak Hellesøy and contributors
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of the Webbit nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR

BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The MIT License (MIT)

Copyright (c) 2000 - 2013 The Legion of the Bouncy Castle Inc.

(<https://www.bouncycastle.org>)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2000,2001,2002,2003,2004 ymnk, JCraft,Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The names of the authors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL JCRAFT, INC. OR ANY CONTRIBUTORS TO THIS SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA,

OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The Netty Project

=====

Please visit the Netty web site for more information:

* <https://netty.io/>

Copyright 2014 The Netty Project

The Netty Project licenses this file to you under the Apache License, version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at:

<https://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Also, please refer to each LICENSE.<component>.txt file, which is located in the 'license' directory of the distribution file, for the license terms of the components that this product depends on.

This product contains the extensions to Java Collections Framework which has been derived from the works by JSR-166 EG, Doug Lea, and Jason T. Greene:

* LICENSE:

* [license/LICENSE.jsr166y.txt](#) (Public Domain)

* HOMEPAGE:

* <http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/>

* <http://viewvc.jboss.org/cgi-bin/viewvc.cgi/jboss/cache/experimental/jsr166/>

This product contains a modified version of Robert Harder's Public Domain Base64 Encoder and Decoder, which can be obtained at:

* LICENSE:

* [license/LICENSE.base64.txt](#) (Public Domain)

* HOMEPAGE:

* <http://iharder.sourceforge.net/current/java/base64/>

This product contains a modified portion of 'Webbit', an event based WebSocket and HTTP server, which can be obtained at:

- * LICENSE:
 - * license/LICENSE.webbit.txt (BSD License)
- * HOMEPAGE:
 - * <https://github.com/joewalnes/webbit>

This product contains a modified portion of 'SLF4J', a simple logging facade for Java, which can be obtained at:

- * LICENSE:
 - * license/LICENSE.slf4j.txt (MIT License)
- * HOMEPAGE:
 - * <https://www.slf4j.org/>

This product contains a modified portion of 'Apache Harmony', an open source Java SE, which can be obtained at:

- * NOTICE:
 - * license/NOTICE.harmony.txt
- * LICENSE:
 - * license/LICENSE.harmony.txt (Apache License 2.0)
- * HOMEPAGE:
 - * <https://archive.apache.org/dist/harmony/>

This product contains a modified portion of 'jzip2', a Java bzip2 compression and decompression library written by Matthew J. Francis. It can be obtained at:

- * LICENSE:
 - * license/LICENSE.jzip2.txt (MIT License)
- * HOMEPAGE:
 - * <https://code.google.com/p/jzip2/>

This product contains a modified portion of 'libdivsufsort', a C API library to construct the suffix array and the Burrows-Wheeler transformed string for any input string of a constant-size alphabet written by Yuta Mori. It can be obtained at:

- * LICENSE:
 - * license/LICENSE.libdivsufsort.txt (MIT License)
- * HOMEPAGE:
 - * <https://github.com/y-256/libdivsufsort>

This product contains a modified portion of Nitsan Wakart's 'JCTools', Java Concurrency Tools for the JVM, which can be obtained at:

- * LICENSE:
 - * license/LICENSE.jctools.txt (ASL2 License)

- * HOMEPAGE:
 - * <https://github.com/JCTools/JCTools>

This product optionally depends on 'JZlib', a re-implementation of zlib in pure Java, which can be obtained at:

- * LICENSE:
 - * [license/LICENSE.jzlib.txt](#) (BSD style License)
- * HOMEPAGE:
 - * <http://www.jcraft.com/jzlib/>

This product optionally depends on 'Compress-LZF', a Java library for encoding and decoding data in LZF format, written by Tatu Saloranta. It can be obtained at:

- * LICENSE:
 - * [license/LICENSE.compress-lzf.txt](#) (Apache License 2.0)
- * HOMEPAGE:
 - * <https://github.com/ning/compress>

This product optionally depends on 'lz4', a LZ4 Java compression and decompression library written by Adrien Grand. It can be obtained at:

- * LICENSE:
 - * [license/LICENSE.lz4.txt](#) (Apache License 2.0)
- * HOMEPAGE:
 - * <https://github.com/jpountz/lz4-java>

This product optionally depends on 'lzma-java', a LZMA Java compression and decompression library, which can be obtained at:

- * LICENSE:
 - * [license/LICENSE.lzma-java.txt](#) (Apache License 2.0)
- * HOMEPAGE:
 - * <https://github.com/jponge/lzma-java>

This product optionally depends on 'zstd-jni', a zstd-jni Java compression and decompression library, which can be obtained at:

- * LICENSE:
 - * [license/LICENSE.zstd-jni.txt](#) (Apache License 2.0)
- * HOMEPAGE:
 - * <https://github.com/luben/zstd-jni>

This product contains a modified portion of 'jfastlz', a Java port of FastLZ compression and decompression library written by William Kinney. It can be obtained at:

- * LICENSE:
 - * [license/LICENSE.jfastlz.txt](#) (MIT License)

* HOMEPAGE:

* <https://code.google.com/p/jfastlz/>

This product contains a modified portion of and optionally depends on 'Protocol Buffers', Google's data interchange format, which can be obtained at:

* LICENSE:

* [license/LICENSE.protobuf.txt](#) (New BSD License)

* HOMEPAGE:

* <https://github.com/google/protobuf>

This product optionally depends on 'Bouncy Castle Crypto APIs' to generate a temporary self-signed X.509 certificate when the JVM does not provide the equivalent functionality. It can be obtained at:

* LICENSE:

* [license/LICENSE.bouncycastle.txt](#) (MIT License)

* HOMEPAGE:

* <https://www.bouncycastle.org/>

This product optionally depends on 'Snappy', a compression library produced by Google Inc, which can be obtained at:

* LICENSE:

* [license/LICENSE.snappy.txt](#) (New BSD License)

* HOMEPAGE:

* <https://github.com/google/snappy>

This product optionally depends on 'JBoss Marshalling', an alternative Java serialization API, which can be obtained at:

* LICENSE:

* [license/LICENSE.jboss-marshalling.txt](#) (Apache License 2.0)

* HOMEPAGE:

* <https://github.com/jboss-remoting/jboss-marshalling>

This product optionally depends on 'Caliper', Google's micro-benchmarking framework, which can be obtained at:

* LICENSE:

* [license/LICENSE.caliper.txt](#) (Apache License 2.0)

* HOMEPAGE:

* <https://github.com/google/caliper>

This product optionally depends on 'Apache Commons Logging', a logging framework, which can be obtained at:

* LICENSE:

- * license/LICENSE.common-logging.txt (Apache License 2.0)

- * HOMEPAGE:

- * <https://commons.apache.org/logging/>

This product optionally depends on 'Apache Log4J', a logging framework, which can be obtained at:

- * LICENSE:

- * license/LICENSE.log4j.txt (Apache License 2.0)

- * HOMEPAGE:

- * <https://logging.apache.org/log4j/>

This product optionally depends on 'Aalto XML', an ultra-high performance non-blocking XML processor, which can be obtained at:

- * LICENSE:

- * license/LICENSE.aalto-xml.txt (Apache License 2.0)

- * HOMEPAGE:

- * <https://wiki.fasterxml.com/AaltoHome>

This product contains a modified version of 'HPACK', a Java implementation of the HTTP/2 HPACK algorithm written by Twitter. It can be obtained at:

- * LICENSE:

- * license/LICENSE.hpack.txt (Apache License 2.0)

- * HOMEPAGE:

- * <https://github.com/twitter/hpack>

This product contains a modified version of 'HPACK', a Java implementation of the HTTP/2 HPACK algorithm written by Cory Benfield. It can be obtained at:

- * LICENSE:

- * license/LICENSE.hyper-hpack.txt (MIT License)

- * HOMEPAGE:

- * <https://github.com/python-hyper/hpack/>

This product contains a modified version of 'HPACK', a Java implementation of the HTTP/2 HPACK algorithm written by Tatsuhiro Tsujikawa. It can be obtained at:

- * LICENSE:

- * license/LICENSE.nghttp2-hpack.txt (MIT License)

- * HOMEPAGE:

- * <https://github.com/nghttp2/nghttp2/>

This product contains a modified portion of 'Apache Commons Lang', a Java library provides utilities for the java.lang API, which can be obtained at:

- * LICENSE:

- * license/LICENSE.common-lang.txt (Apache License 2.0)

- * HOMEPAGE:

- * <https://commons.apache.org/proper/commons-lang/>

This product contains the Maven wrapper scripts from 'Maven Wrapper', that provides an easy way to ensure a user has everything necessary to run the Maven build.

- * LICENSE:

- * license/LICENSE.mvn-wrapper.txt (Apache License 2.0)

- * HOMEPAGE:

- * <https://github.com/takari/maven-wrapper>

This product contains the dnsinfo.h header file, that provides a way to retrieve the system DNS configuration on MacOS.

This private header is also used by Apple's open source mDNSResponder (<https://opensource.apple.com/tarballs/mDNSResponder/>).

- * LICENSE:

- * license/LICENSE.dnsinfo.txt (Apple Public Source License 2.0)

- * HOMEPAGE:

- * <https://www.opensource.apple.com/source/configd/configd-453.19/dnsinfo/dnsinfo.h>

This product optionally depends on 'Brotli4j', Brotli compression and decompression for Java., which can be obtained at:

- * LICENSE:

- * license/LICENSE.brotli4j.txt (Apache License 2.0)

- * HOMEPAGE:

- * <https://github.com/hyperxpro/Brotli4j>

Protocol Buffers - Google's data interchange format

Copyright 2013 Google Inc. All rights reserved.

<https://developers.google.com/protocol-buffers/>

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Code generated by the Protocol Buffer compiler is owned by the owner of the input file used when generating it. This code is not standalone and requires a support library to be linked with it. This support library is itself covered by the above license.

/*

* Copyright (c) 2004-2007 QOS.ch

* All rights reserved.

*

* Permission is hereby granted, free of charge, to any person obtaining

* a copy of this software and associated documentation files (the

* "Software"), to deal in the Software without restriction, including

* without limitation the rights to use, copy, modify, merge, publish,

* distribute, sublicense, and/or sell copies of the Software, and to

* permit persons to whom the Software is furnished to do so, subject to

* the following conditions:

*

* The above copyright notice and this permission notice shall be

* included in all copies or substantial portions of the Software.

*

* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,

* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF

* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND

* NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE

* LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION

* OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION

* WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

*/

This copy of Aalto XML processor is licensed under the

Apache (Software) License, version 2.0 ("the License").

See the License for details about distribution rights, and the

specific rights regarding derivative works.

You may obtain a copy of the License at:

<https://www.apache.org/licenses/>

A copy is also included with both the the downloadable source code package and jar that contains class bytecodes, as file "ASL 2.0". In both cases, that file should be located next to this file: in source distribution the location should be "release-notes/asl"; and in jar "META-INF/" Apache Harmony

Copyright 2006, 2010 The Apache Software Foundation.

This product includes software developed at
The Apache Software Foundation (<https://www.apache.org/>).

1.7 micronaut-servlet 3.3.1

1.7.1 Available under license :

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You

institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.8 java-architecture-for-xml-binding 2.3.3

1.8.1 Available under license :

(See license.txt for the actual license terms)

Copyright 2001-@@YEAR@@ Sun Microsystems, Inc. 901 San Antonio Road, Palo Alto, California, 94303, U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in this product. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed at <http://www.sun.com/patents> and one or more additional patents or pending patent applications in the U.S. and other countries. This product is distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any. Third party software, including font technology, is copyrighted and licensed from Sun suppliers. Sun, the Sun logo, and Sun Microsystems are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>). Federal

Acquisitions: Commercial Software - Government Users Subject to Standard License Terms and Conditions.

Copyright 2001--@@YEAR@@ Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, Californie 94303 tats-Unis. Tous droits rserve. Distribue par des licences qui en restreignent l'utilisation. Sun Microsystems, Inc. a les droits de propriete intellectuels relatants la technologie incorpore dans ce produit. En particulier, et sans la limitation, ces droits de propriete intellectuels peuvent inclure un ou plus des brevets americains numrs <http://www.sun.com/patents> et un ou les brevets plus supplementaires ou les applications de brevet en attente dans les Etats Unis et les autres pays. Ce produit ou document est protge par un copyright et distribue avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la recompilation. Aucune partie de ce produit ou document ne peut tre reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation pralable et crite de Sun et de ses bailleurs de licence, s'il y en a. Le logiciel dtenu par des tiers, et qui comprend la technologie relative aux polices de caracteres, est protge par un copyright et licenci par des fournisseurs de Sun. Sun, le logo Sun, Sun Microsystems et sont des marques de fabrique ou des marques d'poses de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Ce produit inclut le logiciel dvelopp par la base de Apache Software Foundation (<http://www.apache.org/>). L'accord du gouvernement des tats Unis est requis avant l'exportation du produit.

/*

* The Apache Software License, Version 1.1

*

*

* Copyright (c) 1999-2004 The Apache Software Foundation. All rights reserved.

*

* Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

*

* 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

*

* 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

*

* 3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment:

* "This product includes software developed by the
* Apache Software Foundation (<http://www.apache.org/>)."
* Alternately, this acknowledgment may appear in the software itself,
* if and wherever such third-party acknowledgments normally appear.

*
* 4. The names "Xerces" and "Apache Software Foundation" must
* not be used to endorse or promote products derived from this
* software without prior written permission. For written
* permission, please contact apache@apache.org.

*
* 5. Products derived from this software may not be called "Apache",
* nor may "Apache" appear in their name, without prior written
* permission of the Apache Software Foundation.

* THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED
* WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
* DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR
* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
* USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
* ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
* OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
* OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.

* =====

*
* This software consists of voluntary contributions made by many
* individuals on behalf of the Apache Software Foundation and was
* originally based on software copyright (c) 1999, International
* Business Machines, Inc., <http://www.apache.org>. For more
* information on the Apache Software Foundation, please see
* <http://www.apache.org/>.

*/
THIS LICENSE IS INTENDED TO BE USED FOR DEBUGGING THE INSTALLER.

Amendment I

Congress shall make no law respecting an establishment of religion, or prohibiting the free exercise thereof; or abridging the freedom of speech, or of the press; or the right of the people peaceably to assemble, and to petition the government for a redress of grievances.

Amendment II

A well regulated militia, being necessary to the security of a free state, the right of the people to keep and bear arms, shall not be infringed.

Amendment III

No soldier shall, in time of peace be quartered in any house, without the consent of the owner, nor in time of war, but in a manner to be prescribed by law.

Amendment IV

The right of the people to be secure in their persons, houses, papers, and effects, against unreasonable searches and seizures, shall not be violated, and no warrants shall issue, but upon probable cause, supported by oath or affirmation, and particularly describing the place to be searched, and the persons or things to be seized.

Amendment V

No person shall be held to answer for a capital, or otherwise infamous crime, unless on a presentment or indictment of a grand jury, except in cases arising in the land or naval forces, or in the militia, when in actual service in time of war or public danger; nor shall any person be subject for the same offense to be twice put in jeopardy of life or limb; nor shall be compelled in any criminal case to be a witness against himself, nor be deprived of life, liberty, or property, without due process of law; nor shall private property be taken for public use, without just compensation.

Amendment VI

In all criminal prosecutions, the accused shall enjoy the right to a speedy and public trial, by an impartial jury of the state and district wherein the crime shall have been committed, which district shall have been previously ascertained by law, and to be informed of the nature and cause of the accusation; to be confronted with the witnesses against him; to have compulsory process for obtaining witnesses in his favor, and to have the assistance of counsel for his defense.

Amendment VII

In suits at common law, where the value in controversy shall exceed twenty dollars, the right of trial by jury shall be preserved, and no fact tried by a jury, shall be otherwise reexamined in any court of the United States, than according to the rules of the common law.

Amendment VIII

Excessive bail shall not be required, nor excessive fines imposed, nor cruel and unusual punishments inflicted.

Amendment IX

The enumeration in the Constitution, of certain rights, shall not be construed to deny or disparage others retained by the people.

Amendment X

The powers not delegated to the United States by the Constitution, nor prohibited by it to the states, are reserved to the states respectively, or to the people.

(See license.txt for the actual license terms)

Copyright 2001-@@YEAR@@ Sun Microsystems, Inc. 901 San Antonio Road, Palo Alto, California, 94303, U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in this product. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed at <http://www.sun.com/patents> and one or more additional patents or pending patent applications in the U.S. and other countries. This product is distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any. Third party software, including font technology, is copyrighted and licensed from Sun suppliers. Sun, the Sun logo, and Sun Microsystems are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>). Federal Acquisitions: Commercial Software - Government Users Subject to Standard License Terms and Conditions.

Copyright 2001-@@YEAR@@ Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, Californie 94303 tats-Unis. Tous droits rservs. Distribue par des licences qui en restreignent l'utilisation. Sun Microsystems, Inc. a les droits de propriit intellectuels relatants la technologie incorpore dans ce produit. En particulier, et sans la limitation, ces droits de propriit intellectuels peuvent inclure un ou plus des brevets amricains numrs <http://www.sun.com/patents> et un ou les brevets plus supplmentaires ou les applications de brevet en attente dans les Etats Unis et les autres pays. Ce produit ou document est protg par un copyright et distribu avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la dcompilation. Aucune partie de ce produit ou document ne peut tre reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation pralable et crite de Sun et de ses bailleurs de licence, s'il y en a. Le logiciel dtenu par des tiers, et qui comprend la technologie relative aux polices de caractres, est protg par un copyright et licenci par des fournisseurs de Sun. Sun, le logo Sun, Sun Microsystems et sont des marques de fabrique ou des marques d'pos'es de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Ce produit inclut le logiciel dvelopp par la base de Apache Software Foundation (<http://www.apache.org/>). L'accord du gouvernement des tats Unis est requis avant l'exportation du produit.

/* =====

* The Apache Software License, Version 1.1

*

* Copyright (c) 2001-2003 The Apache Software Foundation. All rights

* reserved.

*
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 *
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in
 * the documentation and/or other materials provided with the
 * distribution.
 *
 * 3. The end-user documentation included with the redistribution,
 * if any, must include the following acknowledgment:
 * "This product includes software developed by the
 * Apache Software Foundation (<http://www.apache.org/>)."
 * Alternately, this acknowledgment may appear in the software itself,
 * if and wherever such third-party acknowledgments normally appear.
 *
 * 4. The names "Apache" and "Apache Software Foundation" must
 * not be used to endorse or promote products derived from this
 * software without prior written permission. For written
 * permission, please contact apache@apache.org.
 *
 * 5. Products derived from this software may not be called "Apache",
 * nor may "Apache" appear in their name, without prior written
 * permission of the Apache Software Foundation.
 *
 * THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED
 * WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
 * DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR
 * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
 * LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
 * USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
 * ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
 * OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
 * OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 * =====
 *
 * This software consists of voluntary contributions made by many
 * individuals on behalf of the Apache Software Foundation. For more
 * information on the Apache Software Foundation, please see
 * <<http://www.apache.org/>>.

*/
/* =====
* The Apache Software License, Version 1.1
*
* Copyright (c) 2000 The Apache Software Foundation. All rights
* reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
*
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in
* the documentation and/or other materials provided with the
* distribution.
*
* 3. The end-user documentation included with the redistribution,
* if any, must include the following acknowledgment:
* "This product includes software developed by the
* Apache Software Foundation (<http://www.apache.org/>)."
* Alternately, this acknowledgment may appear in the software itself,
* if and wherever such third-party acknowledgments normally appear.
*
* 4. The names "Apache" and "Apache Software Foundation" must
* not be used to endorse or promote products derived from this
* software without prior written permission. For written
* permission, please contact apache@apache.org.
*
* 5. Products derived from this software may not be called "Apache",
* nor may "Apache" appear in their name, without prior written
* permission of the Apache Software Foundation.
*
* THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
* WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
* DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR
* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
* USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
* ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
* OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
* OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
* =====

- *
* This software consists of voluntary contributions made by many
* individuals on behalf of the Apache Software Foundation. For more
* information on the Apache Software Foundation, please see
* <<http://www.apache.org/>>.
*
- * Portions of this software are based upon public domain software
* originally written at the National Center for Supercomputing Applications,
* University of Illinois, Urbana-Champaign.
*/

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate

as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify

the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include

the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

1. Definitions.

1.1. "Contributor" means each individual or entity that creates or contributes to the creation of Modifications.

1.2. "Contributor Version" means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. "Covered Software" means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. "Executable" means the Covered Software in any form other than Source Code.

1.5. "Initial Developer" means the individual or entity that first makes Original Software available under this License.

1.6. "Larger Work" means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. "License" means this document.

1.8. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. "Modifications" means the Source Code and Executable form of any of the following:

- A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;
- B. Any new file that contains any part of the Original Software or previous Modification; or
- C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. "Original Software" means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. "Source Code" means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your

original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipients rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as "Participant") alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTYS NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and "commercial computer software documentation" as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdictions conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court

costs and reasonable attorneys fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

Copyright (c) 2001, Sun Microsystems, Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Sun Microsystems, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,

STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This license came from: <http://www.megginson.com/SAX/copying.html>

However please note future versions of SAX may be covered under <http://saxproject.org/?selected=pd>

This page is now out of date -- see the new SAX site at <http://www.saxproject.org/> for more up-to-date releases and other information. Please change your bookmarks.

SAX2 is Free!

I hereby abandon any property rights to SAX 2.0 (the Simple API for XML), and release all of the SAX 2.0 source code, compiled code, and documentation contained in this distribution into the Public Domain. SAX comes with NO WARRANTY or guarantee of fitness for any purpose.

David Megginson, david@megginson.com

2000-05-05

This license came from:

<http://www.w3.org/Consortium/Legal/copyright-software-19980720>

W3C SOFTWARE NOTICE AND LICENSE

Copyright 1994-2001 World

Wide Web Consortium, (<http://www.w3.org/>) World

Wide Web Consortium

(<http://www.lcs.mit.edu/>) Massachusetts Institute of

Technology, (<http://www.inria.fr/>) Institut National de

Recherche en Informatique et en Automatique, (<http://www.keio.ac.jp/>) Keio University). All Rights Reserved.

<http://www.w3.org/Consortium/Legal/>

This W3C work (including software, documents, or other related items) is being provided by the copyright holders under the following license. By obtaining, using and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions:

Permission to use, copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications, that you make:

The full text of this NOTICE in a location viewable to users of the redistributed or derivative work.

Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, a short notice of the following form (hypertext is preferred, text is permitted) should be used within the body of any redistributed or derivative code:

"Copyright [\$date-of-software] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University). All Rights Reserved.
<http://www.w3.org/Consortium/Legal/>"

Notice of any changes or modifications to the W3C files, including the date changes were made. (We recommend you provide URIs to the location from which the code is derived.)

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS. COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR DOCUMENTATION.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to the software without specific, written prior permission. Title to copyright in this software and any associated documentation will at all times remain with copyright holders.

This formulation of W3C's notice and license became active on August 14 1998 so as to improve compatibility with GPL. This version ensures that W3C software licensing terms are no more restrictive than GPL and consequently W3C software may be distributed in GPL packages. See the older formulation for the policy prior to this date. Please see our Copyright FAQ for common questions about using materials from our site, including specific terms and conditions for packages like libwww, Amaya, and Jigsaw. Other questions about this notice can be directed to site-policy@w3.org.

webmaster
Sun Microsystems, Inc.
Binary Code License Agreement

READ THE TERMS OF THIS AGREEMENT AND ANY PROVIDED SUPPLEMENTAL LICENSE TERMS (COLLECTIVELY "AGREEMENT") CAREFULLY BEFORE OPENING THE SOFTWARE MEDIA PACKAGE. BY OPENING THE SOFTWARE MEDIA PACKAGE, YOU AGREE TO THE TERMS OF THIS AGREEMENT. IF YOU ARE ACCESSING THE SOFTWARE ELECTRONICALLY, INDICATE YOUR ACCEPTANCE OF THESE TERMS BY SELECTING THE "ACCEPT" BUTTON AT THE END OF THIS AGREEMENT. IF YOU DO NOT AGREE TO ALL THESE TERMS, PROMPTLY RETURN THE UNUSED SOFTWARE TO YOUR PLACE OF PURCHASE FOR A REFUND OR, IF THE SOFTWARE IS ACCESSED ELECTRONICALLY, SELECT THE "DECLINE" BUTTON AT THE END OF THIS AGREEMENT.

1. LICENSE TO USE. Sun grants you a non-exclusive and non-transferable license for the internal use only of the accompanying software and documentation and any error corrections provided by Sun (collectively "Software"), by the number of users and the class of computer hardware for which the corresponding fee has been paid.

2. RESTRICTIONS. Software is confidential and copyrighted. Title to Software and all associated intellectual property rights is retained by Sun and/or its licensors. Except as specifically authorized in any Supplemental License Terms, you may not make copies of Software, other than a single copy of Software for archival purposes. Unless enforcement is prohibited by applicable law, you may not modify, decompile, or reverse engineer Software. You acknowledge that Software is not designed, licensed or intended for use in the design, construction, operation or maintenance of any nuclear facility. Sun disclaims any express or implied warranty of fitness for such uses. No right, title or interest in or to any trademark, service mark, logo or trade name of Sun or its licensors is granted under this Agreement.

3. LIMITED WARRANTY. Sun warrants to you that for a period of ninety (90) days from the date of purchase, as evidenced by a copy of the receipt, the media on which Software is furnished (if any) will be free of defects in materials and workmanship under normal use. Except for the foregoing, Software is provided "AS IS". Your exclusive remedy and Sun's entire liability under this limited warranty will be at Sun's option to replace Software media or refund the fee paid for Software.

4. DISCLAIMER OF WARRANTY. UNLESS SPECIFIED IN THIS AGREEMENT, ALL EXPRESS OR IMPLIED CONDITIONS,

REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT THESE DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

5. LIMITATION OF LIABILITY. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL SUN OR ITS LICENSORS BE LIABLE FOR ANY LOST REVENUE, PROFIT OR DATA, OR FOR SPECIAL, INDIRECT, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF OR RELATED TO THE USE OF OR INABILITY TO USE SOFTWARE, EVEN IF SUN HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

In no event will Sun's liability to you, whether in contract, tort (including negligence), or otherwise, exceed the amount paid by you for Software under this Agreement. The foregoing limitations will apply even if the above stated warranty fails of its essential purpose.

6. Termination. This Agreement is effective until terminated. You may terminate this Agreement at any time by destroying all copies of Software. This Agreement will terminate immediately without notice from Sun if you fail to comply with any provision of this Agreement. Upon Termination, you must destroy all copies of Software.

7. Export Regulations. All Software and technical data delivered under this Agreement are subject to US export control laws and may be subject to export or import regulations in other countries. You agree to comply strictly with all such laws and regulations and acknowledge that you have the responsibility to obtain such licenses to export, re-export, or import as may be required after delivery to you.

8. U.S. Government Restricted Rights. If Software is being acquired by or on behalf of the U.S. Government or by a U.S. Government prime contractor or subcontractor (at any tier), then the Government's rights in Software and accompanying documentation will be only as set forth in this Agreement; this is in accordance with 48 CFR 227.7201 through 227.7202-4 (for Department of Defense (DOD) acquisitions) and with 48 CFR 2.101 and 12.212 (for non-DOD acquisitions).

9. Governing Law. Any action related to this Agreement will be governed by California law and controlling U.S. federal law. No choice of law rules of any jurisdiction

will apply.

10. Severability. If any provision of this Agreement is held to be unenforceable, this Agreement will remain in effect with the provision omitted, unless omission would frustrate the intent of the parties, in which case this Agreement will immediately terminate.

11. Integration. This Agreement is the entire agreement between you and Sun relating to its subject matter. It supersedes all prior or contemporaneous oral or written communications, proposals, representations and warranties and prevails over any conflicting or additional terms of any quote, order, acknowledgment, or other communication between the parties relating to its subject matter during the term of this Agreement. No modification of this Agreement will be binding, unless in writing and signed by an authorized representative of each party.

SUN XML INSTANCE GENERATOR, VERSION 1.0 SUPPLEMENTAL LICENSE TERMS

These supplemental license terms ("Supplemental Terms") add to or modify the terms of the Binary Code License Agreement (collectively, the "Agreement"). Capitalized terms not defined in these Supplemental Terms shall have the same meanings ascribed to them in the Agreement. These Supplemental Terms shall supersede any inconsistent or conflicting terms in the Agreement, or in any license contained within the Software.

1. Software Internal Use and Development License Grant. Subject to the terms and conditions of this Agreement, including, but not limited to Section 4 (Java(TM) Technology Restrictions) of these Supplemental Terms, Sun grants to you, a non-exclusive, non-transferable, royalty-free and limited license to reproduce, modify, and create derivative works of the Software for the sole purpose of adding value and improving the Software for the development of applications ("Programs").

2. License to Distribute Software. Subject to the terms and conditions of this Agreement, including, but not limited to Section 4 (Java (TM) Technology Restrictions) of these Supplemental Terms, Sun grants you a non-exclusive, non-transferable, limited license to reproduce and distribute the Software modified by you as permitted in

Section 1 of these Supplemental Terms ("Modified Software") in source or binary code form, provided that (i) you distribute the Modified Software only bundled as part of, and for the sole purpose of running, your Programs, (ii) the Modified Software adds value and improves the function of the Software, (iv) you do not remove or alter any proprietary legends or notices contained in the Software, (v) you only distribute the Modified Software subject to a license agreement that protects Sun's interests consistent with the terms contained in this Agreement, and (vi) you agree to defend and indemnify Sun and its licensors from and against any damages, costs, liabilities, settlement amounts and/or expenses (including attorneys' fees) incurred in connection with any claim, lawsuit or action by any third party that arises or results from the use or distribution of any and all Programs and/or Modified Software.

3. Experimental Software. You acknowledge that the Software is experimental and may contain errors, defects, or deficiencies which cannot or will not be corrected by Sun. You shall have the sole responsibility to protect adequately and backup your data and/or equipment used in connection with the Software. You shall not claim against Sun for lost data, re-run time, inaccurate output, work delays or lost profits resulting from your use of the Licensed Software.

4. Java Technology Restrictions. You may not modify the Java Platform Interface ("JPI", identified as classes contained within the "java" package or any subpackages of the "java" package), by creating additional classes within the JPI or otherwise causing the addition to or modification of the classes in the JPI. In the event that you create an additional class and associated API(s) which (i) extends the functionality of the Java platform, and (ii) is exposed to third party software developers for the purpose of developing additional software which invokes such additional API, you must promptly publish broadly an accurate specification for such API for free use by all developers. You may not create, or authorize your licensees to create, additional classes, interfaces, or subpackages that are in any way identified as "java", "javax", "sun" or similar convention as specified by Sun in any naming convention designation.

5. Trademarks and Logos. You acknowledge and agree as between you and Sun that Sun owns the SUN, SOLARIS, JAVA, JINI, FORTE, and iPLANET trademarks and all SUN, SOLARIS, JAVA, JINI, FORTE, and iPLANET-related trademarks, service

marks, logos and other brand designations ("Sun Marks"), and you agree to comply with the Sun Trademark and Logo Usage Requirements currently located at <http://www.sun.com/policies/trademarks>. Any use you make of the Sun Marks inures to Sun's benefit.

6. Termination for Infringement. Either party may terminate this Agreement immediately should any Software become, or in either party's opinion be likely to become, the subject of a claim of infringement of any intellectual property right.

For inquiries please contact:

Sun Microsystems, Inc.

901 San Antonio Road, Palo Alto, California 94303

(LFI#100313/Form ID#011801)

Copyright (c) 2003, Kohsuke Kawaguchi

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2003 Sun Microsystems, Inc. All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

-Redistribution of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

-Redistribution in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of Sun Microsystems, Inc. or the names of contributors may be used to endorse or promote products derived from this software without specific prior written permission.

This software is provided "AS IS," without a warranty of any kind. ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE HEREBY EXCLUDED. SUN MIDROSYSTEMS, INC. ("SUN") AND ITS LICENSORS SHALL NOT BE LIABLE FOR ANY DAMAGES SUFFERED BY LICENSEE AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THIS SOFTWARE OR ITS DERIVATIVES. IN NO EVENT WILL SUN OR ITS LICENSORS BE LIABLE FOR ANY LOST REVENUE, PROFIT OR DATA, OR FOR DIRECT, INDIRECT, SPECIAL, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF THE USE OF OR INABILITY TO USE THIS SOFTWARE, EVEN IF SUN HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You acknowledge that this software is not designed, licensed or intended for use in the design, construction, operation or maintenance of any nuclear facility.

Sun Microsystems, Inc.
Binary Code License Agreement

READ THE TERMS OF THIS AGREEMENT AND ANY PROVIDED SUPPLEMENTAL LICENSE TERMS (COLLECTIVELY "AGREEMENT") CAREFULLY BEFORE OPENING THE SOFTWARE MEDIA PACKAGE. BY OPENING THE SOFTWARE MEDIA PACKAGE, YOU AGREE TO THE TERMS OF THIS AGREEMENT. IF YOU ARE ACCESSING THE SOFTWARE ELECTRONICALLY, INDICATE YOUR ACCEPTANCE OF THESE TERMS BY SELECTING THE "ACCEPT" BUTTON AT THE END OF THIS AGREEMENT. IF YOU DO NOT AGREE TO ALL THESE TERMS, PROMPTLY RETURN THE UNUSED SOFTWARE TO YOUR PLACE OF PURCHASE FOR A REFUND OR, IF THE SOFTWARE IS ACCESSED ELECTRONICALLY, SELECT THE "DECLINE" BUTTON AT THE END OF THIS AGREEMENT.

1. LICENSE TO USE. Sun grants you a non-exclusive and non-transferable license for the internal use only of the accompanying software and documentation and any error corrections provided by Sun (collectively "Software"), by the number of users and the class of computer hardware for which the corresponding fee has been paid.

2. **RESTRICTIONS.** Software is confidential and copyrighted. Title to Software and all associated intellectual property rights is retained by Sun and/or its licensors. Except as specifically authorized in any Supplemental License Terms, you may not make copies of Software, other than a single copy of Software for archival purposes. Unless enforcement is prohibited by applicable law, you may not modify, decompile, or reverse engineer Software. You acknowledge that Software is not designed, licensed or intended for use in the design, construction, operation or maintenance of any nuclear facility. Sun disclaims any express or implied warranty of fitness for such uses. No right, title or interest in or to any trademark, service mark, logo or trade name of Sun or its licensors is granted under this Agreement.

3. **LIMITED WARRANTY.** Sun warrants to you that for a period of ninety (90) days from the date of purchase, as evidenced by a copy of the receipt, the media on which Software is furnished (if any) will be free of defects in materials and workmanship under normal use. Except for the foregoing, Software is provided "AS IS". Your exclusive remedy and Sun's entire liability under this limited warranty will be at Sun's option to replace Software media or refund the fee paid for Software.

4. **DISCLAIMER OF WARRANTY.** UNLESS SPECIFIED IN THIS AGREEMENT, ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT THESE DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

5. **LIMITATION OF LIABILITY.** TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL SUN OR ITS LICENSORS BE LIABLE FOR ANY LOST REVENUE, PROFIT OR DATA, OR FOR SPECIAL, INDIRECT, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF OR RELATED TO THE USE OF OR INABILITY TO USE SOFTWARE, EVEN IF SUN HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. In no event will Sun's liability to you, whether in contract, tort (including negligence), or otherwise, exceed the amount paid by you for Software under this Agreement. The foregoing limitations will apply even if the above stated warranty fails of its essential purpose.

6. **Termination.** This Agreement is effective until terminated. You may terminate this Agreement at any time by destroying all copies of Software. This Agreement will terminate immediately without notice from Sun if you fail to comply with any provision of this Agreement. Upon Termination, you must destroy all copies of Software.

7. **Export Regulations.** All Software and technical data delivered under this Agreement are subject to US export control laws and may be subject to export or import regulations in other countries. You agree to comply strictly with all such laws and regulations and acknowledge that you have the

responsibility to obtain such licenses to export, re-export, or import as may be required after delivery to you.

8. U.S. Government Restricted Rights. If Software is being acquired by or on behalf of the U.S. Government or by a U.S. Government prime contractor or subcontractor (at any tier), then the Government's rights in Software and accompanying documentation will be only as set forth in this Agreement; this is in accordance with 48 CFR 227.7201 through 227.7202-4 (for Department of Defense (DOD) acquisitions) and with 48 CFR 2.101 and 12.212 (for non-DOD acquisitions).

9. Governing Law. Any action related to this Agreement will be governed by California law and controlling U.S. federal law. No choice of law rules of any jurisdiction will apply.

10. Severability. If any provision of this Agreement is held to be unenforceable, this Agreement will remain in effect with the provision omitted, unless omission would frustrate the intent of the parties, in which case this Agreement will immediately terminate.

11. Integration. This Agreement is the entire agreement between you and Sun relating to its subject matter. It supersedes all prior or contemporaneous oral or written communications, proposals, representations and warranties and prevails over any conflicting or additional terms of any quote, order, acknowledgment, or other communication between the parties relating to its subject matter during the term of this Agreement. No modification of this Agreement will be binding, unless in writing and signed by an authorized representative of each party.

JAVA OPTIONAL PACKAGE

JAVABEANS(TM) ACTIVATION FRAMEWORK, VERSION 1.0.2 SUPPLEMENTAL LICENSE TERMS

These supplemental license terms ("Supplemental Terms") add to or modify the terms of the Binary Code License Agreement (collectively, the "Agreement"). Capitalized terms not defined in these Supplemental Terms shall have the same meanings ascribed to them in the Agreement. These Supplemental Terms shall supersede any inconsistent or conflicting terms in the Agreement, or in any license contained within the Software.

1. Software Internal Use and Development License Grant. Subject to the terms and conditions of this Agreement, including, but not limited to Section 3 (Java(TM) Technology Restrictions) of these Supplemental Terms, Sun grants you a non-exclusive, non-transferable, limited license to reproduce internally and use internally the binary form of the Software, complete and unmodified, for the sole purpose of designing, developing and testing your Java applets and applications ("Programs").

2. License to Distribute Software. In addition to the license granted in Section 1 (Software Internal Use and Development License Grant) of these Supplemental Terms, subject to the terms and conditions of this Agreement, including but not limited to, Section 3 (Java Technology Restrictions) of these Supplemental Terms, Sun grants you a non-exclusive, non-transferable, limited license to reproduce and distribute the Software in binary code form only, provided that you (i) distribute the Software complete and unmodified and only bundled as part of your Programs, (ii) do not distribute additional software intended to replace any component(s) of the Software, (iii) do not remove or alter any proprietary legends or notices contained in the Software, (iv) only distribute the Software subject to a license agreement that protects Sun's interests consistent with the terms contained in this Agreement, and (v) agree to defend and indemnify Sun and its licensors from and against any damages, costs, liabilities, settlement amounts and/or expenses (including attorneys' fees) incurred in connection with any claim, lawsuit or action by any third party that arises or results from the use or distribution of any and all Programs and/or Software.

3. Java Technology Restrictions. You may not modify the Java Platform Interface ("JPI", identified as classes contained within the "java" package or any subpackages of the "java" package), by creating additional classes within the JPI or otherwise causing the addition to or modification of the classes in the JPI. In the event that you create an additional class and associated API(s) which (i) extends the functionality of the Java platform, and (ii) is exposed to third party software developers for the purpose of developing additional software which invokes such additional API, you must promptly publish broadly an accurate specification for such API for free use by all developers. You may not create, or authorize your licensees to create additional classes, interfaces, or subpackages that are in any way identified as "java", "javax", "sun" or similar convention as specified by Sun in any naming convention designation.

4. No Support. Sun is under no obligation to support the Software or to provide you with updates or error corrections. You acknowledge that the Software may have defects or deficiencies which cannot or will not be corrected by Sun.

5. Trademarks and Logos. You acknowledge and agree as between you and Sun that Sun owns the SUN, SOLARIS, JAVA, JINI, FORTE, and iPLANET trademarks and all SUN, SOLARIS, JAVA, JINI, FORTE, and iPLANET-related trademarks, service marks, logos and other brand designations ("Sun Marks"), and you agree to comply with the Sun Trademark and Logo Usage Requirements currently located at <http://www.sun.com/policies/trademarks>. Any use you make of the Sun Marks inures to Sun's benefit.

6. Source Code. Software may contain source code that is provided solely for reference purposes pursuant to the terms of this Agreement. Source code may

not be redistributed unless expressly provided for in this Agreement.

7. Termination for Infringement. Either party may terminate this Agreement immediately should any Software become, or in either party's opinion be likely to become, the subject of a claim of infringement of any intellectual property right.

For inquiries please contact: Sun Microsystems, Inc. 901 San Antonio Road,
Palo Alto, California 94303
(LFI#115020/Form ID#011801)

/*

* The Apache Software License, Version 1.1

*

*

* Copyright (c) 1999-2002 The Apache Software Foundation. All rights
* reserved.

*

* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:

*

* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.

*

* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in
* the documentation and/or other materials provided with the
* distribution.

*

* 3. The end-user documentation included with the redistribution,
* if any, must include the following acknowledgment:

* "This product includes software developed by the
* Apache Software Foundation (<http://www.apache.org/>)."
* Alternately, this acknowledgment may appear in the software itself,
* if and wherever such third-party acknowledgments normally appear.

*

* 4. The names "Xerces" and "Apache Software Foundation" must
* not be used to endorse or promote products derived from this
* software without prior written permission. For written
* permission, please contact apache@apache.org.

*

* 5. Products derived from this software may not be called "Apache",
* nor may "Apache" appear in their name, without prior written
* permission of the Apache Software Foundation.

*

* THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED
* WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE

* DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR
* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
* USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
* ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
* OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
* OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.

* =====
*

* This software consists of voluntary contributions made by many
* individuals on behalf of the Apache Software Foundation and was
* originally based on software copyright (c) 1999, International
* Business Machines, Inc., <http://www.ibm.com>. For more
* information on the Apache Software Foundation, please see
* <<http://www.apache.org/>>.

*/

Copyright (c) 2004 Kohsuke Kawaguchi

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, sublicense, and/or
sell copies of the Software, and to permit persons to whom
the Software is furnished to do so, subject to the following
conditions:

The above copyright notice and this permission notice shall
be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY
KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR
PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS
OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR
OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR
OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

1. Definitions.

1.1. Contributor. means each individual or entity that creates or
contributes to the creation of Modifications.

1.2. Contributor Version. means the combination of the Original

Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software. means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable. means the Covered Software in any form other than Source Code.

1.5. Initial Developer. means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work. means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License. means this document.

1.8. Licensable. means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications. means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software. means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims. means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code. means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You. (or .Your.) means an individual or a legal entity

exercising rights under, and complying with all of the terms of, this License. For legal entities, .You. includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, .control. means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such

Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not

remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN .AS IS. BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as .Participant.) alleging that the Participant Software

(meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a .commercial item., as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of .commercial computer software. (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and .commercial computer software documentation. as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this

license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and

modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an

appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not

distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the

author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.

Copyright (C)

This program is free software; you can redistribute it and/or modify

it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show
w'. This is free software, and you are welcome to redistribute it under
certain conditions; type `show c' for details.
```

The hypothetical commands ``show w'` and ``show c'` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than ``show w'` and ``show c'`; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the
program `Gnomovision' (which makes passes at compilers) written by James
Hacker.
```

```
signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

"CLASSPATH" EXCEPTION TO THE GPL VERSION 2

Certain source files distributed by Sun Microsystems, Inc. are subject to the following clarification and special exception to the GPL Version 2, but only where Sun has expressly included in the particular source file's header the words

"Sun designates this particular file as subject to the "Classpath" exception as provided by Sun in the License file that accompanied this code."

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License Version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version. Copyright (c) 2001-@@YEAR@@ Sun Microsystems, Inc. All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistribution in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of Sun Microsystems, Inc. or the names of contributors may be used to endorse or promote products derived from this software without specific prior written permission.

This software is provided "AS IS," without a warranty of any kind. ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE HEREBY EXCLUDED. SUN AND ITS LICENSORS SHALL NOT BE LIABLE FOR ANY DAMAGES OR LIABILITIES SUFFERED BY LICENSEE AS A RESULT OF OR RELATING TO USE, MODIFICATION OR DISTRIBUTION OF THE SOFTWARE OR ITS DERIVATIVES. IN NO EVENT WILL

SUN OR ITS LICENSORS BE LIABLE FOR ANY LOST REVENUE, PROFIT OR DATA, OR FOR DIRECT, INDIRECT, SPECIAL, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF THE USE OF OR INABILITY TO USE SOFTWARE, EVEN IF SUN HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You acknowledge that Software is not designed, licensed or intended for use in the design, construction, operation or maintenance of any nuclear facility.

Copyright (c) 2003, Kohsuke Kawaguchi

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright 2001-2005 (C) MetaStuff, Ltd. All Rights Reserved.

Redistribution and use of this software and associated documentation ("Software"), with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain copyright

statements and notices. Redistributions must also contain a copy of this document.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name "DOM4J" must not be used to endorse or promote products derived from this Software without prior written permission of MetaStuff, Ltd. For written permission, please contact dom4j-info@metastuff.com.
4. Products derived from this Software may not be called "DOM4J" nor may "DOM4J" appear in their names without prior written permission of MetaStuff, Ltd. DOM4J is a registered trademark of MetaStuff, Ltd.
5. Due credit should be given to the DOM4J Project - <http://www.dom4j.org>

THIS SOFTWARE IS PROVIDED BY METASTUFF, LTD. AND CONTRIBUTORS ``AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL METASTUFF, LTD. OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright 2001-@@YEAR@@ Sun Microsystems, Inc. 901 San Antonio Road, Palo Alto, California, 94303, U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in this product. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed at <http://www.sun.com/patents> and one or more additional patents or pending patent applications in the U.S. and other countries. This product is distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any. Third party software, including font technology, is copyrighted and licensed from Sun suppliers. Sun, the Sun logo, and Sun Microsystems are trademarks or registered trademarks of Sun Microsystems, Inc. in the

U.S. and other countries. This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>). Federal Acquisitions: Commercial Software - Government Users Subject to Standard License Terms and Conditions.

Copyright 2001-@@YEAR@@ Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, Californie 94303 tats-Unis. Tous droits rservs. Distribue par des licences qui en restreignent l'utilisation. Sun Microsystems, Inc. a les droits de propriit intellectuels relatants la technologie incorpore dans ce produit. En particulier, et sans la limitation, ces droits de propriit intellectuels peuvent inclure un ou plus des brevets amricains numrs <http://www.sun.com/patents> et un ou les brevets plus supplmentaires ou les applications de brevet en attente dans les Etats Unis et les autres pays. Ce produit ou document est protg par un copyright et distribu avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la dcompilation. Aucune partie de ce produit ou document ne peut tre reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation pralable et crite de Sun et de ses bailleurs de licence, s'il y en a. Le logiciel dtenu par des tiers, et qui comprend la technologie relative aux polices de caractres, est protg par un copyright et licenci par des fournisseurs de Sun. Sun, le logo Sun, Sun Microsystems et sont des marques de fabrique ou des marques d'pos?es de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Ce produit inclut le logiciel dvelopp par la base de Apache Software Foundation (<http://www.apache.org/>). L'accord du gouvernement des tats Unis est requis avant l'exportation du produit.

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

1. Definitions.

1.1. Contributor. means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version. means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software. means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable. means the Covered Software in any form other than Source Code.

1.5. Initial Developer. means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work. means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License. means this document.

1.8. Licensable. means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications. means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software. means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims. means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code. means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You. (or .Your.) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, .You. includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, .control. means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the

Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN .AS IS. BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as .Participant.) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a .commercial item., as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of .commercial computer software. (as that term is defined at 48 C.F.R. ? 252.227-7014(a)(1)) and .commercial computer software documentation. as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction.s conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys. fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its

contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER

PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.

Copyright (C)

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author

Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

"CLASSPATH" EXCEPTION TO THE GPL VERSION 2

Certain source files distributed by Sun Microsystems, Inc. are subject to the following clarification and special exception to the GPL Version 2, but only where Sun has expressly included in the particular source file's header the words

"Sun designates this particular file as subject to the "Classpath" exception as provided by Sun in the License file that accompanied this code."

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License Version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module.? An independent module is a module which is not derived from or based on this library.? If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so.? If you do not wish to do so, delete this exception statement from your version.

Copyright (c) 2001-2005 Sun Microsystems, Inc. All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistribution in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of Sun Microsystems, Inc. or the names of contributors may be used to endorse or promote products derived from this software without specific prior written permission.

This software is provided "AS IS," without a warranty of any kind. ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE HEREBY EXCLUDED. SUN AND ITS LICENSORS SHALL NOT BE LIABLE FOR ANY DAMAGES OR LIABILITIES

SUFFERED BY LICENSEE AS A RESULT OF OR RELATING TO USE, MODIFICATION OR DISTRIBUTION OF THE SOFTWARE OR ITS DERIVATIVES. IN NO EVENT WILL SUN OR ITS LICENSORS BE LIABLE FOR ANY LOST REVENUE, PROFIT OR DATA, OR FOR DIRECT, INDIRECT, SPECIAL, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF THE USE OF OR INABILITY TO USE SOFTWARE, EVEN IF SUN HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You acknowledge that Software is not designed, licensed or intended for use in the design, construction, operation or maintenance of any nuclear facility.

```
=====
== NOTICE file corresponding to the section 4 d of           ==
== the Apache License, Version 2.0,                          ==
== in this case for the Apache Ant distribution.              ==
=====
```

This product includes software developed by
The Apache Software Foundation (<http://www.apache.org/>).

This product includes also software developed by :
- the W3C consortium (<http://www.w3c.org/>) ,
- the SAX project (<http://www.saxproject.org/>)

Please read the different LICENSE files present in the root directory of this distribution.

```
/*
 * The Apache Software License, Version 1.1
 *
 * Copyright (c) 2001-2003 Ant-Contrib project. All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 *
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in
 * the documentation and/or other materials provided with the
 * distribution.
 *
 * 3. The end-user documentation included with the redistribution, if
 * any, must include the following acknowledgement:
 * "This product includes software developed by the
 * Ant-Contrib project (http://sourceforge.net/projects/ant-contrib/)."
```

* if and wherever such third-party acknowledgements normally appear.
 *
 * 4. The name Ant-Contrib must not be used to endorse or promote products
 * derived from this software without prior written permission. For
 * written permission, please contact
 * ant-contrib-developers@lists.sourceforge.net.
 *
 * 5. Products derived from this software may not be called "Ant-Contrib"
 * nor may "Ant-Contrib" appear in their names without prior written
 * permission of the Ant-Contrib project.
 *
 * THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED
 * WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
 * DISCLAIMED. IN NO EVENT SHALL THE ANT-CONTRIB PROJECT OR ITS
 * CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
 * LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
 * USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
 * ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
 * OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
 * OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 * =====
 */
 /*
 * \$Id: license.txt,v 1.2 2006/04/01 06:01:50 jeffsuttor Exp \$
 * %W% %E%
 */

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0 1.

Definitions.

- 1.1. Contributor means each individual or entity that creates or contributes to the creation of Modifications.
- 1.2. Contributor Version means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.
- 1.3. Covered Software means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.
- 1.4. Executable means the Covered Software in any form other than Source Code.

- 1.5. Initial Developer means the individual or entity that first makes Original Software available under this License.
- 1.6. Larger Work means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.
- 1.7. License means this document.
- 1.8. Licensable means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.
- 1.9. Modifications means the Source Code and Executable form of any of the following: A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications; B. Any new file that contains any part of the Original Software or previous Modification; or C. Any new file that is contributed or otherwise made available under the terms of this License.
- 1.10. Original Software means the Source Code and Executable form of computer software code that is originally released under this License.
- 1.11. Patent Claims means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.
- 1.12. Source Code means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.
- 1.13. You (or Your) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

- 2.1. The Initial Developer Grant. Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual

property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof);

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License;

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant. Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the

Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code. Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications. The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices. You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms. You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every

Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions. You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipients rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works. You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions. Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions. You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions. When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY. COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY. UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS. The Covered Software is a commercial item, as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of commercial computer software (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer software documentation as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS. This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdictions conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys fees and expenses. The application of the United Nations Convention on Contracts for the International Sale

of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS. As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed

as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this

License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Copyright (c) 2002-@@YEAR@@ Sun Microsystems, Inc. All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistribution in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of Sun Microsystems, Inc. or the names of contributors may be used to endorse or promote products derived from

this software without specific prior written permission.

This software is provided "AS IS," without a warranty of any kind. ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE HEREBY EXCLUDED. SUN AND ITS LICENSORS SHALL NOT BE LIABLE FOR ANY DAMAGES OR LIABILITIES SUFFERED BY LICENSEE AS A RESULT OF OR RELATING TO USE, MODIFICATION OR DISTRIBUTION OF THE SOFTWARE OR ITS DERIVATIVES. IN NO EVENT WILL SUN OR ITS LICENSORS BE LIABLE FOR ANY LOST REVENUE, PROFIT OR DATA, OR FOR DIRECT, INDIRECT, SPECIAL, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF THE USE OF OR INABILITY TO USE SOFTWARE, EVEN IF SUN HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

The CyberNeko Software License, Version 1.0

(C) Copyright 2002-2005, Andy Clark. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment:
"This product includes software developed by Andy Clark."
Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.
4. The names "CyberNeko" and "NekoHTML" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact andyc@cyberneko.net.
5. Products derived from this software may not be called "CyberNeko", nor may "CyberNeko" appear in their name, without prior written permission of the author.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES

OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR OTHER CONTRIBUTORS
BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY,
OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT
OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE
OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE,
EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====
This license is based on the Apache Software License, version 1.1.
Copyright 2001 Sun Microsystems, Inc. 901 San Antonio Road, Palo Alto,
California, 94303, U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to
technology embodied in this product. In particular, and without
limitation, these intellectual property rights may include one or more
of the U.S. patents listed at <http://www.sun.com/patents> and one or
more additional patents or pending patent applications in the U.S. and
other countries. This product is distributed under licenses
restricting its use, copying, distribution, and decompilation. No part
of this product may be reproduced in any form by any means without
prior written authorization of Sun and its licensors, if any. Third
party software, including font technology, is copyrighted and licensed
from Sun suppliers. Sun, the Sun logo, and Sun Microsystems are
trademarks or registered trademarks of Sun Microsystems, Inc. in the
U.S. and other countries. This product includes software developed by
the Apache Software Foundation (<http://www.apache.org/>). Federal
Acquisitions: Commercial Software - Government Users Subject to
Standard License Terms and Conditions.

Copyright 2001 Sun Microsystems, Inc., 901 San Antonio Road, Palo
Alto, Californie 94303 tats-Unis. Tous droits rservs. Distribue
par des licences qui en restreignent l'utilisation. Sun Microsystems,
Inc. a les droits de proprit intellectuels relatants la
technologie incorpore dans ce produit. En particulier, et sans la
limitation, ces droits de proprit intellectuels peuvent inclure un
ou plus des brevets amricains numrs <http://www.sun.com/patents>
et un ou les brevets plus supplmentaires ou les applications de
brevet en attente dans les Etats Unis et les autres pays. Ce produit
ou document est protg par un copyright et distribu avec des
licences qui en restreignent l'utilisation, la copie, la distribution,
et la dcompilation. Aucune partie de ce produit ou document ne peut
tre reproduite sous aucune forme, par quelque moyen que ce soit, sans

l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun. Sun, le logo Sun, Sun Microsystems et sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux États-Unis et dans d'autres pays. Ce produit inclut le logiciel développé par la base de Apache Software Foundation (<http://www.apache.org/>). L'accord du gouvernement des États-Unis est requis avant l'exportation du produit.

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.1

1. Definitions.

- 1.1. "Contributor" means each individual or entity that creates or contributes to the creation of Modifications.
- 1.2. "Contributor Version" means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.
- 1.3. "Covered Software" means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.
- 1.4. "Executable" means the Covered Software in any form other than Source Code.
- 1.5. "Initial Developer" means the individual or entity that first makes Original Software available under this License.
- 1.6. "Larger Work" means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.
- 1.7. "License" means this document.
- 1.8. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.
- 1.9. "Modifications" means the Source Code and Executable form of any of the following:
 - A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;
 - B. Any new file that contains any part of the Original Software or previous Modification; or
 - C. Any new file that is contributed or otherwise made available under the terms of this License.
- 1.10. "Original Software" means the Source Code and Executable form of computer software code that is originally released under this License.
- 1.11. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. "Source Code" means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or

otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the

Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Oracle is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as "Participant") alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. If You assert a patent infringement claim against Participant alleging that the Participant Software directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

6.4. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" (as that term is defined at 48 C.F.R. ? 252.227-7014(a)(1)) and "commercial computer software documentation" as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer

software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307
USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the

software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and

disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other

system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.

Copyright (C)

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author

Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

"CLASSPATH" EXCEPTION TO THE GPL VERSION 2

Certain source files distributed by Oracle are subject to the following clarification and special exception to the GPL Version 2, but only where Oracle has expressly included in the particular source file's header the words "Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the License file that accompanied this code."

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License Version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

```
/*
 *           Apache License
 *           Version 2.0, January 2004
 *           http://www.apache.org/licenses/
 *
 * TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
 *
 * 1. Definitions.
 *
 * "License" shall mean the terms and conditions for use, reproduction,
 * and distribution as defined by Sections 1 through 9 of this document.
 *
 * "Licensor" shall mean the copyright owner or entity authorized by
 * the copyright owner that is granting the License.
 *
 * "Legal Entity" shall mean the union of the acting entity and all
 * other entities that control, are controlled by, or are under common
 * control with that entity. For the purposes of this definition,
 * "control" means (i) the power, direct or indirect, to cause the
 * direction or management of such entity, whether by contract or
 * otherwise, or (ii) ownership of fifty percent (50%) or more of the
 * outstanding shares, or (iii) beneficial ownership of such entity.
 *
 * "You" (or "Your") shall mean an individual or Legal Entity
 * exercising permissions granted by this License.
 *
 * "Source" form shall mean the preferred form for making modifications,
 * including but not limited to software source code, documentation
 * source, and configuration files.
```

*
 * "Object" form shall mean any form resulting from mechanical
 * transformation or translation of a Source form, including but
 * not limited to compiled object code, generated documentation,
 * and conversions to other media types.
 *
 * "Work" shall mean the work of authorship, whether in Source or
 * Object form, made available under the License, as indicated by a
 * copyright notice that is included in or attached to the work
 * (an example is provided in the Appendix below).
 *
 * "Derivative Works" shall mean any work, whether in Source or Object
 * form, that is based on (or derived from) the Work and for which the
 * editorial revisions, annotations, elaborations, or other modifications
 * represent, as a whole, an original work of authorship. For the purposes
 * of this License, Derivative Works shall not include works that remain
 * separable from, or merely link (or bind by name) to the interfaces of,
 * the Work and Derivative Works thereof.
 *
 * "Contribution" shall mean any work of authorship, including
 * the original version of the Work and any modifications or additions
 * to that Work or Derivative Works thereof, that is intentionally
 * submitted to Licensor for inclusion in the Work by the copyright owner
 * or by an individual or Legal Entity authorized to submit on behalf of
 * the copyright owner. For the purposes of this definition, "submitted"
 * means any form of electronic, verbal, or written communication sent
 * to the Licensor or its representatives, including but not limited to
 * communication on electronic mailing lists, source code control systems,
 * and issue tracking systems that are managed by, or on behalf of, the
 * Licensor for the purpose of discussing and improving the Work, but
 * excluding communication that is conspicuously marked or otherwise
 * designated in writing by the copyright owner as "Not a Contribution."
 *
 * "Contributor" shall mean Licensor and any individual or Legal Entity
 * on behalf of whom a Contribution has been received by Licensor and
 * subsequently incorporated within the Work.
 *
 * 2. Grant of Copyright License. Subject to the terms and conditions of
 * this License, each Contributor hereby grants to You a perpetual,
 * worldwide, non-exclusive, no-charge, royalty-free, irrevocable
 * copyright license to reproduce, prepare Derivative Works of,
 * publicly display, publicly perform, sublicense, and distribute the
 * Work and such Derivative Works in Source or Object form.
 *
 * 3. Grant of Patent License. Subject to the terms and conditions of
 * this License, each Contributor hereby grants to You a perpetual,
 * worldwide, non-exclusive, no-charge, royalty-free, irrevocable
 * (except as stated in this section) patent license to make, have made,

* use, offer to sell, sell, import, and otherwise transfer the Work,
* where such license applies only to those patent claims licensable
* by such Contributor that are necessarily infringed by their
* Contribution(s) alone or by combination of their Contribution(s)
* with the Work to which such Contribution(s) was submitted. If You
* institute patent litigation against any entity (including a
* cross-claim or counterclaim in a lawsuit) alleging that the Work
* or a Contribution incorporated within the Work constitutes direct
* or contributory patent infringement, then any patent licenses
* granted to You under this License for that Work shall terminate
* as of the date such litigation is filed.

*
* 4. Redistribution. You may reproduce and distribute copies of the
* Work or Derivative Works thereof in any medium, with or without
* modifications, and in Source or Object form, provided that You
* meet the following conditions:

* (a) You must give any other recipients of the Work or
* Derivative Works a copy of this License; and

* (b) You must cause any modified files to carry prominent notices
* stating that You changed the files; and

* (c) You must retain, in the Source form of any Derivative Works
* that You distribute, all copyright, patent, trademark, and
* attribution notices from the Source form of the Work,
* excluding those notices that do not pertain to any part of
* the Derivative Works; and

* (d) If the Work includes a "NOTICE" text file as part of its
* distribution, then any Derivative Works that You distribute must
* include a readable copy of the attribution notices contained
* within such NOTICE file, excluding those notices that do not
* pertain to any part of the Derivative Works, in at least one
* of the following places: within a NOTICE text file distributed
* as part of the Derivative Works; within the Source form or
* documentation, if provided along with the Derivative Works; or,
* within a display generated by the Derivative Works, if and
* wherever such third-party notices normally appear. The contents
* of the NOTICE file are for informational purposes only and
* do not modify the License. You may add Your own attribution
* notices within Derivative Works that You distribute, alongside
* or as an addendum to the NOTICE text from the Work, provided
* that such additional attribution notices cannot be construed
* as modifying the License.

* You may add Your own copyright statement to Your modifications and
* may provide additional or different license terms and conditions

* for use, reproduction, or distribution of Your modifications, or
* for any such Derivative Works as a whole, provided Your use,
* reproduction, and distribution of the Work otherwise complies with
* the conditions stated in this License.

*
* 5. Submission of Contributions. Unless You explicitly state otherwise,
* any Contribution intentionally submitted for inclusion in the Work
* by You to the Licensor shall be under the terms and conditions of
* this License, without any additional terms or conditions.
* Notwithstanding the above, nothing herein shall supersede or modify
* the terms of any separate license agreement you may have executed
* with Licensor regarding such Contributions.

*
* 6. Trademarks. This License does not grant permission to use the trade
* names, trademarks, service marks, or product names of the Licensor,
* except as required for reasonable and customary use in describing the
* origin of the Work and reproducing the content of the NOTICE file.

*
* 7. Disclaimer of Warranty. Unless required by applicable law or
* agreed to in writing, Licensor provides the Work (and each
* Contributor provides its Contributions) on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
* implied, including, without limitation, any warranties or conditions
* of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
* PARTICULAR PURPOSE. You are solely responsible for determining the
* appropriateness of using or redistributing the Work and assume any
* risks associated with Your exercise of permissions under this License.

*
* 8. Limitation of Liability. In no event and under no legal theory,
* whether in tort (including negligence), contract, or otherwise,
* unless required by applicable law (such as deliberate and grossly
* negligent acts) or agreed to in writing, shall any Contributor be
* liable to You for damages, including any direct, indirect, special,
* incidental, or consequential damages of any character arising as a
* result of this License or out of the use or inability to use the
* Work (including but not limited to damages for loss of goodwill,
* work stoppage, computer failure or malfunction, or any and all
* other commercial damages or losses), even if such Contributor
* has been advised of the possibility of such damages.

*
* 9. Accepting Warranty or Additional Liability. While redistributing
* the Work or Derivative Works thereof, You may choose to offer,
* and charge a fee for, acceptance of support, warranty, indemnity,
* or other liability obligations and/or rights consistent with this
* License. However, in accepting such obligations, You may act only
* on Your own behalf and on Your sole responsibility, not on behalf
* of any other Contributor, and only if You agree to indemnify,
* defend, and hold each Contributor harmless for any liability

* incurred by, or claims asserted against, such Contributor by reason
* of your accepting any such warranty or additional liability.

*

* END OF TERMS AND CONDITIONS

*

* APPENDIX: How to apply the Apache License to your work.

*

* To apply the Apache License to your work, attach the following
* boilerplate notice, with the fields enclosed by brackets "[]"
* replaced with your own identifying information. (Don't include
* the brackets!) The text should be enclosed in the appropriate
* comment syntax for the file format. We also recommend that a
* file or class name and description of purpose be included on the
* same "printed page" as the copyright notice for easier
* identification within third-party archives.

*

* Copyright [yyyy] [name of copyright owner]

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.

*/

/* =====

* The Apache Software License, Version 1.1

*

* Copyright (c) 2001 The Apache Software Foundation. All rights
* reserved.

*

* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:

*

* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.

*

* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in
* the documentation and/or other materials provided with the
* distribution.

*

* 3. The end-user documentation included with the redistribution,
* if any, must include the following acknowledgment:
* "This product includes software developed by the
* Apache Software Foundation (<http://www.apache.org/>)."
* Alternately, this acknowledgment may appear in the software itself,
* if and wherever such third-party acknowledgments normally appear.

* 4. The names "Apache" and "Apache Software Foundation" and
* "Apache BCEL" must not be used to endorse or promote products
* derived from this software without prior written permission. For
* written permission, please contact apache@apache.org.

* 5. Products derived from this software may not be called "Apache",
* "Apache BCEL", nor may "Apache" appear in their name, without
* prior written permission of the Apache Software Foundation.

* THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED
* WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
* DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR
* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
* USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
* ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
* OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
* OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.

* =====
*

* This software consists of voluntary contributions made by many
* individuals on behalf of the Apache Software Foundation. For more
* information on the Apache Software Foundation, please see
* <http://www.apache.org/>.

*/
Copyright (c) 2000-2003 Daisuke Okajima and Kohsuke Kawaguchi.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment:

"This product includes software developed by Daisuke Okajima and Kohsuke Kawaguchi (<http://relaxngcc.sf.net/>)."

Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

4. The names of the copyright holders must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact the copyright holders.

5. Products derived from this software may not be called "RELAXNGCC", nor may "RELAXNGCC" appear in their name, without prior written permission of the copyright holders.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
DO NOT TRANSLATE OR LOCALIZE

%%The following software may be included in this product:

XML-NamespacesSupport

Use of any of this software is governed by the terms of the license below:

The "Artistic License"

Preamble

The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some semblance of artistic control over the development of the package, while giving the users of the package the right to use and distribute the Package in a more-or-less customary fashion, plus the right to make

reasonable modifications.

Definitions:

"Package" refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of files created through textual modification.

"Standard Version" refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright Holder as specified below.

"Copyright Holder" is whoever is named in the copyright or copyrights for the package.

"You" is you, if you're thinking about copying or distributing this Package.

"Reasonable copying fee" is whatever you can justify on the basis of media cost, duplication charges, time of people involved, and so on. (You will not be required to justify it to the Copyright Holder, but only to the computing community at large as a market that must bear the fee.)

"Freely Available" means that no fee is charged for the item itself, though there may be fees involved in handling the item. It also means that recipients of the item may redistribute it under the same conditions they received it.

1. You may make and give away verbatim copies of the source form of the Standard Version of this Package without restriction, provided that you duplicate all of the original copyright notices and associated disclaimers.
2. You may apply bug fixes, portability fixes and other modifications derived from the Public Domain or from the Copyright Holder. A Package modified in such a way shall still be considered the Standard Version.
3. You may otherwise modify your copy of this Package in any way, provided that you insert a prominent notice in each changed file stating how and when you changed that file, and provided that you do at least ONE of the following:
 - a) place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major archive site such as uunet.uu.net, or by allowing the Copyright Holder to include your modifications in the Standard Version of the Package.

b) use the modified Package only within your corporation or organization.

c) rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a separate manual page for each non-standard executable that clearly documents how it differs from the Standard Version.

d) make other distribution arrangements with the Copyright Holder.

4. You may distribute the programs of this Package in object code or executable form, provided that you do at least ONE of the following:

a) distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.

b) accompany the distribution with the machine-readable source of the Package with your modifications.

c) give non-standard executables non-standard names, and clearly document the differences in manual pages (or equivalent), together with instructions on where to get the Standard Version.

d) make other distribution arrangements with the Copyright Holder.

5. You may charge a reasonable copying fee for any distribution of this Package. You may charge any fee you choose for support of this Package. You may not charge a fee for this Package itself. However, you may distribute this Package in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution provided that you do not advertise this Package as a product of your own. You may embed this Package's interpreter within an executable of yours (by linking); this shall be construed as a mere form of aggregation, provided that the complete Standard Version of the interpreter is so embedded.

6. The scripts and library files supplied as input to or produced as output from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whoever generated them, and may be sold commercially, and may be aggregated with this Package. If such scripts or library files are aggregated with this Package via the so-called "undump" or "unexec" methods of producing a binary executable image, then distribution of such an image shall neither be construed as a distribution of this Package nor shall it fall under the restrictions of Paragraphs 3 and 4, provided that you do not represent such an executable image as a Standard Version of this Package.

7. C subroutines (or comparably compiled subroutines in other languages) supplied by you and linked into this Package in order to emulate subroutines and variables of the language defined by this Package shall not be considered part of this Package, but are the equivalent of input as in Paragraph 6, provided these subroutines do not change the language in any way that would cause it to fail the regression tests for the language.

8. Aggregation of this Package with a commercial distribution is always permitted provided that the use of this Package is embedded; that is, when no overt attempt is made to make this Package's interfaces visible to the end user of the commercial distribution. Such use shall not be construed as a distribution of this Package.

9. The name of the Copyright Holder may not be used to endorse or promote products derived from this software without specific prior written permission.

10. THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The End

Additional License(s)

Copyright (c) 2001-2005 Robin Berjon. All rights reserved.

%%The following software may be included in this product:
iso-relax.jar

Use of any of this software is governed by the terms of the license below:

The MIT License

Copyright (c)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright 200

Additional License(s)

"copyright" and "license" results in the following hits:

- > * The above copyright notice and this permission notice shall be included
- > * distribute, sublicense, and/or sell copies of the Software, and to

GNU, GPL, LGPL reveals no hit. "?" hits a lot of things but none of them are relevant to the licensing terms.

% The following software may be included in this product:

relaxngDatatype.jar

Use of any of this software is governed by the terms of the license below:

Copyright (c) 2001, Thai Open Source Software Center Ltd, Sun Microsystems.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the names of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Additional License(s)

Got the following hits. No hit for GNU, GPL, LGPL.

> Redistributions of source code must retain the above copyright

> Neither the names of the copyright holders nor the names of its

> this license is the BSD license.

%%The following software may be included in this product:

RELAX NG Object Model/Parser

Use of any of this software is governed by the terms of the license below:

The MIT License

Copyright (c)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so,

subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Additional License(s)

See <https://rngom.dev.java.net/doc/index.html>

%% The following software may be included in this product:

RelaxNGCC

Use of any of this software is governed by the terms of the license below:

Copyright (c) 2000-2003 Daisuke Okajima and Kohsuke Kawaguchi.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment:

"This product includes software developed by Daisuke Okajima and Kohsuke Kawaguchi (<http://relaxngcc.sf.net/>)."

Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

4. The names of the copyright holders must not be used to endorse or promote products derived from this software without prior written

permission. For written permission, please contact the copyright holders.

5. Products derived from this software may not be called "RELAXNGCC", nor may "RELAXNGCC" appear in their name, without prior written permission of the copyright holders.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Additional License(s)

None found

% The following software may be included in this product:

XML Resolver library

Use of any of this software is governed by the terms of the license below:

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the

direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of

this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and

wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor

has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Additional License(s)

All occurrences of copyright, license and (c) refer to the Apache 1.1 license.

No occurrences of GNU, GPL, LGPL.

%%The following software may be included in this product:

Stax API (only)

Use of any of this software is governed by the terms of the license below:

Streaming API for XML (JSR-173) Specification

Reference Implementation

License Agreement

READ THE TERMS OF THIS (THE "AGREEMENT") CAREFULLY BEFORE VIEWING OR USING THE SOFTWARE LICENS
ED HEREUNDER. BY VIEWING OR USING THE SOFTWARE, YOU AGREE TO THE TERMS OF THIS AGREEMENT. IF
YOU ARE ACCESSING THE SOFTWARE ELECTRONICALLY, INDICATE YOUR ACCEPTANCE OF THESE
TERMS BY SELE
CTING THE "ACCEPT" BUTTON AT THE END OF THIS AGREEMENT. IF YOU DO NOT AGREE TO ALL THESE TERMS
, PROMPTLY RETURN THE UNUSED SOFTWARE TO ORIGINAL CONTRIBUTOR, DEFINED HEREIN.

1.0 DEFINITIONS.

1.1. "BEA" means BEA Systems, Inc., the licensor of the Original Code.

1.2. "Contributor" means BEA and each entity that creates or contributes to the creation of Mo
difications.

1.3. "Covered Code" means the Original Code or Modifications or the combination of the Origina
l Code and Modifications, in each case including portions thereof and corresponding documentat
ion released with the source code.

1.4. "Executable" means Covered Code in any form other than Source Code.

1.5. "FCS" means first commercial shipment of a product.

1.6. "Modifications" means any addition to or deletion from the substance or structure of eith
er the Original Code or any previous Modifications. When Covered Code is released as a series
of files, a Modification is:

(a) Any addition to or deletion from the contents of a file containing Original Code or previ
ous Modifications.

(b) Any new file that contains any part of the Original Code or previous Modifications.

1.7. "Original Code" means Source Code of computer software code Reference Implementation.

1.8. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent for which the grantor has the right to grant a license.

1.9. "Reference Implementation" means the prototype or "proof of concept" implementation of the Specification developed and made available for license by or on behalf of BEA.

1.10. "Source Code" means the preferred form of the Covered Code for making modifications to it, including all modules it contains, plus any associated documentation, interface definition files, scripts used to control compilation and installation of an Executable, or source code differential comparisons against either the Original Code or another well known, available Covered Code of the Contributor's choice.

1.11. "Specification" means the written specification for the Streaming API for XML, Java technology developed pursuant to the Java Community Process.

1.12. "Technology Compatibility Kit" or "TCK" means the documentation, testing tools and test suites associated with the Specification as may be revised by BEA from time to time, that is provided so that an implementer of the Specification may determine if its implementation is compliant with the Specification.

1.13. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this Agreement or a future version of this Agreement issued under Section 6.1. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the

outstanding shares or beneficial ownership of such entity.

2.0 SOURCE CODE LICENSE.

2.1. Copyright Grant. Subject to the terms of this Agreement, each Contributor hereby grants

You a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Covered Code of such Contributor, if any, and such derivative works, in Source Code and Executable form.

2.2. Patent Grant. Subject to the terms of this Agreement, each Contributor hereby grants You

a non-exclusive, worldwide, royalty-free patent license under the Patent Claims to make, use, sell, offer to sell, import and otherwise transfer the Covered Code prepared and provided by such Contributor, if any, in Source Code and Executable form. This patent license shall apply to the Covered Code if, at the time a Modification is added by the Contributor, such addition of the Modification causes such combination to be covered by the Patent Claims. The patent license shall not apply to any other combinations which include the Modification.

2.3. Conditions to Grants. You understand that although each Contributor grants the licenses

to the Covered Code prepared by it, no assurances are provided by any Contributor that the Covered Code does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to You for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, You hereby assume sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow You to distribute Covered Code, it is Your responsibility to acquire that license before distributing such code.

2.4. Contributors' Representation. Each Contributor represents that to its knowledge it has

sufficient copyright rights in the Covered Code it provides , if any, to grant the copyright license set forth in this Agreement.

3.0 DISTRIBUTION RESTRICTIONS.

3.1. Application of Agreement.

The Modifications which You create or to which You contribute are governed by the terms of this Agreement, including without limitation Section 2.0. The Source Code version of Covered Code may be distributed only under the terms of this Agreement or a future version of this Agreement released under Section 6.1, and You must include a copy of this Agreement with every copy of the Source Code You distribute. You may not offer or impose any terms on any Source Code version that alters or restricts the applicable version of this Agreement or the recipients' rights hereunder. However, You may include an additional document offering the additional rights described in Section 3.3.

3.2. Description of Modifications.

You must cause all Covered Code to which You contribute to contain a file documenting the changes You made to create that Covered Code and the date of any change. You must include a prominent statement that the Modification is derived, directly or indirectly, from Original Code provided by BEA and including the name of BEA in (a) the Source Code, and (b) in any notice in an Executable version or related documentation in which You describe the origin or ownership of the Covered Code.

%%The following software may be included in this product:

XMLWriter

Use of any of this software is governed by the terms of the license below:

XMLWriter IS FREE

I hereby abandon any property rights to XMLWriter 0.1, and release all of the XMLWriter 0.1 source code, compiled code, and documentation contained in this distribution into the Public Domain. XMLWriter comes with NO WARRANTY or guarantee of fitness for any purpose.

David Megginson
david@megginson.com
2000-04-19

Additional License(s)

I grep-ed the source. GNU and GPL has no hits, '?' yields 11 hits but none of them are license related. "copyright" and "license" yield no hits either.

/*

* The Apache Software License, Version 1.1

*

*

* Copyright (c) 1999-2004 The Apache Software Foundation. All rights reserved.

*

* Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

*

* 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

*

* 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

*

* 3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment:

* "This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>)."

* Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

*

* 4. The names "Xerces" and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org.

*

* 5. Products derived from this software may not be called "Apache",

* nor may "Apache" appear in their name, without prior written
* permission of the Apache Software Foundation.
*
* THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED
* WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
* DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR
* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
* USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
* ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
* OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
* OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.

* =====

*
* This software consists of voluntary contributions made by many
* individuals on behalf of the Apache Software Foundation and was
* originally based on software copyright (c) 1999, International
* Business Machines, Inc., <http://www.ibm.com>. For more
* information on the Apache Software Foundation, please see
* <http://www.apache.org/>.

*/

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

1. Definitions.

1.1. Contributor means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable means the Covered Software in any form other than Source Code.

1.5. Initial Developer means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License means this document.

1.8. Licensable means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You (or Your) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either

alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipients rights in the Source Code form from the rights set forth in this License. If You distribute the

Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms

herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a commercial item, as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of commercial computer software (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer software documentation as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software

(except to the extent applicable law, if any, provides otherwise), excluding such jurisdictions conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The GlassFish code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

xml-commons/LICENSE.txt \$Id: LICENSE.txt 226068 2003-07-06 03:27:45Z crossley \$

See README.txt for additional licensing information.

/* =====

* The Apache Software License, Version 1.1

*

* Copyright (c) 2001-2003 The Apache Software Foundation. All rights

* reserved.

*

* Redistribution and use in source and binary forms, with or without

* modification, are permitted provided that the following conditions

* are met:

*

* 1. Redistributions of source code must retain the above copyright

* notice, this list of conditions and the following disclaimer.

*

* 2. Redistributions in binary form must reproduce the above copyright

* notice, this list of conditions and the following disclaimer in

* the documentation and/or other materials provided with the

* distribution.

*

* 3. The end-user documentation included with the redistribution,

* if any, must include the following acknowledgment:

* "This product includes software developed by the

* Apache Software Foundation (<http://www.apache.org/>)."

* Alternately, this acknowledgment may appear in the software itself,
 * if and wherever such third-party acknowledgments normally appear.
 *

* 4. The names "Apache" and "Apache Software Foundation" must
 * not be used to endorse or promote products derived from this
 * software without prior written permission. For written
 * permission, please contact apache@apache.org.
 *

* 5. Products derived from this software may not be called "Apache",
 * nor may "Apache" appear in their name, without prior written
 * permission of the Apache Software Foundation.
 *

* THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED
 * WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
 * DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR
 * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
 * LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
 * USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
 * ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
 * OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
 * OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 * =====
 *

* This software consists of voluntary contributions made by many
 * individuals on behalf of the Apache Software Foundation. For more
 * information on the Apache Software Foundation, please see
 * <http://www.apache.org/>.
 */

1.9 micronaut-kubernetes 3.4.0

1.9.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2017-2021 original authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
```

- * distributed under the License is distributed on an "AS IS" BASIS,
- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.
- */

Found in path(s):

- * /opt/cola/permits/1331474306_1653514752.2550623/0/micronaut-kubernetes-client-3-4-0-sources-jar/io/micronaut/kubernetes/client/PodNameResolver.java
 - * /opt/cola/permits/1331474306_1653514752.2550623/0/micronaut-kubernetes-client-3-4-0-sources-jar/io/micronaut/kubernetes/client/NamespaceResolver.java
 - * /opt/cola/permits/1331474306_1653514752.2550623/0/micronaut-kubernetes-client-3-4-0-sources-jar/io/micronaut/kubernetes/client/Apis.java
 - * /opt/cola/permits/1331474306_1653514752.2550623/0/micronaut-kubernetes-client-3-4-0-sources-jar/io/micronaut/kubernetes/client/OkHttpClientLogging.java
 - * /opt/cola/permits/1331474306_1653514752.2550623/0/micronaut-kubernetes-client-3-4-0-sources-jar/io/micronaut/kubernetes/client/ApisAutomaticFeatureMetadata.java
 - * /opt/cola/permits/1331474306_1653514752.2550623/0/micronaut-kubernetes-client-3-4-0-sources-jar/io/micronaut/kubernetes/client/ApiClientConfiguration.java
 - * /opt/cola/permits/1331474306_1653514752.2550623/0/micronaut-kubernetes-client-3-4-0-sources-jar/io/micronaut/kubernetes/client/DefaultNamespaceResolver.java
 - * /opt/cola/permits/1331474306_1653514752.2550623/0/micronaut-kubernetes-client-3-4-0-sources-jar/io/micronaut/kubernetes/client/ApiClientFactory.java
 - * /opt/cola/permits/1331474306_1653514752.2550623/0/micronaut-kubernetes-client-3-4-0-sources-jar/io/micronaut/kubernetes/client/DiscoveryFactory.java
 - * /opt/cola/permits/1331474306_1653514752.2550623/0/micronaut-kubernetes-client-3-4-0-sources-jar/io/micronaut/kubernetes/client/graalvm/KubernetesClientFeature.java
 - * /opt/cola/permits/1331474306_1653514752.2550623/0/micronaut-kubernetes-client-3-4-0-sources-jar/io/micronaut/kubernetes/client/ModelMapper.java
 - * /opt/cola/permits/1331474306_1653514752.2550623/0/micronaut-kubernetes-client-3-4-0-sources-jar/io/micronaut/kubernetes/client/DefaultPodNameResolver.java
- No license file was found, but licenses were detected in source scan.

/*

- * Copyright 2017-2022 original authors
- *
- * Licensed under the Apache License, Version 2.0 (the "License");
- * you may not use this file except in compliance with the License.
- * You may obtain a copy of the License at
- *
- * <https://www.apache.org/licenses/LICENSE-2.0>
- *
- * Unless required by applicable law or agreed to in writing, software
- * distributed under the License is distributed on an "AS IS" BASIS,
- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.
- */

Found in path(s):

* /opt/cola/permits/1331474306_1653514752.2550623/0/micronaut-kubernetes-client-3-4-0-sources-jar/io/micronaut/kubernetes/client/DiscoveryCache.java

1.10 checker-qual 3.12.0

1.10.1 Available under license :

Checker Framework qualifiers

Copyright 2004-present by the Checker Framework developers

MIT License:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.11 swagger-models 2.2.3

1.11.1 Available under license :

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise

designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. **Grant of Copyright License.** Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. **Grant of Patent License.** Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. **Redistribution.** You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must

include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly

negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright (c) 2015. SmartBear Software Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Swagger Core - swagger-models

Copyright (c) 2015. SmartBear Software Inc.

Swagger Core - swagger-models is licensed under Apache 2.0 license.
Copy of the Apache 2.0 license can be found in `LICENSE` file.

1.12 apache-httpmime 4.5.13

1.12.1 Available under license :

Apache HttpClient Mime
Copyright 1999-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a

cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,

any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.13 commons-logging 1.2

1.13.1 Available under license :

/*

- * Licensed to the Apache Software Foundation (ASF) under one or more
- * contributor license agreements. See the NOTICE file distributed with
- * this work for additional information regarding copyright ownership.
- * The ASF licenses this file to You under the Apache License, Version 2.0
- * (the "License"); you may not use this file except in compliance with
- * the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

- * Unless required by applicable law or agreed to in writing, software
- * distributed under the License is distributed on an "AS IS" BASIS,
- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.

*/

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner

or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. **Grant of Copyright License.** Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. **Grant of Patent License.** Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. **Redistribution.** You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions

of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Apache Commons Logging

Copyright 2003-2014 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

1.14 okio 2.8.0

1.14.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2015 Square, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/SegmentedByteString.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/SegmentedByteString.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/ForwardingTimeout.kt
No license file was found, but licenses were detected in source scan.
```

```
/*
 * Copyright (C) 2019 Square, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/internal/RealBufferedSource.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/internal/RealBufferedSink.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/internal/SegmentedByteString.kt
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2017 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/Utf8.kt
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2019 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software

- * distributed under the License is distributed on an "AS IS" BASIS,
- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.
- */

Found in path(s):

- * /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/internal/Buffer.kt
- * /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/Timeout.kt
- * /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/BufferedSource.kt
- * /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/Sink.kt
- * /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/Buffer.kt
- * /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/RealBufferedSink.kt
- * /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/BufferedSink.kt
- * /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/Source.kt
- * /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/RealBufferedSource.kt
- * /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/Okio.kt

No license file was found, but licenses were detected in source scan.

/*

- * Copyright (C) 2014 Square, Inc.
- *
- * Licensed under the Apache License, Version 2.0 (the "License");
- * you may not use this file except in compliance with the License.
- * You may obtain a copy of the License at
- *

* <http://www.apache.org/licenses/LICENSE-2.0>

*

- * Unless required by applicable law or agreed to in writing, software
- * distributed under the License is distributed on an "AS IS" BASIS,
- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.

*/

Found in path(s):

- * /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/SegmentPool.kt
- * /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/InflaterSource.kt
- * /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/ForwardingSource.kt
- * /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/BufferedSink.kt
- * /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/JvmOkio.kt
- * /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/Segment.kt
- * /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/BufferedSource.kt
- * /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/GzipSource.kt
- * /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/DeflaterSink.kt
- * /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/GzipSink.kt
- * /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/Buffer.kt

* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/Source.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/Sink.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/Timeout.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/RealBufferedSource.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/SegmentPool.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/RealBufferedSink.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/AsyncTimeout.kt
* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/ForwardingSink.kt

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2018 Square, Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/-Util.kt

* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/internal/ByteString.kt

* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/-Platform.kt

* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/-Platform.kt

* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/internal/-Utf8.kt

* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/ByteString.kt

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2014 Square Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/ByteString.kt

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2018 Square, Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/-DeprecatedUpgrade.kt

* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/-DeprecatedUtf8.kt

* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/PeekSource.kt

* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/-DeprecatedOkio.kt

* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/Throttler.kt

No license file was found, but licenses were detected in source scan.

/*

* Licensed to the Apache Software Foundation (ASF) under one or more

* contributor license agreements. See the NOTICE file distributed with

* this work for additional information regarding copyright ownership.

* The ASF licenses this file to You under the Apache License, Version 2.0

* (the "License"); you may not use this file except in compliance with

* the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/-Base64.kt

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2016 Square, Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/HashingSink.kt

* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/Pipe.kt

* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/jvmMain/okio/HashingSource.kt

* /opt/cola/permits/1274700344_1645235023.94/0/okio-2-8-0-sources-jar/commonMain/okio/Options.kt

1.15 javabeans-activation-framework-api 1.2.2

1.15.1 Available under license :

Copyright (c) 2018 Oracle and/or its affiliates. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the Eclipse Foundation, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

[subs="normal"]

....

Specification: {doctitle}

Version: {revnumber}

ifeval::["{revremark}" != ""]

Status: {revremark}

endif::[]

ifeval::["{revremark}" == ""]

Status: Final Release

endif::[]

Release: {revdate}

....

Copyright (c) 2019 Eclipse Foundation.

=== Eclipse Foundation Specification License

By using and/or copying this document, or the Eclipse Foundation document from which this statement is linked, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions:

Permission to copy, and distribute the contents of this document, or the Eclipse Foundation document from which this statement is linked, in any medium for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the document, or portions thereof, that you use:

- * link or URL to the original Eclipse Foundation document.
- * All existing copyright notices, or if one does not exist, a notice (hypertext is preferred, but a textual representation is permitted) of the form: "Copyright (c) [\$date-of-document] Eclipse Foundation, Inc. <<url to this license>>"

Inclusion of the full text of this NOTICE must be provided. We request that authorship attribution be provided in any software, documents, or other items or products that you create pursuant to the implementation of the contents of this document, or any portion thereof.

No right to create modifications or derivatives of Eclipse Foundation documents is granted pursuant to this license, except anyone may prepare and distribute derivative works and portions of this document in software that implements the specification, in supporting materials accompanying such software, and in documentation of such software, PROVIDED that all such works include the notice below. HOWEVER, the publication of derivative works of this document for use as a technical specification is expressly prohibited.

The notice is:

"Copyright (c) 2018 Eclipse Foundation. This software or document includes material copied from or derived from [title and URI of the Eclipse Foundation specification document]."

==== Disclaimers

THIS DOCUMENT IS PROVIDED "AS IS," AND THE COPYRIGHT HOLDERS AND THE ECLIPSE FOUNDATION MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THE DOCUMENT ARE SUITABLE FOR ANY PURPOSE; NOR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

THE COPYRIGHT HOLDERS AND THE ECLIPSE FOUNDATION WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE DOCUMENT OR THE PERFORMANCE OR IMPLEMENTATION OF THE CONTENTS THEREOF.

The name and trademarks of the copyright holders or the Eclipse Foundation may NOT be used in advertising or publicity pertaining to this document or its contents without specific, written prior permission. Title to copyright in this document will at all times remain with copyright holders.

Notices for Jakarta Activation

This content is produced and maintained by Jakarta Activation project.

* Project home: <https://projects.eclipse.org/projects/ee4j.jaf>

Copyright

All content is the property of the respective authors or their employers. For more information regarding authorship of content, please consult the listed source code repository logs.

Declared Project Licenses

This program and the accompanying materials are made available under the terms of the Eclipse Distribution License v. 1.0, which is available at <http://www.eclipse.org/org/documents/edl-v10.php>.

SPDX-License-Identifier: BSD-3-Clause

Source Code

The project maintains the following source code repositories:

* <https://github.com/eclipse-ee4j/jaf>

Third-party Content

This project leverages the following third party content.

JUnit (4.12)

* License: Eclipse Public License

1.16 logback-core 1.2.10

1.16.1 Available under license :

Found license 'Eclipse Public License 1.0' in '* Copyright (C) 1999-2015, QOS.ch. All rights reserved. * This program and the accompanying materials are dual-licensed under * either the terms of the Eclipse Public License v1.0 as published by * under the terms of the GNU Lesser General Public License version 2.1 * as published by the Free Software Foundation.'

Found license 'GNU Lesser General Public License' in '* Copyright (C) 1999-2015, QOS.ch. All rights reserved. * This program and the accompanying materials are dual-licensed under * either the terms of the Eclipse Public License v1.0 as published by * under the terms of the GNU Lesser General Public License version 2.1 * as published by the Free Software Foundation.'

1.17 apache-kafka 3.3.1

1.17.1 Available under license :

Apache Kafka

Copyright 2022 The Apache Software Foundation.

This product includes software developed at
The Apache Software Foundation (<https://www.apache.org/>).

This distribution has a binary dependency on jersey, which is available under the CDDL License. The source code of jersey can be found at <https://github.com/jersey/jersey/>.

This distribution has a binary test dependency on jqwik, which is available under the Eclipse Public License 2.0. The source code can be found at <https://github.com/jlink/jqwik>.

The streams-scala (streams/streams-scala) module was donated by Lightbend and the original code was copyrighted by them:

Copyright (C) 2018 Lightbend Inc. <<https://www.lightbend.com>>

Copyright (C) 2017-2018 Alexis Seigneurin.

This project contains the following code copied from Apache Hadoop:

`clients/src/main/java/org/apache/kafka/common/utils/PureJavaCrc32C.java`

Some portions of this file Copyright (c) 2004-2006 Intel Corporation and licensed under the BSD license.

This project contains the following code copied from Apache Hive:

`streams/src/main/java/org/apache/kafka/streams/state/internals/Murmur3.java`

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,

including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf

of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.18 feign-apache-httpclient 10.12

1.18.1 Available under license :

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0

Bundle-Description: Feign Apache HttpClient

Bundle-License: <http://www.apache.org/licenses/LICENSE-2.0.txt>

Bundle-SymbolicName: io.github.openfeign.feign-httpclient

Built-By: circleci

Bnd-LastModified: 1609190915103

Bundle-ManifestVersion: 2

Bundle-DocURL: <https://github.com/openfeign>

Bundle-Vendor: OpenFeign

Import-Package: feign;version="[10.12,11)",org.apache.http.org.apache.

```
http.client,org.apache.http.client.config,org.apache.http.client.methods,org.apache.http.client.utils,org.apache.http.entity,org.apache.http.impl.client,org.apache.http.util
Require-Capability: osgi.ee;filter="(&(osgi.ee=JavaSE)(version=1.8))"
Tool: Bnd-4.0.0.201805111645
Export-Package: feign.httpclient;uses:="feign,org.apache.http.client";
version="10.12.0"
Bundle-Name: Feign Apache HttpClient
Bundle-Version: 10.12.0
Created-By: Apache Maven Bundle Plugin
Build-Jdk: 1.8.0_265
```

Found in path(s):

```
* /opt/cola/permits/1418559534_1663305744.5909555/0/feign-httpclient-10-12-jar/META-INF/MANIFEST.MF
```

No license file was found, but licenses were detected in source scan.

<!--

Copyright 2012-2020 The Feign Authors

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express

or implied. See the License for the specific language governing permissions and limitations under the License.

-->

Found in path(s):

```
* /opt/cola/permits/1418559534_1663305744.5909555/0/feign-httpclient-10-12-jar/META-INF/maven/io.github.openfeign/feign-httpclient/pom.xml
```

1.19 jackson-databind 2.14.0

1.19.1 Available under license :

Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library. It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has been in development since 2007.

It is currently developed by a community of developers.

Licensing

Jackson 2.x core and extension components are licensed under Apache License 2.0
To find the details that apply to this artifact see the accompanying LICENSE file.

Credits

A list of contributors may be found from CREDITS(-2.x) file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work

(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses

granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]"

replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.20 fabric8-::-kubernetes-model-::-discovery

4.13.3

1.20.1 Available under license :

No license file was found, but licenses were detected in source scan.

*

* Copyright (C) 2015 Red Hat, Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*

Found in path(s):

* /opt/cola/permits/1288520128_1647861750.66/0/kubernetes-model-discovery-4-13-3-jar/manifest.vm

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0
Bnd-LastModified: 1619068579958
Build-Jdk-Spec: 1.8
Bundle-Description: Java client for Kubernetes and OpenShift
Bundle-DocURL: <http://redhat.com>
Bundle-License: <http://www.apache.org/licenses/LICENSE-2.0.txt>
Bundle-ManifestVersion: 2
Bundle-Name: Fabric8 :: Kubernetes Model :: Discovery
Bundle-SymbolicName: io.fabric8.kubernetes-model-discovery
Bundle-Vendor: Red Hat
Bundle-Version: 4.13.3
Created-By: Apache Maven Bundle Plugin
Export-Package: io.fabric8.kubernetes.api.model.discovery;uses:="com.fasterxml.jackson.annotation,com.fasterxml.jackson.databind,com.fasterxml.jackson.databind.annotation,io.fabric8.kubernetes.api.builder,io.fabric8.kubernetes.api.model,io.fabric8.kubernetes.model.annotation";version="4.13.3"
Implementation-Title: Fabric8 :: Kubernetes Model :: Discovery
Implementation-Vendor: Red Hat
Implementation-Version: 4.13.3
Import-Package: com.fasterxml.jackson.annotation;version="[2.11,3)",com.fasterxml.jackson.databind;version="[2.11,3)",com.fasterxml.jackson.databind.annotation;version="[2.11,3)",io.fabric8.kubernetes.api.builder;version="[4.13,5)",io.fabric8.kubernetes.api.model,io.fabric8.kubernetes.model.annotation;version="[4.13,5)"
Require-Capability: osgi.ee;filter="(&(osgi.ee=JavaSE)(version=1.8))"
Specification-Title: Fabric8 :: Kubernetes Model :: Discovery
Specification-Vendor: Red Hat
Specification-Version: 4.13
Tool: Bnd-5.1.1.202006162103

Found in path(s):

* /opt/cola/permits/1288520128_1647861750.66/0/kubernetes-model-discovery-4-13-3-jar/META-INF/MANIFEST.MF

No license file was found, but licenses were detected in source scan.

Copyright (C) 2015 Red Hat, Inc.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE>

2.0

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Found in path(s):

* /opt/cola/permits/1288520128_1647861750.66/0/kubernetes-model-discovery-4-13-3-jar/META-INF/maven/io.fabric8/kubernetes-model-discovery/pom.xml

1.21 junit-5-bill-of-materials 5.9.1

1.21.1 Available under license :

```
import java.io.File
import java.net.URI
```

```
data class License(val name: String, val url: URI, val headerFile: File)
```

```
Apache License
```

```
=====
```

```
_Version 2.0, January 2004_
```

```
_&lt;<https://www.apache.org/licenses/>&gt;_
```

```
### Terms and Conditions for use, reproduction, and distribution
```

```
#### 1. Definitions
```

License shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

Licensor shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

Legal Entity shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, control means **(i)** the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or **(ii)** ownership of fifty percent (50%) or more of the outstanding shares, or **(iii)** beneficial ownership of such entity.

You (or Your) shall mean an individual or Legal Entity exercising permissions granted by this License.

Source form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

Object form shall mean any form resulting from mechanical transformation or

translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

Work shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

Derivative Works shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

Contribution shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, submitted means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as Not a Contribution.

Contributor shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was

submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution

You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- * **(a)** You must give any other recipients of the Work or Derivative Works a copy of this License; and
- * **(b)** You must cause any modified files to carry prominent notices stating that You changed the files; and
- * **(c)** You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- * **(d)** If the Work includes a NOTICE text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions

Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks

This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty

Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an AS IS BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability

While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

Eclipse Public License - v 2.0

=====

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE (AGREEMENT). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. Definitions

Contribution means:

- * **a)** in the case of the initial Contributor, the initial content Distributed under this Agreement, and
- * **b)** in the case of each subsequent Contributor:
 - * **i)** changes to the Program, and
 - * **ii)** additions to the Program;

where such changes and/or additions to the Program originate from and are Distributed by that particular Contributor. A Contribution originates from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include changes or additions to the Program that are not Modified Works.

Contributor means any person or entity that Distributes the Program.

Licensed Patents mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

Program means the Contributions Distributed in accordance with this Agreement.

Recipient means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

Derivative Works shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

Modified Works shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

Distribute means the acts of **a)** distributing or **b)** making available in any manner that enables the transfer of a copy.

Source Code means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

Secondary License means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.

2. Grant of Rights

* **a)** Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

* **b)** Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive,

worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

****c)**** Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

****d)**** Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

****e)**** Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

3. Requirements

****3.1)**** If a Contributor Distributes the Program in any form, then:

* ****a)**** the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and

* ****b)**** the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:

* ****i)**** effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

* ****ii)**** effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

* ****iii)**** does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and

* ****iv)**** requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.

****3.2)**** When the Program is Distributed as Source Code:

* ****a)**** it must be made available under this Agreement, or if the Program ****i)**** is combined with other material in a separate file or files made available under a Secondary License, and ****ii)**** the initial Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available under the terms of such Secondary Licenses, and

* ****b)**** a copy of this Agreement must be included with each copy of the Program.

****3.3**** Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability (notices) contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

4. Commercial Distribution

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor (Commercial Contributor) hereby agrees to defend and indemnify every other Contributor (Indemnified Contributor) against any losses, damages and costs (collectively Losses) arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: ****a)**** promptly notify the Commercial Contributor in writing of such claim, and ****b)**** allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. No Warranty

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN AS IS BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. Disclaimer of Liability

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF

SUCH DAMAGES.

7. General

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.

Exhibit A - Form of Secondary Licenses Notice

> This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: {name license(s), version(s), and exceptions or additional permissions here}.

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Apache License

=====

Version 2.0, January 2004

<<<https://www.apache.org/licenses/>>>>

Terms and Conditions for use, reproduction, and distribution

1. Definitions

License shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

Licensor shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

Legal Entity shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, control means **(i)** the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or **(ii)** ownership of fifty percent (50%) or more of the outstanding shares, or **(iii)** beneficial ownership of such entity.

You (or Your) shall mean an individual or Legal Entity exercising permissions granted by this License.

Source form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

Object form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

Work shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

Derivative Works shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

Contribution shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work

by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, submitted means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as Not a Contribution.

Contributor shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution

You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- * **(a)** You must give any other recipients of the Work or Derivative Works a copy of this License; and
- * **(b)** You must cause any modified files to carry prominent notices stating that You changed the files; and
- * **(c)** You must retain, in the Source form of any Derivative Works that You distribute,

all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

* **(d)** If the Work includes a NOTICE text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions

Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks

This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty

Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an AS IS BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of

permissions under this License.

8. Limitation of Liability

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability

While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets `[]` replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same printed page as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<https://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

Open Source Licenses

=====

This product may include a number of subcomponents with separate copyright notices and license terms. Your use of the source code for these subcomponents is subject to the terms and conditions of the subcomponent's license, as noted in the LICENSE-<subcomponent>.md files.

[[contributors]]

== Contributors

Browse the {junit5-repo}/graphs/contributors[current list of contributors] directly on GitHub.

1.22 jakarta-dependency-injection 2.0.1

1.22.1 Available under license :

Notices for Eclipse Jakarta Dependency Injection

This content is produced and maintained by the Eclipse Jakarta Dependency Injection project.

* Project home: <https://projects.eclipse.org/projects/cdi.batch>

Trademarks

Jakarta Dependency Injection is a trademark of the Eclipse Foundation.

Copyright

All content is the property of the respective authors or their employers. For more information regarding authorship of content, please consult the listed source code repository logs.

Declared Project Licenses

This program and the accompanying materials are made available under the terms of the Apache License, Version 2.0 which is available at <https://www.apache.org/licenses/LICENSE-2.0>.

SPDX-License-Identifier: Apache-2.0

Source Code

The project maintains the following source code repositories:

<https://github.com/eclipse-ee4j/injection-api>

<https://github.com/eclipse-ee4j/injection-spec>

<https://github.com/eclipse-ee4j/injection-tck>

Third-party Content

This project leverages the following third party content.

None

Cryptography

None

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or

Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work

or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work

by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.23 jackson-jaxrs 2.14.0

1.23.1 Available under license :

This copy of Jackson JSON processor databind module is licensed under the Apache (Software) License, version 2.0 ("the License"). See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

<http://www.apache.org/licenses/LICENSE-2.0>

Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library. It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has been in development since 2007.

It is currently developed by a community of developers, as well as supported commercially by FasterXML.com.

Licensing

Jackson core and extension components may be licensed under different licenses. To find the details that apply to this artifact see the accompanying LICENSE file. For more information, including possible other licensing options, contact

FasterXML.com (<http://fasterxml.com>).

Credits

A list of contributors may be found from CREDITS file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.

1.24 apache-commons-lang 3.9

1.24.1 Available under license :

No license file was found, but licenses were detected in source scan.

The ASF licenses this file to You under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.apache.org/licenses/LICENSE-2.0> distributed under the License is distributed on an "AS IS" BASIS,

Found in path(s):

* /opt/cola/permits/1062991095_1611209694.78/0/apache-commons-lang-pom-zip/apache-commons-lang.pom.rtf

1.25 guava-internalfuturefailureaccess-and-internalfutures 1.0.1

1.25.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2018 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License

* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express

* or implied. See the License for the specific language governing permissions and limitations under

* the License.

*/

Found in path(s):

* /opt/cola/permits/1130987386_1612872111.26/0/failureaccess-1-0-1-sources-

jar/com/google/common/util/concurrent/internal/InternalFutureFailureAccess.java
* /opt/cola/permits/1130987386_1612872111.26/0/failureaccess-1-0-1-sources-
jar/com/google/common/util/concurrent/internal/InternalFutures.java

1.26 fabric8-::-kubernetes-model-::-settings

4.13.3

1.26.1 Available under license :

No license file was found, but licenses were detected in source scan.

Copyright (C) 2015 Red Hat, Inc.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE>

2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Found in path(s):

* /opt/cola/permits/1288519918_1647861426.57/0/kubernetes-model-settings-4-13-3-jar/META-
INF/maven/io.fabric8/kubernetes-model-settings/pom.xml

No license file was found, but licenses were detected in source scan.

*

* Copyright (C) 2015 Red Hat, Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.

*

Found in path(s):

* /opt/cola/permits/1288519918_1647861426.57/0/kubernetes-model-settings-4-13-3-jar/manifest.vm

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0

Bnd-LastModified: 1619068658143

Build-Jdk-Spec: 1.8

Bundle-Description: Java client for Kubernetes and OpenShift

Bundle-DocURL: http://redhat.com

Bundle-License: http://www.apache.org/licenses/LICENSE-2.0.txt

Bundle-ManifestVersion: 2

Bundle-Name: Fabric8 :: Kubernetes Model :: Settings

Bundle-SymbolicName: io.fabric8.kubernetes-model-settings

Bundle-Vendor: Red Hat

Bundle-Version: 4.13.3

Created-By: Apache Maven Bundle Plugin

Export-Package: io.fabric8.kubernetes.api.model.settings;uses:="com.fasterxml.jackson.annotation,com.fasterxml.jackson.databind,com.fasterxml.jackson.databind.annotation,io.fabric8.kubernetes.api.builder,io.fabric8.kubernetes.api.model,io.fabric8.kubernetes.model.annotation";version="4.13.3"

Implementation-Title: Fabric8 :: Kubernetes Model :: Settings

Implementation-Vendor: Red Hat

Implementation-Version: 4.13.3

Import-Package: com.fasterxml.jackson.annotation;version="[2.11,3)",com.fasterxml.jackson.databind;version="[2.11,3)",com.fasterxml.jackson.databind.annotation;version="[2.11,3)",io.fabric8.kubernetes.api.builder;version="[4.13,5)",io.fabric8.kubernetes.api.model,io.fabric8.kubernetes.model.annotation;version="[4.13,5)"

Require-Capability: osgi.ee;filter="(&(osgi.ee=JavaSE)(version=1.8))"

Specification-Title: Fabric8 :: Kubernetes Model :: Settings

Specification-Vendor: Red Hat

Specification-Version: 4.13

Tool: Bnd-5.1.1.202006162103

Found in path(s):

* /opt/cola/permits/1288519918_1647861426.57/0/kubernetes-model-settings-4-13-3-jar/META-INF/MANIFEST.MF

1.27 micronaut-security 3.8.0

1.27.1 Available under license :

Apache License

Version 2.0, January 2004

<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and

attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the

appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.28 opentelemetry-java---

io.opentelemetry:opentelemetry-sdk-metrics

1.9.1-alpha

1.28.1 Available under license :

```
/*  
 * Copyright The OpenTelemetry Authors  
 * SPDX-License-Identifier: Apache-2.0  
 */  
Apache License  
    Version 2.0, January 2004  
    http://www.apache.org/licenses/
```

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but

not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their

Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with

the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.29 jetbrains-kotlin-kotlin-stdlib-jdk8 1.6.21

1.29.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2010-2017 JetBrains s.r.o.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

- * /opt/cola/permits/1343419559_1655661587.9846208/0/kotlin-stdlib-jdk8-1-6-21-sources-jar/kotlin/internal/jdk8/JDK8PlatformImplementations.kt
- * /opt/cola/permits/1343419559_1655661587.9846208/0/kotlin-stdlib-jdk8-1-6-21-sources-jar/kotlin/streams/Streams.kt
- * /opt/cola/permits/1343419559_1655661587.9846208/0/kotlin-stdlib-jdk8-1-6-21-sources-jar/kotlin/text/RegexExtensions.kt
- * /opt/cola/permits/1343419559_1655661587.9846208/0/kotlin-stdlib-jdk8-1-6-21-sources-jar/kotlin/collections/Collections.kt

1.30 stringtemplate4 4.3

1.30.1 Available under license :

ANTLR Project Contributors Certification of Origin and Rights

All contributors to StringTemplate v4 must formally agree to abide by this certificate of origin by signing on the bottom with their github userid, full name, email address (you can obscure your e-mail, but it must be computable by human), and date.

By signing this agreement, you are warranting and representing that you have the right to release code contributions or other content free of any obligations to third parties and are granting Terence Parr and ANTLR project contributors, henceforth referred to as The ANTLR Project, a license to incorporate it into The ANTLR Project tools (such as ANTLRWorks and StringTemplate) or related works under the BSD license. You understand that The ANTLR Project may or may not incorporate your contribution and you warrant and represent the following:

1. I am the creator of all my contributions. I am the author of all contributed work submitted and further warrant and represent that such work is my original creation and I have the right to license it to The ANTLR Project for release under the 3-clause BSD license. I hereby grant The ANTLR Project a nonexclusive, irrevocable, royalty-free, worldwide license to reproduce, distribute, prepare derivative works, and otherwise use this contribution as part of the ANTLR project, associated documentation, books, and tools at no cost to The ANTLR Project.
2. I have the right to submit. This submission does not violate the rights of any person or entity and that I have legal authority over this submission and to make this certification.
3. If I violate another's rights, liability lies with me. I agree to defend, indemnify, and hold The ANTLR Project and ANTLR users

harmless from any claim or demand, including reasonable attorney fees, made by any third party due to or arising out of my violation of these terms and conditions or my violation of the rights of another person or entity.

4. I understand and agree that this project and the contribution are public and that a record of the contribution (including all personal information I submit with it, including my sign-off) is maintained indefinitely and may be redistributed consistent with this project or the open source license indicated in the file.

I have read this agreement and do so certify by adding my signoff to the end of the following contributors list.

CONTRIBUTORS:

YYYY/MM/DD, github id, Full name, email

2012/07/12, parrt, Terence Parr, parrt@antlr.org

2012/08/13, pgelinas, Pascal Glinas, pascal.gelinas@polymtl.ca

2015/05/28, jsnyders, John Snyder, jjsnyders at rcn.com

2015/12/07, sharwell, Sam Harwell, sam@tunnelvisionlabs.com

2016/08/23, BurtHarris, Burt Harris, Burt_Harris.github@azxs.33mail.com

2016/07/18, jeff5, Jeff Allen, ja.py@farowl.co.uk

2018/11/06, drealeed , Drea Leed, drealeed2@yahoo.com

2018/11/08, leonlee, Leon Lee, blackicebird@gmail.com

2018/11/11, adityanarkar, Aditya Narkar, aditya.narkar25@gmail.com

2019/01/21, cfraizer, Colin Frazier, colin.fraizer@gmail.com

2019/09/09, seanabraham, Sean Abraham, Sean.A208@gmail.com

2019/12/22, Clashsoft, Adrian Kunz, clashsoft at hotmail dot com

2020/04/21, steinybot, Jason Pickens, jasonpickensnz@gmail.com

2020/07/23, mma-tapad, Marvin Ma, marvin.ma@tapad.com

2020/07/23, zjzsliyang, Yang Li, zjzsliyang@gmail.com

2020/07/23, jamesmahler2, James Mahler, jmahler@andrew.cmu.edu

2020/07/23, dyuan0226, David Yuan, dyuan1@andrew.cmu.edu

2020/09/06, peteruhnak, Peter Uhnak, i.uhnak@gmail.com

2020/09/16, Dvoreth, Jarmila Emanuela Panwitz, jaremapan@gmail.com

2020/10/01, beccagaspard, Becca Gaspard, beccagaspard at gmail dot com

2021/11/10, StephanRichter, Stephan Richter, postbox: s.richter domain: srsoftware.de

[The "BSD license"]

Copyright (c) 2011-2022 Terence Parr

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.31 commons-csv commons-csv-1.8

1.31.1 Available under license :

Apache Commons CSV
Copyright 2005-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<https://www.apache.org/>).

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of,

publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution

notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing

the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.32 jetty-java-based-http-1-x-http-2-servlet-websocket-server 9.4.49.v20220914

1.32.1 Available under license :

This program and the accompanying materials are made available under the terms of the Eclipse Public License 2.0 which is available at <http://www.eclipse.org/legal/epl-2.0>, or the Apache Software License 2.0 which is available at <https://www.apache.org/licenses/LICENSE-2.0>.

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

- a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
- b) in the case of each subsequent Contributor:
 - i) changes to the Program, and
 - ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

- a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.
- b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and

object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

- c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.
- d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

- a) it complies with the terms and conditions of this Agreement; and
- b) its license agreement:
 - i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
 - ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
 - iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
 - iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

- a) it must be made available under this Agreement; and
- b) a copy of this Agreement must be included with each copy of the Program. Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must:

- a) promptly notify the Commercial Contributor in writing of such claim, and
- b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the

risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the

Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation,

and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s)

with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

=====
Jetty Web Container

Copyright 1995-2018 Mort Bay Consulting Pty Ltd.
=====

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd unless otherwise noted.

Jetty is dual licensed under both

* The Apache 2.0 License

<http://www.apache.org/licenses/LICENSE-2.0.html>

and

* The Eclipse Public 1.0 License

<http://www.eclipse.org/legal/epl-v10.html>

Jetty may be distributed under either license.

Eclipse

The following artifacts are EPL.

- * org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.

- * org.eclipse.jetty.orbit:javac.security.auth.message

The following artifacts are EPL and CDDL 1.0.

- * org.eclipse.jetty.orbit:javac.mail.glassfish

Oracle

The following artifacts are CDDL + GPLv2 with classpath exception.

<https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html>

- * javax.servlet:javax.servlet-api
- * javax.annotation:javax.annotation-api
- * javax.transaction:javax.transaction-api
- * javax.websocket:javax.websocket-api

Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN module is selected.

- * java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions.

<http://openjdk.java.net/legal/gplv2+ce.html>

OW2

The following artifacts are licensed by the OW2 Foundation according to the terms of <http://asm.ow2.org/license.html>

- org.ow2.asm:asm-commons
- org.ow2.asm:asm

Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

MortBay

The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util

org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

Mortbay

The following artifacts are CDDL + GPLv2 with classpath exception.

<https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html>

org.eclipse.jetty.toolchain:jetty-schemas

Assorted

The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville. Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies.

1.33 jackson-dataformat-yaml 2.14.0

1.33.1 Available under license :

Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library. It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has been in development since 2007.

It is currently developed by a community of developers, as well as supported commercially by FasterXML.com.

Licensing

Jackson core and extension components may be licensed under different licenses. To find the details that apply to this artifact see the accompanying LICENSE file. For more information, including possible other licensing options, contact FasterXML.com (<http://fasterxml.com>).

Credits

A list of contributors may be found from CREDITS file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.

This copy of Jackson JSON processor YAML module is licensed under the Apache (Software) License, version 2.0 ("the License").

See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

<http://www.apache.org/licenses/LICENSE-2.0>

1.34 apiguardian-apiguardian-api 1.1.0

1.34.1 Available under license :

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but

excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. **Grant of Copyright License.** Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. **Grant of Patent License.** Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. **Redistribution.** You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its

distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise,

unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.35 okhttp 4.9.3

1.35.1 Available under license :

Note that publicsuffices.gz is compiled from The Public Suffix List:
https://publicsuffix.org/list/public_suffix_list.dat

It is subject to the terms of the Mozilla Public License, v. 2.0:
<https://mozilla.org/MPL/2.0/>

1.36 google-gson 2.9.0

1.36.1 Available under license :

Google Gson

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but

not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their

Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with

the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 2008-2011 Google Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of,

publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution

notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing

the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.37 junit-jupiter-junit-jupiter-engine 5.7.1

1.37.1 Available under license :

Eclipse Public License - v 2.0

=====

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE (AGREEMENT). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. Definitions

Contribution means:

- * **a)** in the case of the initial Contributor, the initial content Distributed under this Agreement, and
- * **b)** in the case of each subsequent Contributor:
 - * **i)** changes to the Program, and
 - * **ii)** additions to the Program;

where such changes and/or additions to the Program originate from and are Distributed by that particular Contributor. A Contribution originates from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include changes or additions to the Program that are not Modified Works.

Contributor means any person or entity that Distributes the Program.

Licensed Patents mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

Program means the Contributions Distributed in accordance with this Agreement.

Recipient means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

Derivative Works shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

Modified Works shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

Distribute means the acts of **a)** distributing or **b)** making available in any manner that enables the transfer of a copy.

Source Code means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

Secondary License means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.

2. Grant of Rights

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive,

worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

****c)**** Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

****d)**** Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

****e)**** Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

3. Requirements

****3.1)**** If a Contributor Distributes the Program in any form, then:

* ****a)**** the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and

* ****b)**** the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:

* ****i)**** effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

* ****ii)**** effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

* ****iii)**** does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and

* ****iv)**** requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.

****3.2)**** When the Program is Distributed as Source Code:

* ****a)**** it must be made available under this Agreement, or if the Program ****i)**** is combined with other material in a separate file or files made available under a Secondary License, and ****ii)**** the initial Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available under the terms of such Secondary Licenses, and

* ****b)**** a copy of this Agreement must be included with each copy of the Program.

****3.3**** Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability (notices) contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

4. Commercial Distribution

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor (Commercial Contributor) hereby agrees to defend and indemnify every other Contributor (Indemnified Contributor) against any losses, damages and costs (collectively Losses) arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: ****a)**** promptly notify the Commercial Contributor in writing of such claim, and ****b)**** allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. No Warranty

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN AS IS BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. Disclaimer of Liability

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF

SUCH DAMAGES.

7. General

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.

Exhibit A - Form of Secondary Licenses Notice

> This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: {name license(s), version(s), and exceptions or additional permissions here}.

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

=====

This product may include a number of subcomponents with separate copyright notices and license terms. Your use of the source code for these subcomponents is subject to the terms and conditions of the subcomponent's license, as noted in the LICENSE-<subcomponent>.md files.

1.38 prometheus-java-simpleclient-common 0.15.0

1.38.1 Available under license :

No license file was found, but licenses were detected in source scan.

<url><http://www.apache.org/licenses/LICENSE-2.0.txt></url>

Found in path(s):

* /opt/cola/permits/1341561718_1654802874.19552/0/simpleclient-common-0-15-0-sources-jar/META-INF/maven/io.prometheus/simpleclient_common/pom.xml

1.39 opentelemetry-instrumentation-for-java 1.15.0-alpha

1.39.1 Available under license :

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or

otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents

of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.40 jcommander-library 1.72

1.40.1 Available under license :

No license file was found, but licenses were detected in source scan.

/**

* Copyright (C) 2011 the original author or authors.

* See the notice.md file distributed with this work for additional

* information regarding copyright ownership.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1009978799_1649800662.98/0/jcommander-1-72-sources-
jar/com/beust/jcommander/validators/NoValueValidator.java
* /opt/cola/permits/1009978799_1649800662.98/0/jcommander-1-72-sources-
jar/com/beust/jcommander/IPParameterValidator.java
* /opt/cola/permits/1009978799_1649800662.98/0/jcommander-1-72-sources-
jar/com/beust/jcommander/validators/PositiveInteger.java
* /opt/cola/permits/1009978799_1649800662.98/0/jcommander-1-72-sources-
jar/com/beust/jcommander/IPParameterValidator2.java
* /opt/cola/permits/1009978799_1649800662.98/0/jcommander-1-72-sources-
jar/com/beust/jcommander/validators/NoValidator.java

No license file was found, but licenses were detected in source scan.

/**

* Copyright (C) 2010 the original author or authors.
* See the notice.md file distributed with this work for additional
* information regarding copyright ownership.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1009978799_1649800662.98/0/jcommander-1-72-sources-

jar/com/beust/jcommander/ResourceBundle.java
* /opt/cola/permits/1009978799_1649800662.98/0/jcommander-1-72-sources-
jar/com/beust/jcommander/converters/BaseConverter.java
* /opt/cola/permits/1009978799_1649800662.98/0/jcommander-1-72-sources-
jar/com/beust/jcommander/internal/Sets.java
* /opt/cola/permits/1009978799_1649800662.98/0/jcommander-1-72-sources-
jar/com/beust/jcommander/ParameterDescription.java
* /opt/cola/permits/1009978799_1649800662.98/0/jcommander-1-72-sources-
jar/com/beust/jcommander/converters/IntegerConverter.java
* /opt/cola/permits/1009978799_1649800662.98/0/jcommander-1-72-sources-
jar/com/beust/jcommander/converters/ISO8601DateConverter.java
* /opt/cola/permits/1009978799_1649800662.98/0/jcommander-1-72-sources-
jar/com/beust/jcommander/converters/StringConverter.java
* /opt/cola/permits/1009978799_1649800662.98/0/jcommander-1-72-sources-
jar/com/beust/jcommander/ParametersDelegate.java
* /opt/cola/permits/1009978799_1649800662.98/0/jcommander-1-72-sources-
jar/com/beust/jcommander/converters/LongConverter.java
* /opt/cola/permits/1009978799_1649800662.98/0/jcommander-1-72-sources-
jar/com/beust/jcommander/IDefaultProvider.java
* /opt/cola/permits/1009978799_1649800662.98/0/jcommander-1-72-sources-
jar/com/beust/jcommander/converters/FileConverter.java
* /opt/cola/permits/1009978799_1649800662.98/0/jcommander-1-72-sources-
jar/com/beust/jcommander/converters/PathConverter.java
* /opt/cola/permits/1009978799_1649800662.98/0/jcommander-1-72-sources-
jar/com/beust/jcommander/defaultprovider/PropertyFileDefaultProvider.java
* /opt/cola/permits/1009978799_1649800662.98/0/jcommander-1-72-sources-
jar/com/beust/jcommander/Parameters.java
* /opt/cola/permits/1009978799_1649800662.98/0/jcommander-1-72-sources-
jar/com/beust/jcommander/converters/InetAddressConverter.java
* /opt/cola/permits/1009978799_1649800662.98/0/jcommander-1-72-sources-
jar/com/beust/jcommander/IStringConverterFactory.java
* /opt/cola/permits/1009978799_1649800662.98/0/jcommander-1-72-sources-
jar/com/beust/jcommander/converters/BigDecimalConverter.java
* /opt/cola/permits/1009978799_1649800662.98/0/jcommander-1-72-sources-
jar/com/beust/jcommander/ParameterException.java
* /opt/cola/permits/1009978799_1649800662.98/0/jcommander-1-72-sources-
jar/com/beust/jcommander/converters/FloatConverter.java
* /opt/cola/permits/1009978799_1649800662.98/0/jcommander-1-72-sources-
jar/com/beust/jcommander/converters/NoConverter.java
* /opt/cola/permits/1009978799_1649800662.98/0/jcommander-1-72-sources-
jar/com/beust/jcommander/internal/Maps.java
* /opt/cola/permits/1009978799_1649800662.98/0/jcommander-1-72-sources-
jar/com/beust/jcommander/MissingCommandException.java
* /opt/cola/permits/1009978799_1649800662.98/0/jcommander-1-72-sources-
jar/com/beust/jcommander/converters/URICConverter.java
* /opt/cola/permits/1009978799_1649800662.98/0/jcommander-1-72-sources-
jar/com/beust/jcommander/IStringConverter.java
* /opt/cola/permits/1009978799_1649800662.98/0/jcommander-1-72-sources-

```
jar/com/beust/jcommander/internal/Lists.java
* /opt/cola/permits/1009978799_1649800662.98/0/jcommander-1-72-sources-
jar/com/beust/jcommander/converters/URLConverter.java
* /opt/cola/permits/1009978799_1649800662.98/0/jcommander-1-72-sources-
jar/com/beust/jcommander/converters/BooleanConverter.java
* /opt/cola/permits/1009978799_1649800662.98/0/jcommander-1-72-sources-
jar/com/beust/jcommander/internal/DefaultConverterFactory.java
* /opt/cola/permits/1009978799_1649800662.98/0/jcommander-1-72-sources-
jar/com/beust/jcommander/converters/DoubleConverter.java
* /opt/cola/permits/1009978799_1649800662.98/0/jcommander-1-72-sources-
jar/com/beust/jcommander/converters/CharArrayConverter.java
* /opt/cola/permits/1009978799_1649800662.98/0/jcommander-1-72-sources-
jar/com/beust/jcommander/Parameter.java
No license file was found, but licenses were detected in source scan.
```

```
/**
 * Copyright (C) 2010 the original author or authors.
 * See the notice.md file distributed with this work for additional
 * information regarding copyright ownership.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1009978799_1649800662.98/0/jcommander-1-72-sources-
jar/com/beust/jcommander/JCommander.java
```

1.41 curator-recipes 5.2.1

1.41.1 Available under license :

Curator Recipes

Copyright 2011-2022 The Apache Software Foundation

This product includes software developed at

The Apache Software Foundation (<http://www.apache.org/>).

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or

agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.42 micronaut-object-storage 1.0.0

1.42.1 Available under license :

Apache License

Version 2.0, January 2004

<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You

institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.43 jvm-integration-for-metrics 4.0.5

1.43.1 Available under license :

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0

Bnd-LastModified: 1545937890888

Build-Jdk: 1.8.0_191

Built-By: artem

Bundle-Description: A set of classes which allow you to monitor critical aspects of your Java Virtual Machine using Metrics.

Bundle-License: <http://www.apache.org/licenses/LICENSE-2.0.html>

Bundle-ManifestVersion: 2

Bundle-Name: JVM Integration for Metrics

Bundle-SymbolicName: io.dropwizard.metrics.jvm

Bundle-Version: 4.0.5

Created-By: Apache Maven Bundle Plugin

Export-Package: com.codahale.metrics.jvm;uses:="com.codahale.metrics,javax.management";version="4.0.5"

Implementation-Title: JVM Integration for Metrics

Implementation-URL: <http://metrics.dropwizard.io/metrics-jvm>

Implementation-Vendor-Id: io.dropwizard.metrics

Implementation-Version: 4.0.5
Import-Package: org.slf4j;version="[1.6.0,2.0.0)",com.sun.management;
resolution:=optional,com.codahale.metrics;version="[4.0,5)",javax.man
agement
Require-Capability: osgi.ee;filter="(&(osgi.ee=JavaSE)(version=1.8))"
Tool: Bnd-3.3.0.201609221906

Found in path(s):

* /opt/cola/permits/1274701574_1648835908.24/0/metrics-jvm-4-0-5-jar/META-INF/MANIFEST.MF

1.44 metrics---dropwizard v4.0.5

1.44.1 Available under license :

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0
Bnd-LastModified: 1545937975057
Build-Jdk: 1.8.0_191
Built-By: artem
Bundle-Description: An Apache HttpClient wrapper providing Metrics ins
trumentation of connection pools, request durations and rates,
and other useful information.
Bundle-License: <http://www.apache.org/licenses/LICENSE-2.0.html>
Bundle-ManifestVersion: 2
Bundle-Name: Metrics Integration for Apache HttpClient
Bundle-SymbolicName: io.dropwizard.metrics.httpclient
Bundle-Version: 4.0.5
Created-By: Apache Maven Bundle Plugin
Export-Package: com.codahale.metrics.httpclient;uses:="com.codahale.me
trics,org.apache.http,org.apache.http.config,org.apache.http.conn,org
.apache.http.conn.routing,org.apache.http.conn.socket,org.apache.http
.impl.client,org.apache.http.impl.conn,org.apache.http.protocol";vers
ion="4.0.5"
Implementation-Title: Metrics Integration for Apache HttpClient
Implementation-URL: <http://metrics.dropwizard.io/metrics-httpclient>
Implementation-Vendor-Id: io.dropwizard.metrics
Implementation-Version: 4.0.5
Import-Package: com.codahale.metrics;version="[4.0,5)",org.apache.http
,org.apache.http.client,org.apache.http.client.methods,org.apache.htt
p.client.utils,org.apache.http.config,org.apache.http.conn,org.apache
.http.conn.routing,org.apache.http.conn.socket,org.apache.http.conn.s
sl,org.apache.http.impl.client,org.apache.http.impl.conn,org.apache.h
ttp.pool,org.apache.http.protocol
Require-Capability: osgi.ee;filter="(&(osgi.ee=JavaSE)(version=1.8))"
Tool: Bnd-3.3.0.201609221906

Found in path(s):

1.45 protobuf-java 3.19.4

1.45.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
// Copyright 2008 Google Inc. All rights reserved.  
// Redistribution and use in source and binary forms, with or without  
// modification, are permitted provided that the following conditions are  
// * Redistributions of source code must retain the above copyright  
// notice, this list of conditions and the following disclaimer.  
// * Redistributions in binary form must reproduce the above  
// copyright notice, this list of conditions and the following disclaimer  
// in the documentation and/or other materials provided with the  
// * Neither the name of Google Inc. nor the names of its  
// this software without specific prior written permission.
```

Found in path(s):

```
* /opt/cola/permits/1454789367_1666952560.680916/0/protobuf-java-3-19-4-4-jar/google/protobuf/descriptor.proto  
* /opt/cola/permits/1454789367_1666952560.680916/0/protobuf-java-3-19-4-4-jar/google/protobuf/duration.proto  
* /opt/cola/permits/1454789367_1666952560.680916/0/protobuf-java-3-19-4-4-jar/google/protobuf/struct.proto  
* /opt/cola/permits/1454789367_1666952560.680916/0/protobuf-java-3-19-4-4-  
jar/google/protobuf/field_mask.proto  
* /opt/cola/permits/1454789367_1666952560.680916/0/protobuf-java-3-19-4-4-  
jar/google/protobuf/compiler/plugin.proto  
* /opt/cola/permits/1454789367_1666952560.680916/0/protobuf-java-3-19-4-4-jar/google/protobuf/empty.proto  
* /opt/cola/permits/1454789367_1666952560.680916/0/protobuf-java-3-19-4-4-jar/google/protobuf/type.proto  
* /opt/cola/permits/1454789367_1666952560.680916/0/protobuf-java-3-19-4-4-jar/google/protobuf/any.proto  
* /opt/cola/permits/1454789367_1666952560.680916/0/protobuf-java-3-19-4-4-jar/google/protobuf/wrappers.proto  
* /opt/cola/permits/1454789367_1666952560.680916/0/protobuf-java-3-19-4-4-  
jar/google/protobuf/timestamp.proto  
* /opt/cola/permits/1454789367_1666952560.680916/0/protobuf-java-3-19-4-4-  
jar/google/protobuf/source_context.proto  
* /opt/cola/permits/1454789367_1666952560.680916/0/protobuf-java-3-19-4-4-jar/google/protobuf/api.proto
```

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0

Automatic-Module-Name: com.google.protobuf

Bnd-LastModified: 1643389670477

Build-Jdk: 1.8.0_181-google-v7

Built-By: acozzette

Bundle-Description: Core Protocol Buffers library. Protocol Buffers are a way of encoding structured data in an efficient yet extensible format.

Bundle-DocURL: <https://developers.google.com/protocol-buffers/>

Bundle-License: <https://opensource.org/licenses/BSD-3-Clause>

Bundle-ManifestVersion: 2
Bundle-Name: Protocol Buffers [Core]
Bundle-SymbolicName: com.google.protobuf
Bundle-Version: 3.19.4
Created-By: Apache Maven Bundle Plugin
Export-Package: com.google.protobuf;version="3.19.4"
Import-Package: sun.misc;resolution:=optional,com.google.protobuf;version="[3.19,4)"
Require-Capability: osgi.ee;filter="(&(osgi.ee=JavaSE)(version=1.7))"
Tool: Bnd-3.0.0.201509101326

Found in path(s):

* /opt/cola/permits/1454789367_1666952560.680916/0/protobuf-java-3-19-4-4-jar/META-INF/MANIFEST.MF

1.46 micronaut-problem-json 2.5.1

1.46.1 Available under license :

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical

transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable

by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use,

reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.47 cloudevents---core 2.2.0

1.47.1 Available under license :

No license file was found, but licenses were detected in source scan.

<!--

~ Copyright 2018-Present The CloudEvents Authors

~ <p>

~ Licensed under the Apache License, Version 2.0 (the "License");

~ you may not use this file except in compliance with the License.

~ You may obtain a copy of the License at

~ <p>

~ <http://www.apache.org/licenses/LICENSE-2.0>

~ <p>

~ Unless required by applicable law or agreed to in writing, software

~ distributed under the License is distributed on an "AS IS" BASIS,

~ WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

~ See the License for the specific language governing permissions and

~ limitations under the License.

~
-->

Found in path(s):

* /opt/cola/permits/1340815883_1654861257.1698394/0/cloudevents-core-2-2-0-sources-jar/META-INF/maven/io.cloudevents/cloudevents-core/pom.xml

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2018-Present The CloudEvents Authors

* <p>

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

* <p>

* <http://www.apache.org/licenses/LICENSE-2.0>

* <p>

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*

*/

Found in path(s):

* /opt/cola/permits/1340815883_1654861257.1698394/0/cloudevents-core-2-2-0-sources-jar/io/cloudevents/core/v03/CloudEventBuilder.java

* /opt/cola/permits/1340815883_1654861257.1698394/0/cloudevents-core-2-2-0-sources-jar/io/cloudevents/core/v1/V03ToV1AttributesConverter.java

* /opt/cola/permits/1340815883_1654861257.1698394/0/cloudevents-core-2-2-0-sources-jar/io/cloudevents/core/impl/CloudEventContextReaderAdapter.java

* /opt/cola/permits/1340815883_1654861257.1698394/0/cloudevents-core-2-2-0-sources-jar/io/cloudevents/core/v1/CloudEventBuilder.java

* /opt/cola/permits/1340815883_1654861257.1698394/0/cloudevents-core-2-2-0-sources-jar/io/cloudevents/core/CloudEventUtils.java

* /opt/cola/permits/1340815883_1654861257.1698394/0/cloudevents-core-2-2-0-sources-jar/io/cloudevents/core/builder/CloudEventBuilder.java

* /opt/cola/permits/1340815883_1654861257.1698394/0/cloudevents-core-2-2-0-sources-jar/io/cloudevents/core/extensions/impl/ExtensionUtils.java

* /opt/cola/permits/1340815883_1654861257.1698394/0/cloudevents-core-2-2-0-sources-jar/io/cloudevents/core/message/StructuredMessageReader.java

* /opt/cola/permits/1340815883_1654861257.1698394/0/cloudevents-core-2-2-0-sources-jar/io/cloudevents/core/v03/V1ToV03AttributesConverter.java

* /opt/cola/permits/1340815883_1654861257.1698394/0/cloudevents-core-2-2-0-sources-jar/io/cloudevents/core/message/impl/GenericStructuredMessageReader.java

* /opt/cola/permits/1340815883_1654861257.1698394/0/cloudevents-core-2-2-0-sources-jar/io/cloudevents/core/provider/EventFormatProvider.java

* /opt/cola/permits/1340815883_1654861257.1698394/0/cloudevents-core-2-2-0-sources-jar/io/cloudevents/core/provider/ExtensionProvider.java
* /opt/cola/permits/1340815883_1654861257.1698394/0/cloudevents-core-2-2-0-sources-jar/io/cloudevents/core/message/impl/BaseGenericBinaryMessageReaderImpl.java
* /opt/cola/permits/1340815883_1654861257.1698394/0/cloudevents-core-2-2-0-sources-jar/io/cloudevents/core/message/Encoding.java
* /opt/cola/permits/1340815883_1654861257.1698394/0/cloudevents-core-2-2-0-sources-jar/io/cloudevents/core/v1/CloudEventV1.java
* /opt/cola/permits/1340815883_1654861257.1698394/0/cloudevents-core-2-2-0-sources-jar/io/cloudevents/core/message/MessageReader.java
* /opt/cola/permits/1340815883_1654861257.1698394/0/cloudevents-core-2-2-0-sources-jar/io/cloudevents/core/format/EventFormat.java
* /opt/cola/permits/1340815883_1654861257.1698394/0/cloudevents-core-2-2-0-sources-jar/io/cloudevents/core/v03/CloudEventV03.java
* /opt/cola/permits/1340815883_1654861257.1698394/0/cloudevents-core-2-2-0-sources-jar/io/cloudevents/core/impl/BaseCloudEventBuilder.java
* /opt/cola/permits/1340815883_1654861257.1698394/0/cloudevents-core-2-2-0-sources-jar/io/cloudevents/core/extensions/DatarefExtension.java
* /opt/cola/permits/1340815883_1654861257.1698394/0/cloudevents-core-2-2-0-sources-jar/io/cloudevents/core/message/MessageWriter.java
* /opt/cola/permits/1340815883_1654861257.1698394/0/cloudevents-core-2-2-0-sources-jar/io/cloudevents/core/format/EventDeserializationException.java
* /opt/cola/permits/1340815883_1654861257.1698394/0/cloudevents-core-2-2-0-sources-jar/io/cloudevents/core/message/impl/BaseStructuredMessageReader.java
* /opt/cola/permits/1340815883_1654861257.1698394/0/cloudevents-core-2-2-0-sources-jar/io/cloudevents/core/impl/CloudEventReaderAdapter.java
* /opt/cola/permits/1340815883_1654861257.1698394/0/cloudevents-core-2-2-0-sources-jar/io/cloudevents/core/format/EventSerializationException.java
* /opt/cola/permits/1340815883_1654861257.1698394/0/cloudevents-core-2-2-0-sources-jar/io/cloudevents/core/extensions/DistributedTracingExtension.java
* /opt/cola/permits/1340815883_1654861257.1698394/0/cloudevents-core-2-2-0-sources-jar/io/cloudevents/core/message/impl/MessageUtils.java
* /opt/cola/permits/1340815883_1654861257.1698394/0/cloudevents-core-2-2-0-sources-jar/io/cloudevents/core/message/impl/BaseBinaryMessageReader.java
* /opt/cola/permits/1340815883_1654861257.1698394/0/cloudevents-core-2-2-0-sources-jar/io/cloudevents/core/message/StructuredMessageWriter.java
* /opt/cola/permits/1340815883_1654861257.1698394/0/cloudevents-core-2-2-0-sources-jar/io/cloudevents/core/impl/BaseCloudEvent.java

1.48 micronaut-kafka 4.4.0

1.48.1 Available under license :

Apache License

Version 2.0, January 2004

<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and

attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the

appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.49 fabric8-::-kubernetes-model-::- networking 4.13.3

1.49.1 Available under license :

No license file was found, but licenses were detected in source scan.

Copyright (C) 2015 Red Hat, Inc.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE>

2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Found in path(s):

* /opt/cola/permits/1288519867_1647861830.19/0/kubernetes-model-networking-4-13-3-jar/META-
INF/maven/io.fabric8/kubernetes-model-networking/pom.xml

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0

Bnd-LastModified: 1619068616615

Build-Jdk-Spec: 1.8

Bundle-Description: Java client for Kubernetes and OpenShift

Bundle-DocURL: <http://redhat.com>

Bundle-License: <http://www.apache.org/licenses/LICENSE-2.0.txt>

Bundle-ManifestVersion: 2

Bundle-Name: Fabric8 :: Kubernetes Model :: Networking

Bundle-SymbolicName: io.fabric8.kubernetes-model-networking

Bundle-Vendor: Red Hat

Bundle-Version: 4.13.3

Created-By: Apache Maven Bundle Plugin

Export-Package: io.fabric8.kubernetes.api.model.networking.v1;uses:="com.fasterxml.jackson.annotation,com.fasterxml.jackson.databind,com.fasterxml.jackson.databind.annotation,io.fabric8.kubernetes.api.builder

,io.fabric8.kubernetes.api.model,io.fabric8.kubernetes.model.annotati
on";version="4.13.3",io.fabric8.kubernetes.api.model.networking.v1beta
1;uses="com.fasterxml.jackson.annotation,com.fasterxml.jackson.data
bind,com.fasterxml.jackson.databind.annotation,io.fabric8.kubernetes.
api.builder,io.fabric8.kubernetes.api.model,io.fabric8.kubernetes.mod
el.annotation";version="4.13.3"

Implementation-Title: Fabric8 :: Kubernetes Model :: Networking

Implementation-Vendor: Red Hat

Implementation-Version: 4.13.3

Import-Package: com.fasterxml.jackson.annotation;version="[2.11,3)",co
m.fasterxml.jackson.databind;version="[2.11,3)",com.fasterxml.jackson
.databind.annotation;version="[2.11,3)",io.fabric8.kubernetes.api.bui
lder;version="[4.13,5)",io.fabric8.kubernetes.api.model,io.fabric8.ku
bernetes.model.annotation;version="[4.13,5)"

Require-Capability: osgi.ee;filter="(&(osgi.ee=JavaSE)(version=1.8))"

Specification-Title: Fabric8 :: Kubernetes Model :: Networking

Specification-Vendor: Red Hat

Specification-Version: 4.13

Tool: Bnd-5.1.1.202006162103

Found in path(s):

* /opt/cola/permits/1288519867_1647861830.19/0/kubernetes-model-networking-4-13-3-jar/META-
INF/MANIFEST.MF

No license file was found, but licenses were detected in source scan.

*

* Copyright (C) 2015 Red Hat, Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*

Found in path(s):

* /opt/cola/permits/1288519867_1647861830.19/0/kubernetes-model-networking-4-13-3-jar/manifest.vm

1.50 guava-listenablefuture-only 9999.0- empty-to-avoid-conflict-with-guava

1.50.1 Available under license :

Found license 'GNU Lesser General Public License' in '// This library is free software; you can redistribute it and/or // modify it under the terms of the GNU Lesser General Public // License as published by the Free Software Foundation; either // version 2.1 of the License, or (at your option) any later version. // This library is distributed in the hope that it will be useful, // but WITHOUT ANY WARRANTY; without even the implied warranty of // MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU // Lesser General Public License for more details. // You should have received a copy of the GNU Lesser General Public * This grammar is in the PUBLIC DOMAIN'

GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some
specially designated software packages--typically libraries--of the
Free Software Foundation and other authors who decide to use it. You
can use it too, but we suggest you first think carefully about whether
this license or the ordinary General Public License is the better
strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use,
not price. Our General Public Licenses are designed to make sure that
you have the freedom to distribute copies of free software (and charge
for this service if you wish); that you receive source code or can get
it if you want it; that you can change the software and use pieces of
it in new free programs; and that you are informed that you can do
these things.

To protect your rights, we need to make restrictions that forbid
distributors to deny you these rights or to ask you to surrender these
rights. These restrictions translate to certain responsibilities for
you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis

or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for

making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the

application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work

during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally

accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or

otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

////////////////////////////////////

// checkstyle: Checks Java source code for adherence to a set of rules.

// Copyright (C) 2001-2020 the original author or authors.

//

// This library is free software; you can redistribute it and/or

// modify it under the terms of the GNU Lesser General Public

// License as published by the Free Software Foundation; either

// version 2.1 of the License, or (at your option) any later version.

//

// This library is distributed in the hope that it will be useful,

// but WITHOUT ANY WARRANTY; without even the implied warranty of

// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU

// Lesser General Public License for more details.

//

// You should have received a copy of the GNU Lesser General Public

// License along with this library; if not, write to the Free Software

// Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

////////////////////////////////////

1.51 jackson-module-afterburner 2.14.0

1.51.1 Available under license :

Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library.

It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has been in development since 2007.

It is currently developed by a community of developers, as well as supported commercially by FasterXML.com.

Licensing

Jackson core and extension components (as well their dependencies) may be licensed under different licenses.

To find the details that apply to this artifact see the accompanying LICENSE file.

For more information, including possible other licensing options, contact FasterXML.com (<http://fasterxml.com>).

Credits

A list of contributors may be found from CREDITS file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.

This copy of Jackson JSON processor `jackson-module-afterburner` module is licensed under the Apache (Software) License, version 2.0 ("the License").

See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

<http://www.apache.org/licenses/LICENSE-2.0>

Additional licensing information exists for following 3rd party library dependencies

ASM

ASM: a very small and fast Java bytecode manipulation framework

Copyright (c) 2000-2011 INRIA, France Telecom

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright

- notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
 3. Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.52 open-telemetry/opentelemetry-java 1.9.1

1.52.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2014 The gRPC Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1340815879_1654812121.818436/0/opentelemetry-exporter-otlp-common-1-9-1-sources-
jar/io/opentelemetry/exporter/otlp/internal/grpc/MarshalerInputStream.java
* /opt/cola/permits/1340815879_1654812121.818436/0/opentelemetry-exporter-otlp-common-1-9-1-sources-
jar/io/opentelemetry/exporter/otlp/internal/grpc/OkHttpGrpcExporter.java
```

No license file was found, but licenses were detected in source scan.

```
// Copyright 2008 Google Inc. All rights reserved.  
// Redistribution and use in source and binary forms, with or without  
// modification, are permitted provided that the following conditions are  
// * Redistributions of source code must retain the above copyright  
// notice, this list of conditions and the following disclaimer.  
// * Redistributions in binary form must reproduce the above  
// copyright notice, this list of conditions and the following disclaimer  
// in the documentation and/or other materials provided with the  
// * Neither the name of Google Inc. nor the names of its  
// this software without specific prior written permission.
```

Found in path(s):

```
* /opt/cola/permits/1340815879_1654812121.818436/0/opentelemetry-exporter-otlp-common-1-9-1-sources-  
jar/io/opentelemetry/exporter/otlp/internal/CodedOutputStream.java  
* /opt/cola/permits/1340815879_1654812121.818436/0/opentelemetry-exporter-otlp-common-1-9-1-sources-  
jar/io/opentelemetry/exporter/otlp/internal/CodedInputStream.java  
* /opt/cola/permits/1340815879_1654812121.818436/0/opentelemetry-exporter-otlp-common-1-9-1-sources-  
jar/io/opentelemetry/exporter/otlp/internal/WireFormat.java
```

1.53 junit-jupiter-junit-jupiter-engine 5.9.1

1.53.1 Available under license :

Eclipse Public License - v 2.0

=====

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE (AGREEMENT). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. Definitions

Contribution means:

- * **a)** in the case of the initial Contributor, the initial content Distributed under this Agreement, and
- * **b)** in the case of each subsequent Contributor:
 - * **i)** changes to the Program, and
 - * **ii)** additions to the Program;

where such changes and/or additions to the Program originate from and are Distributed by that particular Contributor. A Contribution originates from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include changes or additions to the Program that are not Modified Works.

Contributor means any person or entity that Distributes the Program.

Licensed Patents mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

Program means the Contributions Distributed in accordance with this Agreement.

Recipient means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

Derivative Works shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

Modified Works shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

Distribute means the acts of ****a)**** distributing or ****b)**** making available in any manner that enables the transfer of a copy.

Source Code means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

Secondary License means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.

2. Grant of Rights

****a)**** Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

****b)**** Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

****c)**** Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

****d)**** Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

****e)**** Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

3. Requirements

****3.1)**** If a Contributor Distributes the Program in any form, then:

****a)**** the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and

****b)**** the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:

****i)**** effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

****ii)**** effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

****iii)**** does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and

****iv)**** requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.

****3.2)**** When the Program is Distributed as Source Code:

****a)**** it must be made available under this Agreement, or if the Program ****i)**** is combined with other material in a separate file or files made available under a Secondary License, and ****ii)**** the initial Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available under the terms of such Secondary Licenses, and

****b)**** a copy of this Agreement must be included with each copy of the Program.

****3.3)**** Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability (notices) contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

4. Commercial Distribution

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor (Commercial Contributor) hereby agrees to defend and indemnify every other Contributor (Indemnified Contributor) against any losses, damages and costs (collectively Losses) arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or

alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: ****a)**** promptly notify the Commercial Contributor in writing of such claim, and ****b)**** allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. No Warranty

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN AS IS BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. Disclaimer of Liability

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. General

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and

distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.

Exhibit A - Form of Secondary Licenses Notice

> This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: {name license(s), version(s), and exceptions or additional permissions here}.

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Open Source Licenses

=====

This product may include a number of subcomponents with separate copyright notices and license terms. Your use of the source code for these subcomponents is subject to the terms and conditions of the subcomponent's license, as noted in the LICENSE-<subcomponent>.md files.

1.54 byte-buddy byte-buddy-1.9.10

1.54.1 Available under license :

Apache License
Version 2.0, January 2004

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally

submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or

implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Apache Maven includes a number of components and libraries with separate copyright notices and license terms. Your use of those components are subject to the terms and conditions of the following licenses.

AOP alliance (<http://aopalliance.sourceforge.net>) aopalliance:aopalliance:jar:1.0

License: Public Domain

JSR-250 Common Annotations for the Java™ Platform

(<http://jcp.org/aboutJava/communityprocess/final/jsr250/index.html>) javax.annotation:jsr250-api:jar:1.0

License: COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

<https://glassfish.java.net/public/CDDLv1.0.html> (lib/jsr250-api.license)

CDI APIs (<http://www.seamframework.org/Weld/cdi-api>) javax.enterprise:cdi-api:jar:1.0

License: Apache License, Version 2.0 <http://www.apache.org/licenses/LICENSE-2.0> (lib/cdi-api.license)

Maven Aether Provider (<http://maven.apache.org/ref/3.2.5/maven-aether-provider>) org.apache.maven:maven-aether-provider:jar:3.2.5

License: Apache License, Version 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt> (lib/maven-aether-provider.license)

Maven Artifact (<http://maven.apache.org/ref/3.2.5/maven-artifact>) org.apache.maven:maven-artifact:jar:3.2.5

License: Apache License, Version 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt> (lib/maven-artifact.license)

Maven Compat (<http://maven.apache.org/ref/3.2.5/maven-compat>) org.apache.maven:maven-compat:jar:3.2.5

License: Apache License, Version 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt> (lib/maven-compat.license)

Maven Core (<http://maven.apache.org/ref/3.2.5/maven-core>) org.apache.maven:maven-core:jar:3.2.5

License: Apache License, Version 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt> (lib/maven-core.license)

Maven Embedder (<http://maven.apache.org/ref/3.2.5/maven-embedder>) org.apache.maven:maven-embedder:jar:3.2.5

License: Apache License, Version 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt> (lib/maven-embedder.license)

Maven Model (<http://maven.apache.org/ref/3.2.5/maven-model>) org.apache.maven:maven-model:jar:3.2.5

License: Apache License, Version 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt> (lib/maven-

model.license)

Maven Model Builder (<http://maven.apache.org/ref/3.2.5/maven-model-builder>) org.apache.maven:maven-model-builder:jar:3.2.5

License: Apache License, Version 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt> (lib/maven-model-builder.license)

Maven Plugin API (<http://maven.apache.org/ref/3.2.5/maven-plugin-api>) org.apache.maven:maven-plugin-api:jar:3.2.5

License: Apache License, Version 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt> (lib/maven-plugin-api.license)

Maven Repository Metadata Model (<http://maven.apache.org/ref/3.2.5/maven-repository-metadata>) org.apache.maven:maven-repository-metadata:jar:3.2.5

License: Apache License, Version 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt> (lib/maven-repository-metadata.license)

Maven Settings (<http://maven.apache.org/ref/3.2.5/maven-settings>) org.apache.maven:maven-settings:jar:3.2.5

License: Apache License, Version 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt> (lib/maven-settings.license)

Maven Settings Builder (<http://maven.apache.org/ref/3.2.5/maven-settings-builder>) org.apache.maven:maven-settings-builder:jar:3.2.5

License: Apache License, Version 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt> (lib/maven-settings-builder.license)

Apache Maven Wagon :: Providers :: File Provider (<http://maven.apache.org/wagon/wagon-providers/wagon-file>) org.apache.maven.wagon:wagon-file:jar:2.8

License: Apache License, Version 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt> (lib/wagon-file.license)

Apache Maven Wagon :: Providers :: HTTP Provider (<http://maven.apache.org/wagon/wagon-providers/wagon-http>) org.apache.maven.wagon:wagon-http:jar:2.8

License: Apache License, Version 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt> (lib/wagon-http.license)

Apache Maven Wagon :: Providers :: HTTP Shared Library (<http://maven.apache.org/wagon/wagon-providers/wagon-http-shared>) org.apache.maven.wagon:wagon-http-shared:jar:2.8

License: Apache License, Version 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt> (lib/wagon-http-shared.license)

Apache Maven Wagon :: API (<http://maven.apache.org/wagon/wagon-provider-api>) org.apache.maven.wagon:wagon-provider-api:jar:2.8

License: Apache License, Version 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt> (lib/wagon-provider-api.license)

Aether API (<http://www.eclipse.org/aether/aether-api/>) org.eclipse.aether:aether-api:jar:1.0.0.v20140518

License: Eclipse Public License, Version 1.0 <http://www.eclipse.org/legal/epl-v10.html> (lib/aether-api.license)

Aether Connector Basic (<http://www.eclipse.org/aether/aether-connector-basic/>) org.eclipse.aether:aether-connector-

basic:jar:1.0.0.v20140518

License: Eclipse Public License, Version 1.0 <http://www.eclipse.org/legal/epl-v10.html> (lib/aether-connector-basic.license)

Aether Implementation (<http://www.eclipse.org/aether/aether-impl/>) org.eclipse.aether:aether-impl:jar:1.0.0.v20140518

License: Eclipse Public License, Version 1.0 <http://www.eclipse.org/legal/epl-v10.html> (lib/aether-impl.license)

Aether SPI (<http://www.eclipse.org/aether/aether-spi/>) org.eclipse.aether:aether-spi:jar:1.0.0.v20140518

License: Eclipse Public License, Version 1.0 <http://www.eclipse.org/legal/epl-v10.html> (lib/aether-spi.license)

Aether Transport Wagon (<http://www.eclipse.org/aether/aether-transport-wagon/>) org.eclipse.aether:aether-transport-wagon:jar:1.0.0.v20140518

License: Eclipse Public License, Version 1.0 <http://www.eclipse.org/legal/epl-v10.html> (lib/aether-transport-wagon.license)

Aether Utilities (<http://www.eclipse.org/aether/aether-util/>) org.eclipse.aether:aether-util:jar:1.0.0.v20140518

License: Eclipse Public License, Version 1.0 <http://www.eclipse.org/legal/epl-v10.html> (lib/aether-util.license)

org.eclipse.sisu.inject (<http://www.eclipse.org/sisu/org.eclipse.sisu.inject/>)

org.eclipse.sisu:org.eclipse.sisu.inject:eclipse-plugin:0.3.0.M1

License: Eclipse Public License, Version 1.0 <http://www.eclipse.org/legal/epl-v10.html> (lib/org.eclipse.sisu.inject.license)

org.eclipse.sisu.plexus (<http://www.eclipse.org/sisu/org.eclipse.sisu.plexus/>)

org.eclipse.sisu:org.eclipse.sisu.plexus:eclipse-plugin:0.3.0.M1

License: Eclipse Public License, Version 1.0 <http://www.eclipse.org/legal/epl-v10.html> (lib/org.eclipse.sisu.plexus.license)

jsoup (<http://jsoup.org/>) org.jsoup:jsoup:jar:1.7.2

License: The MIT License <http://jsoup.com/license> (lib/jsoup.license)

SLF4J API Module (<http://www.slf4j.org>) org.slf4j:slf4j-api:jar:1.7.5

License: MIT License <http://www.opensource.org/licenses/mit-license.php> (lib/slf4j-api.license)

SLF4J Simple Binding (<http://www.slf4j.org>) org.slf4j:slf4j-simple:jar:1.7.5

License: MIT License <http://www.opensource.org/licenses/mit-license.php> (lib/slf4j-simple.license)

Plexus Cipher: encryption/decryption Component (<http://spice.sonatype.org/plexus-cipher>)

org.sonatype.plexus:plexus-cipher:jar:1.7

License: Apache Public License 2.0 <http://www.apache.org/licenses/LICENSE-2.0> (lib/plexus-cipher.license)

Plexus Security Dispatcher Component (<http://spice.sonatype.org/plexus-sec-dispatcher>)

org.sonatype.plexus:plexus-sec-dispatcher:jar:1.3

License: Apache Public License 2.0 <http://www.apache.org/licenses/LICENSE-2.0> (lib/plexus-sec-dispatcher.license)

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML+RDFa 1.0//EN"

"<http://www.w3.org/MarkUp/DTD/xhtml-rdfa-1.dtd>">

```

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" version="XHTML+RDFa 1.0" dir="ltr">

<head profile="http://www.w3.org/1999/xhtml/vocab">
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <link rel="shortcut icon" href="http://opensource.org/files/garland_favicon.png" type="image/png" />
  <link rel="shortlink" href="/node/66" />
  <link rel="canonical" href="/licenses/MIT" />
  <meta name="Generator" content="Drupal 7 (http://drupal.org)" />
  <title>The MIT License (MIT) | Open Source Initiative</title>
  <link type="text/css" rel="stylesheet" href="http://opensource.org/files/css/css_xE-rWrJf-fncB6ztZfd2huxqgxu4WO-qwma6Xer30m4.css" media="all" />
  <link type="text/css" rel="stylesheet" href="http://opensource.org/files/css/css_2ATB4XKGEvmoUk_p62PwI-o2aW47EqqS0nD2dmPZoV4.css" media="all" />
  <link type="text/css" rel="stylesheet" href="http://opensource.org/files/css/css_2wI77kyP-rJKVpFW5M3KFcj7Cb99lZalmubKIwWwsmU.css" media="all" />
  <link type="text/css" rel="stylesheet"
href="http://opensource.org/files/css/css_k3snrbsthqot7V7ccRZHS9OkCZkwBv4adtNieIVlbEU.css" media="print"
/>

<!--[if lt IE 7]>
<link type="text/css" rel="stylesheet" href="http://opensource.org/themes/garland/fix-ie.css?fnb7pm" media="all"
/>
<![endif]-->
<script type="text/javascript"
src="http://opensource.org/files/js/js_xAPI0qIk9eowy_is9tNkCWXLUVoat94SQT48UBCFkyQ.js"></script>
<script type="text/javascript">
<!--/--><![CDATA[//><!--
jQuery.extend(Drupal.settings,
{ "basePath": "\", "pathPrefix": "", "ajaxPageState": { "theme": "garland", "theme_token": "meUjjBBfr6QFJv5kp0oKi152
l673C3xJMLGIQzbH9g0", "js": { "misc/jquery.js": 1, "misc/jquery.once.js": 1, "misc/drupal.js": 1 }, "css": { "modules/system/system.base.css": 1, "modules/system/system.menus.css": 1, "modules/system/system.messages.css": 1, "modules/system/system.theme.css": 1, "modules/aggregator/aggregator.css": 1, "modules/comment/comment.css": 1, "modules/field/theme/field.css": 1, "sites/all/modules/mollom/mollom.css": 1, "modules/node/node.css": 1, "modules/search/search.css": 1, "modules/user/user.css": 1, "themes/garland/style.css": 1, "themes/garland/print.css": 1, "themes/garland/fix-ie.css": 1 } } });
//--><![ ]>
</script>
</head>
<body class="html not-front not-logged-in one-sidebar sidebar-first page-node page-node- page-node-66 node-type-
page fluid-width" >
  <div id="skip-link">
    <a href="#main-content" class="element-invisible element-focusable">Skip to main content</a>
  </div>

  <div id="wrapper">
    <div id="container" class="clearfix">

      <div id="header">

```

```
<div id="logo-floater">
  <div id="branding"><strong><a href="/">
    
    <span>Open Source Initiative</span>      </a></strong></div>
  </div>
```

```
</div> <!-- /#header -->
```

```
<div id="sidebar-first" class="sidebar">
  <div class="region region-sidebar-first">
    <div id="block-search-form" class="block block-search clearfix">
```

```
<h2 class="title">Search this site:</h2>
```

```
<div class="content">
```

```
<form action="/licenses/mit-license.php" method="post" id="search-block-form" accept-charset="UTF-
8"><div><div class="container-inline">
```

```
<div class="form-item form-type-textfield form-item-search-block-form">
```

```
<label class="element-invisible" for="edit-search-block-form--2">Search </label>
```

```
<input title="Enter the terms you wish to search for." type="text" id="edit-search-block-form--2"
name="search_block_form" value="" size="15" maxlength="128" class="form-text" />
```

```
</div>
```

```
<div class="form-actions form-wrapper" id="edit-actions"><input type="submit" id="edit-submit" name="op"
value="Search" class="form-submit" /></div><input type="hidden" name="form_build_id" value="form-
MyBqFtDvdzOmaSnCHKIG9yhm0ofMr7fNMG5Vy76N_uk" />
```

```
<input type="hidden" name="form_id" value="search_block_form" />
```

```
</div>
```

```
</div></form> </div>
```

```
</div>
```

```
<div id="block-system-navigation" class="block block-system block-menu clearfix">
```

```
<h2 class="title">Navigation</h2>
```

```
<div class="content">
```

```
<ul class="menu"><li class="first collapsed"><a href="/about" title="About the Open Source Initiative">About
the OSI</a></li>
```

```
<li class="collapsed"><a href="/osd" title="The actual OSD defining what constitutes an Open Source licence">The
Open Source Definition</a></li>
```

```
<li class="collapsed"><a href="/licenses">Open Source Licenses</a></li>
```

```
<li class="leaf"><a href="/working_groups">Working Groups</a></li>
```

```
<li class="leaf"><a href="/faq" title="Frequently Asked Questions about open source and about the
OSI.">FAQ</a></li>
```

```
<li class="collapsed"><a href="/trademark" title="Information about trademark and logo usage">Trademark and
Logo Usage</a></li>
```

```
<li class="collapsed"><a href="/osr-intro" title="Open Standards Requirement for Software">Open
Standards</a></li>
```

```
<li class="leaf"><a href="/osi-open-source-education" title="OSI's Open Source Education Initiative and
```

Activities">Open Source Education
 <li class="collapsed">Mailing lists
 <li class="collapsed">Getting Help
 <li class="collapsed">Donate to the OSI
 <li class="leaf">OSI Individual Membership
 <li class="leaf">OSI Store
 <li class="collapsed">OSI Affiliate Membership
 <li class="leaf">Contact OSI
 <li class="leaf">Terms of Service
 <li class="last leaf">OSI Corporate Support
 </div>
 </div>
 </div>
 </div>

<div id="center"><div id="squeeze"><div class="right-corner"><div class="left-corner">
 <h2 class="element-invisible">You are here</h2><div class="breadcrumb">Home</div>

 <div id="tabs-wrapper" class="clearfix"> <h1 class="with-tabs">The MIT License
 (MIT)</h1>
 </div> <div class="clearfix">
 <div class="region region-content">
 <div id="block-system-main" class="block block-system clearfix">

<div class="content">
 <div id="node-66" class="node node-page">

<div class="content clearfix">

 <div class="field field-name-body field-type-text-with-summary field-label-hidden"><div class="field-items"><div class="field-item even"><p>The MIT License (MIT)</p>
 <p>Copyright (c) <year> <copyright holders></p>
 <p>Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:</p>
 <p>The above copyright notice and this permission notice shall be included in

all copies or substantial portions of the Software.</p>
<p>THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,

FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE

AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER

LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,

OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN

THE SOFTWARE.</p>
</div></div></div> </div>

<div class="clearfix">
 <div class="links"></div>

</div>

</div>

</div>

</div>

</div>

</div>

 <div class="region region-footer">

 <div id="block-block-11" class="block block-block clearfix">

<div class="content">

 <p style="text-align:center">Help shape the future of the Open Source Initiative...
visit and participate in the OSI wiki.</p>

<div>

<a href="https://twitter.com/OpenSourceOrg" class="twitter-follow-button" data-show-count="false" data-
lang="en">Follow @OpenSourceOrg

<script>

<!--/--><![CDATA[// ><!--

!function(d,s,id){var

js,fjs=d.getElementsByTagName(s)[0];if(!d.getElementById(id)){js=d.createElement(s);js.id=id;js.src="//platform.t
witter.com/widgets.js";fjs.parentNode.insertBefore(js,fjs);}(document,"script","twitter-wjs");

//--><![]]>

</script></div>

<p>

<!-- Creative Commons License -->
Opensource.org site content is licensed under a <a rel="license"
href="http://creativecommons.org/licenses/by/4.0/">Creative Commons Attribution 4.0 International
License.<!-- /Creative Commons License -->

<!-- <rdf:RDF xmlns="http://web.resource.org/cc/" xmlns:dc="http://purl.org/dc/elements/1.1/"

```
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
<Work rdf:about="">
  <license rdf:resource="http://creativecommons.org/licenses/by/3.0/" />
</Work>
<License rdf:about="http://creativecommons.org/licenses/by/3.0/"><permits
rdf:resource="http://web.resource.org/cc/Reproduction"/><permits
rdf:resource="http://web.resource.org/cc/Distribution"/><requires
rdf:resource="http://web.resource.org/cc/Notice"/><requires
rdf:resource="http://web.resource.org/cc/Attribution"/><permits
rdf:resource="http://web.resource.org/cc/DerivativeWorks"/></License></rdf:RDF>
-->
```

```
| <a href=" ../ToS">Terms of Service</a>
```

```
</p>
</div>
</div>
<div id="block-block-7" class="block block-block clearfix">
```

```
<div class="content">
  <script src="http://www.google-analytics.com/urchin.js" type="text/javascript">
<!--/--><![CDATA[// ><!--
```

```
//--><!]]>
</script><script type="text/javascript">
<!--/--><![CDATA[// ><!--
```

```
_uacct = "UA-3916956-1";
urchinTracker();
```

```
//--><!]]>
</script> </div>
</div>
</div>
  </div></div></div></div> <!-- /.left-corner, /.right-corner, /#squeeze, /#center -->
```

```
</div> <!-- /#container -->
</div> <!-- /#wrapper -->
</body>
</html>
```

```
=====
== NOTICE file corresponding to the section 4 d of           ==
== the Apache License, Version 2.0,                          ==
== in this case for the Gradle distribution.                  ==
=====
```

This product includes software developed by
The Apache Software Foundation (<http://www.apache.org/>).

It includes the following other software:

Groovy (<http://groovy-lang.org>)

SLF4J (<http://www.slf4j.org>)

Junit (<http://www.junit.org>)

JCIFS (<http://jcifs.samba.org>)

For licenses see the LICENSE file.

If any software distributed with Gradle does not have an Apache 2 License, its license is explicitly listed in the LICENSE file.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation,

and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s)

with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

=====
Gradle Subcomponents:

License for the slf4j package

SLF4J License

Copyright (c) 2004-2007 QOS.ch
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be

included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

These terms are identical to those of the MIT License, also called the X License or the X11 License, which is a simple, permissive non-copyleft free software license. It is deemed compatible with virtually all types of licenses, commercial or otherwise. In particular, the Free Software Foundation has declared it compatible with GNU GPL. It is also known to be approved by the Apache Software Foundation as compatible with Apache Software License.

License for the JUnit package

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS COMMON PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and

b) in the case of each subsequent Contributor:

i) changes to the Program, and

ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents " mean patent claims licensable by a Contributor which are

necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:

i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and

iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and

b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order

to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against a Contributor with respect to a patent applicable to software (including a cross-claim or counterclaim in a

lawsuit), then any patent licenses granted by that Contributor to such Recipient under this Agreement shall terminate as of the date such litigation is filed. In addition, if Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. IBM is the initial Agreement Steward. IBM may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

License for the JCIFS package

JCIFS License

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some
specially designated software packages--typically libraries--of the
Free Software Foundation and other authors who decide to use it. You
can use it too, but we suggest you first think carefully about whether
this license or the ordinary General Public License is the better
strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use,
not price. Our General Public Licenses are designed to make sure that
you have the freedom to distribute copies of free software (and charge
for this service if you wish); that you receive source code or can get
it if you want it; that you can change the software and use pieces of
it in new free programs; and that you are informed that you can do
these things.

To protect your rights, we need to make restrictions that forbid
distributors to deny you these rights or to ask you to surrender these
rights. These restrictions translate to certain responsibilities for
you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis
or for a fee, you must give the recipients all the rights that we gave
you. You must make sure that they, too, receive or can get the source
code. If you link other code with the library, you must provide
complete object files to the recipients, so that they can relink them
with the library after making changes to the library and recompiling
it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the
library, and (2) we offer you this license, which gives you legal
permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in

non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does

and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based

on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and

therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License.

Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

- a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the

user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any

particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU

Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

License for the JGit package

Copyright (c) 2007, Eclipse Foundation, Inc. and its licensors.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of the Eclipse Foundation, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE

OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright \${project.inceptionYear} - \${current.year} \${copyright.holder}

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

```
<html>
```

```
<head>
```

```
<meta http-equiv="content-type"
content="text/html; charset=ISO-8859-1">
```

```
<title>CDDL ver. 1.0</title>
```

```
<meta name="author" content="Cliff Allen">
```

```
</head>
```

```
<body>
```

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

1.

Definitions.

1.1. Contributor means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable means the

Covered Software in any form other than
Source Code.

1.5. *Initial Developer* means
the individual or entity that first makes
Original Software available under this License.

1.6. *Larger Work* means a work
which combines Covered Software or
portions thereof with code not governed by the terms of this License.

1.7. *License* means this
document.

1.8. *Licensable* means having
the right to grant, to the maximum extent
possible, whether at the time of the initial grant or subsequently
acquired, any and all of the rights conveyed herein.

1.9. *Modifications* means the
Source Code and Executable form of any of
the following:

A. Any file that results from an addition to, deletion from or
modification of the contents of a file containing Original Software or
previous Modifications;

B. Any new file that contains any part of the Original Software or
previous Modification; or

C. Any new file that is contributed or otherwise made available under
the terms of this License.

1.10. *Original Software* means
the Source Code and Executable form of
computer software code that is originally released under this License.

1.11. *Patent Claims* means any
patent claim(s), now owned or hereafter
acquired, including without limitation, method, process, and apparatus
claims, in any patent Licensable by grantor.

1.12. *Source Code* means (a)

the common form of computer software code
in which modifications are made and (b) associated documentation
included in or with such code.

1.13. *You (or Your)* means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof);

 (c) The licenses granted in Sections 2.1(a) and (b) are effective on

the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License;

 (d) Notwithstanding Section 2.1(b) above, no patent license is granted:

(1) for code that You delete from the Original Software, or

(2) for

infringements caused by: (i) the modification of the Original Software,
or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted:

(1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications

of Contributor Version, or (ii) the combination of Modifications made

by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by

that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipients rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against

such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a commercial item, as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of commercial computer software

(as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer software documentation as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48

C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the

law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdictions conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions).

Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

</body>

</html>

<!DOCTYPE html>

<html>

<head>

<title>jsoup License</title>

<meta name="keywords" content="license, open source, mit">

<meta name="description" content="jsoup is licensed under the MIT open source license">

<link type="text/css" rel="stylesheet" href="/rez/style.css">

<script type="text/javascript">

var _gaq = _gaq || [];

_gaq.push(['_setAccount', 'UA-89734-10']);

_gaq.push(['_setDomainName', 'jsoup.org']);

_gaq.push(['_trackPageview']);

```

(function() {
  var ga = document.createElement('script'); ga.type = 'text/javascript'; ga.async = true;
  ga.src = ('https:' == document.location.protocol ? 'https://ssl' : 'http://www') + '.google-analytics.com/ga.js';
  var s = document.getElementsByTagName('script')[0]; s.parentNode.insertBefore(ga, s);
})();
</script>
</head>
<body class="n1-">
<div class="wrap">
<div class="header">
<div class="nav-sections">
<ul>
<li class="n1-home"><h4><a href="/">jsoup</a></h4></li>
<li class="n1-news"><a href="/news/">News</a></li>
<li class="n1-bugs"><a href="/bugs">Bugs</a></li>
<li class="n1-discussion"><a href="/discussion">Discussion</a></li>
<li class="n1-download"><a href="/download">Download</a></li>
<li class="n1-api"><a href="/apidocs/">API Reference</a></li>
<li class="n1-cookbook"><a href="/cookbook/">Cookbook</a></li>
<li class="n1-try"><a href="http://try.jsoup.org/">Try jsoup</a></li>
</ul>
</div>
</div>
<div class="breadcrumb">
<a href="/">jsoup</a>
<span class="seperator">&raquo;</span> jsoup License
</div>
<div class="content">
<div class="col1">
<h1>jsoup License</h1>
<p>The jsoup code-base (include source and compiled packages) are distributed under the open source MIT license as described below.</p>
<h3>The MIT License</h3>
<p>Copyright &copy; 2009 - 2014 <a href="http://jonathanhedley.com">Jonathan Hedley</a> (<a href="mailto:jonathan@hedley.net">jonathan@hedley.net</a>)</p>
<p>Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:</p>
<p>The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.</p>
<p>THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.</p>

```

```

</div>
<!-- /col1 -->
<div class="col2">
</div>
<!-- /col2 -->
</div>
<!-- /content-->
<div class="footer">
<b>jsoup HTML parser</b> &copy; 2009 - 2014
<a href="http://jhy.io/" rel="author"><b>Jonathan Hedley</b></a>
</div>
</div>
<!-- /wrap -->
<script src="/rez/prettify.js"></script>
<script>prettyPrint();</script>
</body>
</html>
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
<title>Eclipse Public License - Version 1.0</title>
<style type="text/css">
body {
size: 8.5in 11.0in;
margin: 0.25in 0.5in 0.25in 0.5in;
tab-interval: 0.5in;
}
p {
margin-left: auto;
margin-top: 0.5em;
margin-bottom: 0.5em;
}
p.list {
margin-left: 0.5in;
margin-top: 0.05em;
margin-bottom: 0.05em;
}
</style>

</head>

<body lang="EN-US">

<h2>Eclipse Public License - v 1.0</h2>

```

<p>THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.</p>

<p>1. DEFINITIONS</p>

<p>"Contribution" means:</p>

<p class="list">a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and</p>

<p class="list">b) in the case of each subsequent Contributor:</p>

<p class="list">i) changes to the Program, and</p>

<p class="list">ii) additions to the Program;</p>

<p class="list">where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.</p>

<p>"Contributor" means any person or entity that distributes the Program.</p>

<p>"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.</p>

<p>"Program" means the Contributions distributed in accordance with this Agreement.</p>

<p>"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.</p>

<p>2. GRANT OF RIGHTS</p>

<p class="list">a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.</p>

<p class="list">b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide,

royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.</p>

<p class="list">c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.</p>

<p class="list">d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.</p>

<p>3. REQUIREMENTS</p>

<p>A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:</p>

<p class="list">a) it complies with the terms and conditions of this Agreement; and</p>

<p class="list">b) its license agreement:</p>

<p class="list">i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;</p>

<p class="list">ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;</p>

<p class="list">iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other

party; and

iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and

b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial

Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

<p>All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.</p>

<p>Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.</p>

<p>This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.</p>

</body>

</html>

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems,

and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and

limitations under the License.

License for the slf4j package

SLF4J License

Copyright (c) 2004-2007 QOS.ch
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

These terms are identical to those of the MIT License, also called the X License or the X11 License, which is a simple, permissive non-copyleft free software license. It is deemed compatible with virtually all types of licenses, commercial or otherwise. In particular, the Free Software Foundation has declared it compatible with GNU GPL. It is also known to be approved by the Apache Software Foundation as compatible with Apache Software License.

License for the JUnit package

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS COMMON PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and

b) in the case of each subsequent Contributor:

i) changes to the Program, and

ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents " mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any

Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:

i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and

iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and

b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against a Contributor with respect to a patent applicable to software (including a cross-claim or counterclaim in a lawsuit), then any patent licenses granted by that Contributor to such Recipient under this Agreement shall terminate as of the date such litigation is filed. In addition, if Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. IBM is the initial Agreement Steward. IBM may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its

Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

License for the JCIFS package

JCIFS License

GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of

it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d) If a facility in the modified Library refers to a function or a

table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or

link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for

reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

- a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.
- b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the

original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version,

but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that

everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the library's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>
```

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

```
<signature of Ty Coon>, 1 April 1990  
Ty Coon, President of Vice
```

That's all there is to it!

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">  
<html lang="en">  
<head>  
<title>Apache License, Version 2.0</title>  
  
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  
<meta property="og:image" content="http://www.apache.org/images/asf_logo.gif" />  
  
<link rel="stylesheet" type="text/css" media="screen" href="/css/style.css">
```


<p>"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.</p>

<p>"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.</p>

<p>"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.</p>

<p>"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.</p>

<p>"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.</p>

<p>"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.</p>

<p>"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).</p>

<p>"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.</p>

<p>"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."</p>

<p>"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.</p>

<p>2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You

a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

-

- You must give any other recipients of the Work or Derivative Works a copy of this License; and

- You must cause any modified files to carry prominent notices stating that You changed the files; and

- You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

- If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

<p>5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.</p>

<p>6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.</p>

<p>7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.</p>

<p>8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.</p>

<p>9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf

and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.</p>

<p>END OF TERMS AND CONDITIONS</p>

<h1 id="apply">APPENDIX: How to apply the Apache License to your work</h1>

<p>To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.</p>

<div class="codehilite"><pre>Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

</pre></div></div></div>

<div class="clear"></div>

</div>

<div id="footer" class="container_16">

<div class="links grid_16">

<div class="grid_3">

<h4>Projects</h4>

HTTP Server

Abdera

Accumulo

ACE

ActiveMQ

Airavata

Allura

Ambari

Ant

Any23

APR

- Archiva
- Aries
- Avro
- Axis
- Bigtop
- Bloodhound
- Buildr
- BVal
- Camel
- Cassandra
- Cayenne
- Chemistry
- Chukwa
- Clerezza
- CloudStack
- Cocoon
- Commons
- Continuum
- Cordova
- CouchDB
- Creadur
- Crunch
- cTAKES
- Curator
- CXF
- DB
- Deltacloud
- DeltaSpike
- DirectMemory
- Directory
- Empire-db
- Etch

- Felix
- Flex
- Flume
- Forrest
- Geronimo
- Giraph
- Gora
- Gump
- Hadoop
- Hama
- HBase
- Helix
- Hive
- HttpComponents
- Isis
- Jackrabbit
- James
- jclouds
- Jena
- JMeter
- JSPWiki
- jUDDI
- Kafka
- Karaf
- Knox
- Lenya
- Libcloud
- Logging
- Lucene
- Lucene.Net
- Lucy
- Mahout
- ManifoldCF

- Marmotta
- Maven
- Mesos
- MINA
- MRUnit
- MyFaces
- Nutch
- ODE
- OFBiz
- Olingo
- Oltu
- Onami
- OODT
- Oozie
- Open Climate Workbench
- OpenJPA
- OpenMeetings
- OpenNLP
- OpenOffice
- OpenWebBeans
- PDFBox
- Perl
- Pig
- Pivot
- POI
- Portals
- Qpid
- Rave
- River
- Roller
- Santuario
- ServiceMix
- Shindig
- Shiro
- SIS

- Sling
- SpamAssassin
- Spark
- Sqoop
- Stanbol
- STeVe
- Storm
- Struts
- Subversion
- Synapse
- Syncope
- Tajo
- Tapestry
- Tcl
- Tez
- Thrift
- Tika
- Tiles
- Tomcat
- TomEE
- Traffic Server
- Turbine
- Tuscany
- UIMA
- VCL
- Velocity
- VXQuery
- Web Services
- Whirr
- Wicket
- Wink
- Wookie
- Xalan

```
<li><a href="http://xerces.apache.org/" title="XML parsers in Java, C++ and Perl">Xerces</a></li>
<li><a href="http://xmlbeans.apache.org/" title="XML-Java binding tool">XMLBeans</a></li>
<li><a href="http://xmlgraphics.apache.org/" title="Conversion from XML to graphical output">XML
Graphics</a></li>
<li><a href="http://zookeeper.apache.org/" title="Centralized service for maintaining configuration
information">ZooKeeper</a></li>
</ul>
</div>
<div class="grid_3">
  <h4>Foundation</h4>
  <ul>
    <li><a href="/foundation/faq.html">FAQ</a></li>
    <li><a href="/foundation/glossary.html">Glossary</a></li>
    <li><a href="/licenses/" title="Overview of the Apache Licenses">Licenses</a></li>
    <li><a href="/foundation/marks/" title="Apache marks policies and listing">Trademarks</a></li>
    <li><a href="/foundation/news.html" title="Official news feed of Foundation announcements">News</a></li>

    <li><a href="/press/" title="Press, Media, and Analyst contact">Press Inquiries</a></li>
    <li><a href="/foundation/records/" title="Formal corporate records and board meeting minutes">Public
Records</a></li>
    <li><a href="/foundation/maillinglists.html" title="Mailing lists and Apache">Mailing Lists</a></li>
    <li><a href="/foundation/sponsorship.html" title="Sponsor the Foundation">Sponsorship</a></li>
    <li><a href="/foundation/contributing.html" title="Donate to the Foundation">Donations</a></li>
    <li><a href="/foundation/buy_stuff.html" title="Buy Apache branded merchandise">Buy Stuff</a></li>
    <li><a href="/foundation/thanks.html" title="Thank you to our Sponsors">Thanks</a></li>
    <li><a href="/foundation/contact.html" title="Contact Us">Contact</a></li>
  </ul>
</div>
<div class="grid_3 suffix_1">
  <h4>Foundation Projects</h4>
  <ul>
    <li><a href="http://attic.apache.org/" title="Inactive projects repository">Attic</a></li>
    <li><a href="/foundation/conferences.html" title="Meetings of developers and users">Conferences</a></li>
    <li><a href="http://community.apache.org/" title="Helping newcomers to the ASF">Community
Development</a></li>
    <li><a href="http://incubator.apache.org/" title="Shepherd for new projects">Incubator</a></li>
    <li><a href="/dev/" title="ASF Infrastructure: Operations and howto documents for PMCs and
contributors">Infrastructure</a></li>
    <li><a href="http://labs.apache.org/" title="The Innovation Laboratories of the Apache Software
Foundation">Labs</a></li>
    <li><a href="/legal/" title="Legal Affairs">Legal Affairs</a></li>
    <li><a href="/press/" title="Public Relations">Public Relations</a></li>
    <li><a href="/security/" title="Security">Security</a></li>
    <li><a href="/travel/" title="Travel Assistance">Travel Assistance</a></li>
  </ul>
</div>
<div class="grid_3">
```

```
<h4>Community</h4>
<ul>
  <li><a href="http://people.apache.org/" title="Apache committer homepages">People</a></li>
  <li><a href="/memorials/" title="In memoriam of past committers">Memorials</a></li>
  <li><a href="http://feathercast.apache.org/" title="Apache Podcasts">Feathercast</a></li>
  <li><a href="http://blogs.apache.org/" title="Apache Project Blogs">Project Blogs</a></li>
  <li><a href="http://planet.apache.org/committers/" title="Apache Committers' Blogs">PlanetApache</a></li>
</ul>
</div>
<div class="grid_3">
  <h4>How It Works</h4>
  <ul>
    <li><a href="/foundation/how-it-works.html">Introduction</a></li>
    <li><a href="/foundation/how-it-works.html#meritocracy">Meritocracy</a></li>
    <li><a href="/foundation/how-it-works.html#structure">Structure</a></li>
    <li><a href="/foundation/how-it-works.html#roles">Roles</a></li>
    <li><a href="/foundation/how-it-works.html#management">Collaboration</a></li>
    <li><a href="/foundation/how-it-works.html#incubator">Incubator</a></li>
    <li><a href="/foundation/how-it-works.html#other">Other entities</a></li>
    <li><a href="/foundation/glossary.html">Glossary</a></li>
    <li><a href="/foundation/voting.html">Voting</a></li>
  </ul>
</div>
</div>
<div class="clear"></div>
</div>
<div id="copyright" class="container_16">
  <p>Copyright © 2012 The Apache Software Foundation, Licensed under the <a
href="http://www.apache.org/licenses/LICENSE-2.0">Apache License, Version 2.0</a>. <br/>Apache and the
Apache feather logo are trademarks of The Apache Software Foundation.</p>
</div>
</body>
</html>
```

Apache Maven Distribution
Copyright 2001-2014 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).
Apache License

Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to

communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of

the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

```
=====
== NOTICE file corresponding to the section 4 d of      ==
== the Apache License, Version 2.0,                    ==
== in this case for the Gradle distribution.            ==
=====
```

This product includes software developed by
The Apache Software Foundation (<http://www.apache.org/>).

It includes the following other software:

Groovy (<http://groovy.codehaus.org>)

SLF4J (<http://www.slf4j.org>)

Junit (<http://www.junit.org>)

JCIFS (<http://jcifs.samba.org>)

For licenses see the LICENSE file.

If any software distributed with Gradle does not have an Apache 2 License, its license is explicitly listed in the LICENSE file.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of

this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only

on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.55 coherence-bom 22.06.1

1.55.1 Available under license :

/*

* Copyright YYYY Dustin Sallings

*

* Licensed under the MIT License, (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.opensource.org/licenses/mit-license.php>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

coherence-grpc-proxy\src\test\proto\route_guide.proto
coherence-grpc-proxy\src\test\java\com\oracle\coherence\grpc\proxy\routeguide\
coherence-http-netty\src\main\java\com\tangosol\coherence\http\netty\NettyInputStream.java
coherence-javadoc\src\main\javadoc\resources\
coherence-json\src\main\java\com\oracle\coherence\io\json\genson\
coherence-json\src\test\java\com\oracle\coherence\io\json\genson\
examples\tutorials\500-graphql\complete\src\main\resources\web\index.html
examples\tutorials\500-graphql\initial\src\main\resources\web\index.html
test\distribution\target\
test\functional\logging\src\test\java\org\slf4j\
tools\ant-contrib\
tools\internal\
tools\maven\
tools\tde\
tools\slt-antutils\
tools\wls\
dependency-reduced-pom.xml
etc\
ExcludeList
.git\
.gradle\
build\
gradlew
gradlew.bat
gradle-wrapper.properties
.idea\
LicensedObject.java
login.config
Makefile
MANIFEST.MF
manifest.mf
\META-INF\services\
module-info.java
.mvn\
mvnw
package.html
\README
security.policy
target\
test-security.policy
cache-config-UTF8-BOM.xml
.chain
.cmd
.cnf

.csr
.data
.exe
.flattened-pom.xml
.gar
.gif
.ico
.iml
.jar
.jks
.jpg
.json
.key
.p12
.pem
.png
.ptn
.ser
.sql
.svg
.txt
.war
.yaml
.zip
.class
.cdb
/*
* Copyright YYYY The Apache Software Foundation
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
@rem
@rem %file%
@rem
@rem Copyright (c) 2000, 2022, Oracle and/or its affiliates. All rights reserved.
@rem
@rem Oracle is a registered trademarks of Oracle Corporation and/or its
@rem affiliates.

@rem
@rem This software is the confidential and proprietary information of Oracle
@rem Corporation. You shall not disclose such confidential and proprietary
@rem information and shall use it only in accordance with the terms of the
@rem license agreement you entered into with Oracle.
@rem
@rem This notice may not be removed or altered.
@rem
/*
* Copyright (c) YYYY Oracle and/or its affiliates. All rights reserved.
*
* This program and the accompanying materials are made available under the
* terms of the Eclipse Public License v. 2.0 which is available at
* <http://www.eclipse.org/legal/epl-2.0>,
* or the Eclipse Distribution License v. 1.0 which is available at
* <http://www.eclipse.org/org/documents/edl-v10.php>.
*
* SPDX-License-Identifier: EPL-2.0 OR BSD-3-Clause
*/
/*
* The contents of this file are subject to the terms
* of the Common Development and Distribution License
* (the "License"). You may not use this file except
* in compliance with the License.
*
* You can obtain a copy of the license at
* [glassfish/bootstrap/legal/CDDLv1.0.txt](https://glassfish.bootstrap/legal/CDDLv1.0.txt) or
* <https://glassfish.dev.java.net/public/CDDLv1.0.html>.
* See the License for the specific language governing
* permissions and limitations under the License.
*
* When distributing Covered Code, include this CDDL
* HEADER in each file and include the License file at
* [glassfish/bootstrap/legal/CDDLv1.0.txt](https://glassfish.bootstrap/legal/CDDLv1.0.txt). If applicable,
* add the following below this CDDL HEADER, with the
* fields enclosed by brackets "[]" replaced with your
* own identifying information: Portions Copyright [yyyy]
* [name of copyright owner]
*/
/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright YYYY Sun Microsystems, Inc. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can obtain

* a copy of the License at <https://glassfish.dev.java.net/public/CDDL+GPL.html>
* or [glassfish/bootstrap/legal/LICENSE.txt](https://glassfish.org/legal/LICENSE.txt). See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at [glassfish/bootstrap/legal/LICENSE.txt](https://glassfish.org/legal/LICENSE.txt).
* Sun designates this particular file as subject to the "Classpath" exception
* as provided by Sun in the GPL Version 2 section of the License file that
* accompanied this code. If applicable, add the following below the License
* Header, with the fields enclosed by brackets [] replaced by your own
* identifying information: "Portions Copyrighted [year]
* [name of copyright owner]"
*
* Contributor(s):
*
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made subject to such option by the copyright
* holder.
*
*
* This file incorporates work covered by the following copyright and
* permission notice:
*
* Licensed to the Apache Software Foundation (ASF) under one or more
* contributor license agreements. See the NOTICE file distributed with
* this work for additional information regarding copyright ownership.
* The ASF licenses this file to You under the Apache License, Version 2.0
* (the "License"); you may not use this file except in compliance with
* the License. You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
/*
* The contents of this file are subject to the terms
* of the Common Development and Distribution License

* (the License). You may not use this file except in
* compliance with the License.
*
* You can obtain a copy of the license at
* <https://glassfish.dev.java.net/public/CDDLv1.0.html> or
* glassfish/bootstrap/legal/CDDLv1.0.txt.
* See the License for the specific language governing
* permissions and limitations under the License.
*
* When distributing Covered Code, include this CDDL
* Header Notice in each file and include the License file
* at glassfish/bootstrap/legal/CDDLv1.0.txt.
* If applicable, add the following below the CDDL Header,
* with the fields enclosed by brackets [] replaced by
* you own identifying information:
* "Portions Copyrighted [year] [name of copyright owner]"
*
* Copyright YYYY Sun Microsystems, Inc. All rights reserved.
*/
/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) YYYY Oracle and/or its affiliates. All rights reserved.
*
* Oracle licenses this file to You under the Apache License, Version 2.0
* (the "License"); you may not use this file except in compliance with
* the License. You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
Copyright (c) 2000, 2020, Oracle and/or its affiliates.

The Universal Permissive License (UPL), Version 1.0

Subject to the condition set forth below, permission is hereby granted to any person obtaining a copy of this software, associated documentation and/or data (collectively the "Software"), free of charge and under any and all copyright rights in the Software, and any and all patent rights owned or freely licensable by each licensor hereunder covering either (i) the unmodified Software as contributed to or provided by such licensor, or (ii) the Larger Works (as defined below), to deal in both

(a) the Software, and
(b) any piece of software and/or hardware listed in the lrgwrks.txt file if one is included with the Software (each a "Larger Work" to which the Software is contributed by such licensors),

without restriction, including without limitation the rights to copy, create derivative works of, display, perform, and distribute the Software and make, use, sell, offer for sale, import, export, have made, and have sold the Software and the Larger Work(s), and to sublicense the foregoing rights on either these or other terms.

This license is subject to the following condition:

The above copyright notice and either this complete permission notice or at a minimum a reference to the UPL must be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

/*

* Licensed to the Apache Software Foundation (ASF) under one or more
* contributor license agreements. See the NOTICE file distributed with
* this work for additional information regarding copyright ownership.
* The ASF licenses this file to You under the Apache License, Version 2.0
* (the "License"); you may not use this file except in compliance with
* the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.

*/

/*

* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.

*

* Copyright YYYY Sun Microsystems, Inc. All rights reserved.

*

* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can obtain

* a copy of the License at <https://glassfish.dev.java.net/public/CDDL+GPL.html>
* or [glassfish/bootstrap/legal/LICENSE.txt](https://glassfish.org/legal/LICENSE.txt). See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at [glassfish/bootstrap/legal/LICENSE.txt](https://glassfish.org/legal/LICENSE.txt).
* Sun designates this particular file as subject to the "Classpath" exception
* as provided by Sun in the GPL Version 2 section of the License file that
* accompanied this code. If applicable, add the following below the License
* Header, with the fields enclosed by brackets [] replaced by your own
* identifying information: "Portions Copyrighted [year]
* [name of copyright owner]"
*
* Contributor(s):
*
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made subject to such option by the copyright
* holder.
*
*
* This file incorporates work covered by the following copyright and
* permission notice:
*
* Copyright YYYY The Apache Software Foundation
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
/*
* %file%
*
* Copyright (c) 2000, 2022, Oracle and/or its affiliates. All rights reserved.

*
 * Oracle is a registered trademarks of Oracle Corporation and/or its
 * affiliates.
 *
 * This software is the confidential and proprietary information of Oracle
 * Corporation. You shall not disclose such confidential and proprietary
 * information and shall use it only in accordance with the terms of the
 * license agreement you entered into with Oracle.
 *
 * This notice may not be removed or altered.
 */
 /*
 * Copyright (c) YYYY Oracle and/or its affiliates. All rights reserved.
 *
 * This program and the accompanying materials are made available under the
 * terms of the Eclipse Public License v. 2.0, which is available at
 * <http://www.eclipse.org/legal/epl-2.0>.
 *
 * This Source Code may also be made available under the following Secondary
 * Licenses when the conditions for such availability set forth in the
 * Eclipse Public License v. 2.0 are satisfied: GNU General Public License,
 * version 2 with the GNU Classpath Exception, which is available at
 * <https://www.gnu.org/software/classpath/license.html>.
 *
 * SPDX-License-Identifier: EPL-2.0 OR GPL-2.0 WITH Classpath-exception-2.0
 */
 -----BEGIN CERTIFICATE-----
 MIIDHTCCAatsCBEURqEEwCwYHKoZlZjgEAwUAMHMxCzAJBgNVBAYTAIVTMQswCQYDVQQIEwJNQTE
 ET
 MBEGA1UEBxMKU29tZXJ2aWxsZTEXMBUGA1UEChMOVGFuZ29zb2wsIEluYy4xEDAOBgNVBAsTB1Vu
 a25vd24xZzAVBgNVBAMTDiRhbmdvc29sLCBJbmMuMCAXDTA2MDkyMDIwNDQ0OV0oYDZlYODAwNzA1
 MjA0NDQ5WjBzMQswCQYDVQQGEwJVUzELMAkGA1UECBMCTUEExEzARBgNVBAcTCiNvbWVydmlsb
 GUx
 FzAVBgNVBAoTDiRhbmdvc29sLCBJbmMuMRAwDgYDVQQLEwdVbmtub3duMRcwFQYDVQQDEw5UYW
 5n
 b3NvbCwgSW5jLjCCAbgwgEsBgcqhkJOOAQBMiIBHwKBgQD9f1OBHXUSKVLfSpwu7OTn9hG3Ujzv
 RADDHj+AtlEmaUVdQCJR+1k9jVj6v8X1ujD2y5tVbNeBO4AdNG/yZmC3a5lQpaSfn+gEexAiwk+7
 qdf+t8Yb+DtX58aophUPBPuD9tPFHsMCNVQTWhaRMvZ1864rYdcq7/IiAxmd0UgBxwIVAjdGUI8V
 IwvMspK5gqLrhAvwWBz1AoGBAPfhoIXWmz3ey7yrXDa4V7l5lK+7+jrqgv1XTAs9B4JnUVIXjrrU
 WU/mcQcQgYC0SRZxI+hMKBYTt88JMozIpuE8FnqLVHYNKOCjrh4rs6Z1kW6jfwv6ITVi8ftiegEk
 O8yk8b6oUZCJqIPf4VrlnwaSi2ZegHtVJWQBTDv+z0kqA4GFAAKBgQDclms4MK7gp9Ybx7lznEKG
 Z0VrExC3xFISJ8luNVTIebcqQ1Tq9TQG2BSi4SbiMzFcpPy6vvA79dpK1mMx8HUPc9fksbnwKr5
 9MJt53K0UvlsZ+w/x//j1mX/DIZvKxBQRInOp1lrCquYJWETYBpknuKecjiqJuMVU6vmHsqJ+TAL
 BgcqhkJOOAQDBQADLwAwLAIUthsTUE1Ht9Vd4dC4n3TE9oMJeeUCFCElz4I8DH2Qc+IDH8hPfNgD
 97aA
 -----END CERTIFICATE-----
 /*
 * DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.

*
 * Copyright (c) YYYY Oracle and/or its affiliates. All rights reserved.
 *
 * The contents of this file are subject to the terms of either the GNU
 * General Public License Version 2 only ("GPL") or the Common Development
 * and Distribution License("CDDL") (collectively, the "License"). You
 * may not use this file except in compliance with the License. You can
 * obtain a copy of the License at
 * https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
 * or packager/legal/LICENSE.txt. See the License for the specific
 * language governing permissions and limitations under the License.
 *
 * When distributing the software, include this License Header Notice in each
 * file and include the License file at packager/legal/LICENSE.txt.
 *
 * GPL Classpath Exception:
 * Oracle designates this particular file as subject to the "Classpath"
 * exception as provided by Oracle in the GPL Version 2 section of the License
 * file that accompanied this code.
 *
 * Modifications:
 * If applicable, add the following below the License Header, with the fields
 * enclosed by brackets [] replaced by your own identifying information:
 * "Portions Copyright [year] [name of copyright owner]"
 *
 * Contributor(s):
 * If you wish your version of this file to be governed by only the CDDL or
 * only the GPL Version 2, indicate your decision by adding "[Contributor]
 * elects to include this software in this distribution under the [CDDL or GPL
 * Version 2] license." If you don't indicate a single choice of license, a
 * recipient has the option to distribute your version of this file under
 * either the CDDL, the GPL Version 2 or to extend the choice of license to
 * its licensees as provided above. However, if you add GPL Version 2 code
 * and therefore, elected the GPL Version 2 license, then the option applies
 * only if the new code is made subject to such option by the copyright
 * holder.
 */

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

1. Definitions.

1.1. Contributor. means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version. means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software. means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable. means the Covered Software in any form other than Source Code.

1.5. Initial Developer. means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work. means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License. means this document.

1.8. Licensable. means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications. means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software. means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims. means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code. means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You. (or .Your.) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, .You. includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, .control. means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You

represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN .AS IS. BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as .Participant.) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH

PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY.S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a .commercial item., as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of .commercial computer software. (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and .commercial computer software documentation. as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction.s conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys. fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307
USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying

it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on

the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as

to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.

Copyright (C)

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author

Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

"CLASSPATH" EXCEPTION TO THE GPL VERSION 2

Certain source files distributed by Sun Microsystems, Inc. are subject to the following clarification and special exception to the GPL Version 2, but only where Sun has expressly included in the particular source file's header the words

"Sun designates this particular file as subject to the "Classpath" exception as provided by Sun in the License file that accompanied this code."

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License Version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module.? An independent module is a module which is not derived from or based on this library.? If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so.? If you do not wish to do so, delete this exception statement from your version.

route_guide.proto

coherence-grpc-proxy/src/test/java/com/oracle/coherence/grpc/proxy/routeguide/

coherence-http-netty/src/main/java/com/tangosol/coherence/http/netty/NettyInputStream.java

coherence-javadoc/src/main/javadoc/resources/

coherence-json/src/main/java/com/oracle/coherence/io/json/genson/

coherence-json/src/test/java/com/oracle/coherence/io/json/genson/

examples/tutorials/500-graphql/complete/src/main/resources/web/index.html

examples/tutorials/500-graphql/initial/src/main/resources/web/index.html

test/distribution/target/

test/functional/logging/src/test/java/org/slf4j/

tools/ant-contrib/

tools/internal/

tools/maven/

tools/tde/

tools/tlt-antutils/
tools/wls/
dependency-reduced-pom.xml
.Dockerfile
etc/
ExcludeList
.git/
.gradle/
build/
gradlew
gradlew.bat
gradle-wrapper.properties
.idea/
LicensedObject.java
login.config
Makefile
MANIFEST.MF
manifest.mf
/META-INF/services/
module-info.java
.mvn/
mvnw
package.html
/README
security.policy
target/
test-security.policy
cache-config-UTF8-BOM.xml
.chain
.cmd
.cnf
.csr
.data
.exe
.flattened-pom.xml
.gar
.gif
.ico
.iml
.jar
.jks
.jpg
.json
.key
.p12
.pem
.png
.ptn

.ser
.sql
.svg
.txt
.war
.yaml
.zip
.class
.cdb
/*

* Copyright YYYY Sun Microsystems, Inc. All Rights Reserved.

*

* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:

*

* - Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.

*

* - Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.

*

* - Neither the name of Sun Microsystems nor the names of its
* contributors may be used to endorse or promote products derived
* from this software without specific prior written permission.

*

* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
* IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
* THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
* CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
* EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
* PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
* PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
* LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
* NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
* SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

Apache Log4j API

Copyright 1999-2022 The Apache Software Foundation

This product includes software developed at

The Apache Software Foundation (<http://www.apache.org/>).

Copyright (c) 2000, 2022, Oracle and/or its affiliates. All rights reserved.

Redistribution and use in source and binary forms, with or without

modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement:

This product includes software developed by Oracle

4. Neither the name of Oracle nor the names of its employees may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY ORACLE AND ANY CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL TANGOSOL OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

/*

* Copyright (c) YYYY World Wide Web Consortium,

*

* (Massachusetts Institute of Technology, European Research Consortium for

* Informatics and Mathematics, Keio University). All Rights Reserved. This

* work is distributed under the W3C(r) Software License [1] in the hope that

* it will be useful, but WITHOUT ANY WARRANTY; without even the implied

* warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

*

* [1] <http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231>

*/

Third Party Attributions

The following software (or certain identified files distributed with the software) may be included in this product. Unless otherwise specified, the software identified in this file is licensed under the licenses described below. The disclaimers and copyright notices provided are based on information made available to Oracle by the third party licensors listed.

The following applies to all products licensed under the Apache 2.0 License:

You may not use the identified files except in compliance with the Apache License, Version 2.0 (the "License.")

You may obtain a copy of the License at <http://www.apache.org/licenses/LICENSE-2.0>.
A copy of the license is also reproduced below.

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or

Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work

or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work

by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

=====
MVFlex Expression Language (MVFL)

=====
MVFL 2.0
Copyright (C) 2007 The Codehaus
Mike Brock, Dhanji Prasanna, John Graham, Mark Proctor
Apache License, Version 2.0

=====
jackson-annotations

=====
Jackson Annotations
Copyright (c) 2019 Tatu Saloranta <tatu.saloranta@iki.fi>
Apache License, Version 2.0

=====
jackson-jaxrs-base

=====
jackson-jaxrs-base
COPYRIGHT: Copyright FasterXML.com
Apache License, Version 2.0

=====
jackson-jaxrs-json-provider

Jackson JAXRS JSON
Copyright (c) 2019 Tatu Saloranta <tatu.saloranta@iki.fi>
Copyright (c) Fasterxml
Apache License, Version 2.0

=====
jackson-module-jaxb-annotations
=====

jackson-module-jaxb-annotations
COPYRIGHT: Copyright (c) 2007- Tatu Saloranta, tatu.saloranta@iki.fi
Apache License, Version 2.0

=====
OpenTracing API for Java
=====

opentracing-util: 0.33.0, Apache 2.0
opentracing-mock: 0.33.0, Apache 2.0
opentracing-api: 0.33.0, Apache 2.0
opentracing-noop: 0.33.0, Apache 2.0
Copyright 2016-2019 The OpenTracing Authors
Apache License, Version 2.0

=====
Apache log4j 1.2.17
=====

Apache log4j
Copyright 2010 The Apache Software Foundation
Apache License, Version 2.0

=====
Apache log4j 2.11.1
=====

Apache log4j
COPYRIGHT: Copyright 1999-2017 Apache Software Foundation
LICENSE: Apache 2.0

=====
Modules:
log4j-1.2-api
log4j-api-java9
log4j-api
log4j-appserver
log4j-bom
log4j-cassandra
log4j-core-its
log4j-core-java9
log4j-core
log4j-couchdb

log4j-distribution
log4j-flume-ng
log4j-iostreams
log4j-jcl
log4j-jdbc-dbc2
log4j-jmx-gui
log4j-jpa
log4j-jul
log4j-liquibase
log4j-mongodb2
log4j-mongodb3
log4j-osgi
log4j-perf
log4j-samples
log4j-slf4j-impl
log4j-slf4j18-impl
log4j-taglib
log4j-to-slf4j
log4j-web

=====

4P Dependencies

FROM NOTICE FILE:

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

ResolverUtil.java
Copyright 2005-2006 Tim Fennell
LICENSE: Apache 2.0 <http://www.apache.org/licenses/LICENSE-2.0>

Dumbster SMTP test server
Copyright 2004 Jason Paul Kitchen
LICENSE: Apache 2.0 <http://www.apache.org/licenses/LICENSE-2.0>

TypeUtil.java
Copyright 2002-2012 Ramnivas Laddad, Juergen Hoeller, Chris Beams
LICENSE: Apache 2.0 <http://www.apache.org/licenses/LICENSE-2.0>

picocli (<http://picocli.info>)
Copyright 2017 Remko Popma
LICENSE: Apache 2.0 <http://www.apache.org/licenses/LICENSE-2.0>

FROM POM FILES (Compile Dependencies Only)

com.lmax disruptor 3.4.2

Copyright 2011 - 2018 LMAX Ltd.

LICENSE: Apache 2.0 <https://github.com/LMAX-Exchange/disruptor/blob/3.4.2/LICENCE.txt>

org.springframework spring-aop 3.2.18.RELEASE

COPYRIGHT: Copyright 2002-2018 the original author or authors.

LICENSE: Apache 2.0 <http://www.apache.org/licenses/LICENSE-2.0>

slf4j-api 1.7.25

slf4j-ext 1.7.25

COPYRIGHT and LICENSE:

Copyright (c) 2004-2017 QOS.ch

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=====

Maven

=====

Apache Maven

Copyright 2001-2018 The Apache Software Foundation

Apache License, Version 2.0

This product includes software developed at The Apache Software Foundation (<http://www.apache.org/>).

////////////////////////////////////

DEPENDENCIES

////////////////////////////////////

AOP alliance (<http://aopalliance.sourceforge.net>) aopalliance:aopalliance:jar:1.0

License: Public Domain

JSR-250 Common Annotations for the Java™ Platform

(<http://jcp.org/aboutJava/communityprocess/final/jsr250/index.html>) javax.annotation:jsr250-api:jar:1.0

Copyright 2005-2006 Sun Microsystems, Inc. All Rights Reserved.

License: COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

<https://glassfish.java.net/public/CDDLv1.0.html> (lib/jsr250-api.license)

CDI APIs (<http://www.seamframework.org/Weld/cdi-api>) javax.enterprise:cdi-api:jar:1.0

Copyright 2008, Red Hat Middleware LLC, and individual contributors by the @authors tag. See the copyright.txt in the distribution for a full listing of individual contributors.

License: Apache License, Version 2.0 <http://www.apache.org/licenses/LICENSE-2.0> (lib/cdi-api.license)

Maven Aether Provider (<http://maven.apache.org/ref/3.2.5/maven-aether-provider>) org.apache.maven:maven-aether-provider:jar:3.2.5

Copyright 2001-2014 The Apache Software Foundation

License: Apache License, Version 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt> (lib/maven-aether-provider.license)

Maven Artifact (<http://maven.apache.org/ref/3.2.5/maven-artifact>) org.apache.maven:maven-artifact:jar:3.2.5

Copyright 2001-2014 The Apache Software Foundation

License: Apache License, Version 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt> (lib/maven-artifact.license)

Maven Compat (<http://maven.apache.org/ref/3.2.5/maven-compat>) org.apache.maven:maven-compat:jar:3.2.5

Copyright 2001-2014 The Apache Software Foundation

License: Apache License, Version 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt> (lib/maven-compat.license)

Maven Core (<http://maven.apache.org/ref/3.2.5/maven-core>) org.apache.maven:maven-core:jar:3.2.5

Copyright 2001-2014 The Apache Software Foundation

License: Apache License, Version 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt> (lib/maven-core.license)

Maven Embedder (<http://maven.apache.org/ref/3.2.5/maven-embedder>) org.apache.maven:maven-embedder:jar:3.2.5

Copyright 2001-2014 The Apache Software Foundation

License: Apache License, Version 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt> (lib/maven-embedder.license)

Maven Model (<http://maven.apache.org/ref/3.2.5/maven-model>) org.apache.maven:maven-model:jar:3.2.5
Copyright 2001-2014 The Apache Software Foundation
License: Apache License, Version 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt> (lib/maven-model.license)

Maven Model Builder (<http://maven.apache.org/ref/3.2.5/maven-model-builder>) org.apache.maven:maven-model-builder:jar:3.2.5
Copyright 2001-2014 The Apache Software Foundation
License: Apache License, Version 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt> (lib/maven-model-builder.license)

Maven Plugin API (<http://maven.apache.org/ref/3.2.5/maven-plugin-api>) org.apache.maven:maven-plugin-api:jar:3.2.5
Copyright 2001-2014 The Apache Software Foundation
License: Apache License, Version 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt> (lib/maven-plugin-api.license)

Maven Repository Metadata Model (<http://maven.apache.org/ref/3.2.5/maven-repository-metadata>) org.apache.maven:maven-repository-metadata:jar:3.2.5
Copyright 2001-2014 The Apache Software Foundation
License: Apache License, Version 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt> (lib/maven-repository-metadata.license)

Maven Settings (<http://maven.apache.org/ref/3.2.5/maven-settings>) org.apache.maven:maven-settings:jar:3.2.5
Copyright 2001-2014 The Apache Software Foundation
License: Apache License, Version 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt> (lib/maven-settings.license)

Maven Settings Builder (<http://maven.apache.org/ref/3.2.5/maven-settings-builder>) org.apache.maven:maven-settings-builder:jar:3.2.5
Copyright 2001-2014 The Apache Software Foundation
License: Apache License, Version 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt> (lib/maven-settings-builder.license)

Apache Maven Wagon :: Providers :: File Provider (<http://maven.apache.org/wagon/wagon-providers/wagon-file>) org.apache.maven.wagon:wagon-file:jar:2.8
Copyright 2003-2019 The Apache Software Foundation
License: Apache License, Version 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt> (lib/wagon-file.license)

Apache Maven Wagon :: Providers :: HTTP Provider (<http://maven.apache.org/wagon/wagon-providers/wagon-http>) org.apache.maven.wagon:wagon-http:jar:2.8
Copyright 2003-2019 The Apache Software Foundation
License: Apache License, Version 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt> (lib/wagon-http.license)

Apache Maven Wagon :: Providers :: HTTP Shared Library (<http://maven.apache.org/wagon/wagon-providers/wagon-http-shared>) org.apache.maven.wagon:wagon-http-shared:jar:2.8
Copyright 2003-2019 The Apache Software Foundation
License: Apache License, Version 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt> (lib/wagon-http-

shared.license)

Apache Maven Wagon :: API (<http://maven.apache.org/wagon/wagon-provider-api>)

org.apache.maven.wagon:wagon-provider-api:jar:2.8

Copyright 2003-2019 The Apache Software Foundation

License: Apache License, Version 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt> (lib/wagon-provider-api.license)

Aether API (<http://www.eclipse.org/aether/aether-api/>) org.eclipse.aether:aether-api:jar:1.0.0.v20140518

Copyright (c) 2010-2014 Sonatype, Inc.

License: Eclipse Public License, Version 1.0 <http://www.eclipse.org/legal/epl-v10.html> (lib/aether-api.license)

Aether Connector Basic (<http://www.eclipse.org/aether/aether-connector-basic/>) org.eclipse.aether:aether-connector-basic:jar:1.0.0.v20140518

Copyright (c) 2010-2014 Sonatype, Inc.

License: Eclipse Public License, Version 1.0 <http://www.eclipse.org/legal/epl-v10.html> (lib/aether-connector-basic.license)

Aether Implementation (<http://www.eclipse.org/aether/aether-impl/>) org.eclipse.aether:aether-impl:jar:1.0.0.v20140518

Copyright (c) 2010-2014 Sonatype, Inc.

License: Eclipse Public License, Version 1.0 <http://www.eclipse.org/legal/epl-v10.html> (lib/aether-impl.license)

Aether SPI (<http://www.eclipse.org/aether/aether-spi/>) org.eclipse.aether:aether-spi:jar:1.0.0.v20140518

Copyright (c) 2010-2014 Sonatype, Inc.

License: Eclipse Public License, Version 1.0 <http://www.eclipse.org/legal/epl-v10.html> (lib/aether-spi.license)

Aether Transport Wagon (<http://www.eclipse.org/aether/aether-transport-wagon/>) org.eclipse.aether:aether-transport-wagon:jar:1.0.0.v20140518

Copyright (c) 2010-2014 Sonatype, Inc.

License: Eclipse Public License, Version 1.0 <http://www.eclipse.org/legal/epl-v10.html> (lib/aether-transport-wagon.license)

Aether Utilities (<http://www.eclipse.org/aether/aether-util/>) org.eclipse.aether:aether-util:jar:1.0.0.v20140518

Copyright (c) 2010-2014 Sonatype, Inc.

License: Eclipse Public License, Version 1.0 <http://www.eclipse.org/legal/epl-v10.html> (lib/aether-util.license)

org.eclipse.sisu.inject (<http://www.eclipse.org/sisu/org.eclipse.sisu.inject/>)

org.eclipse.sisu:org.eclipse.sisu.inject:eclipse-plugin:0.3.0.M1

Copyright (c) 2010, 2015 Sonatype, Inc.

License: Eclipse Public License, Version 1.0 <http://www.eclipse.org/legal/epl-v10.html> (lib/org.eclipse.sisu.inject.license)

org.eclipse.sisu.plexus (<http://www.eclipse.org/sisu/org.eclipse.sisu.plexus/>)

org.eclipse.sisu:org.eclipse.sisu.plexus:eclipse-plugin:0.3.0.M1

Copyright (c) 2010, 2015 Sonatype, Inc.

License: Eclipse Public License, Version 1.0 <http://www.eclipse.org/legal/epl-v10.html> (lib/org.eclipse.sisu.plexus.license)

jsoup (<http://jsoup.org/>) org.jsoup:jsoup:jar:1.7.2

Copyright (c) 2009-2019 Jonathan Hedley <jonathan@hedley.net>

License: The MIT License <http://jsoup.com/license> (lib/jsoup.license)

SLF4J API Module (<http://www.slf4j.org>) org.slf4j:slf4j-api:jar:1.7.5

Copyright (c) 2004-2017 QOS.ch All rights reserved.

License: MIT License <http://www.opensource.org/licenses/mit-license.php> (lib/slf4j-api.license)

SLF4J Simple Binding (<http://www.slf4j.org>) org.slf4j:slf4j-simple:jar:1.7.5

Copyright (c) 2004-2017 QOS.ch All rights reserved.

License: MIT License <http://www.opensource.org/licenses/mit-license.php> (lib/slf4j-simple.license)

Plexus Cipher: encryption/decryption Component (<http://spice.sonatype.org/plexus-cipher>)

org.sonatype.plexus:plexus-cipher:jar:1.7

Copyright (c) 2008 Sonatype, Inc. All rights reserved.

License: Apache Public License 2.0 <http://www.apache.org/licenses/LICENSE-2.0> (lib/plexus-cipher.license)

Plexus Security Dispatcher Component (<http://spice.sonatype.org/plexus-sec-dispatcher>)

org.sonatype.plexus:plexus-sec-dispatcher:jar:1.3

Copyright (c) 2008 Sonatype, Inc. All rights reserved.

License: Apache Public License 2.0 <http://www.apache.org/licenses/LICENSE-2.0> (lib/plexus-sec-dispatcher.license)

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.1

1. Definitions.

1.1. "Contributor" means each individual or entity that creates or contributes to the creation of Modifications.

1.2. "Contributor Version" means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. "Covered Software" means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. "Executable" means the Covered Software in any form other than Source Code.

1.5. "Initial Developer" means the individual or entity that first makes Original Software available under this License.

1.6. "Larger Work" means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. "License" means this document.

1.8. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. "Modifications" means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. "Original Software" means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. "Source Code" means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to

offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Oracle is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the

License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as "Participant") alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the

Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. If You assert a patent infringement claim against Participant alleging that the Participant Software directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

6.4. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" (as that term is defined at 48 C.F.R.

252.227-7014(a)(1)) and "commercial computer software documentation" as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the

state courts of the State of California, with venue lying in Santa Clara County, California.

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:

i) changes to the Program, and

ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents " mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and

such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

- a) it complies with the terms and conditions of this Agreement; and
- b) its license agreement:
 - i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
 - ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
 - iii) states that any provisions which differ from this Agreement are

offered by that Contributor alone and not by any other party; and

iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and

b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other

Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue

and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

The MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License, Version 2.0

=====
ASM

=====
Copyright INRIA, France Telecom
<https://gitlab.ow2.org/asm/asm/blob/master/LICENSE.txt>

ASM: a very small and fast Java bytecode manipulation framework
Copyright (c) 2000-2011 INRIA, France Telecom
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====
JLine

=====
Copyright Marc Prud'hommeaux
Copyright (c) 2002-2016, the original author or authors.
All rights reserved.

<http://www.opensource.org/licenses/bsd-license.php>

Redistribution and use in source and binary forms, with or

without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of JLine nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====

jansi 1.12 Apache 2.0

hawtjni 2.4.1 Apache 2.0

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or,

within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all

other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

=====

Jakarta Restful Web Services

=====

jakarta.ws.rs:jakarta.ws.rs-api

Copyright (c) 2019 Eclipse Foundation.

Copyright (c) 2010,2019 Oracle and/or its affiliates. All rights reserved.

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

- a) in the case of the initial Contributor, the initial content Distributed under this Agreement, and
- b) in the case of each subsequent Contributor:
 - i) changes to the Program, and ii) additions to the Program; where such changes and/or additions to the Program originate from and are Distributed by that particular Contributor. A Contribution "originates" from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include changes or additions to the Program that are not Modified Works.

"Contributor" means any person or entity that Distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions Distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

"Derivative Works" shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

"Modified Works" shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of

the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

"Distribute" means the acts of a) distributing or b) making available in any manner that enables the transfer of a copy.

"Source Code" means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Secondary License" means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

e) Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

3. REQUIREMENTS

3.1 If a Contributor Distributes the Program in any form, then:

a) the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and

b) the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:

i) effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and

iv) requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.

3.2 When the Program is Distributed as Source Code:

a) it must be made available under this Agreement, or if the Program (i) is combined with other material in a separate file or files made available under a Secondary License, and (ii) the initial Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made

available under the terms of such Secondary Licenses, and

b) a copy of this Agreement must be included with each copy of the Program.

3.3 Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability ("notices") contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable.

However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.

Exhibit A - Form of Secondary Licenses Notice

"This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: {name license(s), version(s), and exceptions or additional permissions here}."

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

***** NOTICE FILE: *****

Notices for the Jakarta RESTful Web Services Project

This content is produced and maintained by the Jakarta RESTful Web Services project.

Project home: <https://projects.eclipse.org/projects/ee4j.jaxrs>

Trademarks

Jakarta RESTful Web Services is a trademark of the Eclipse Foundation.

Copyright

All content is the property of the respective authors or their employers. For more information regarding authorship of content, please consult the listed source code repository logs. Declared Project Licenses

This program and the accompanying materials are made available under the terms of the Eclipse Public License v. 2.0 which is available at <http://www.eclipse.org/legal/epl-2.0>. This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License v. 2.0 are satisfied: GNU General Public License, version 2 with the GNU Classpath Exception which is available at <https://www.gnu.org/software/classpath/license.html>.

SPDX-License-Identifier: EPL-2.0 OR GPL-2.0 WITH Classpath-exception-2.0
Source Code

The project maintains the following source code repositories:

<https://github.com/eclipse-ee4j/jaxrs-api>

Third-party Content

This project leverages the following third party content.

--- javaee-api (7.0)

License: Apache-2.0 AND W3C

--- The W3C SOFTWARE NOTICE AND LICENSE (W3C)

View Summary of W3C Software Notice and License (W3C) on TLDRLegal
(Disclaimer) W3C SOFTWARE NOTICE AND LICENSE

Copyright 1994-2001 World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University). All Rights Reserved.
<http://www.w3.org/Consortium/Legal/>

This W3C work (including software, documents, or other related items) is being provided by the copyright holders under the following license. By obtaining, using and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions:

Permission to use, copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications, that you make:

1. The full text of this NOTICE in a location viewable to users of the redistributed or derivative work.
2. Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, a short notice of the following form (hypertext is preferred, text is permitted) should be used within the body of any redistributed or derivative code: "Copyright [\$date-of-software] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University). All Rights Reserved. <http://www.w3.org/Consortium/Legal/>"
3. Notice of any changes or modifications to the W3C files, including the date changes were made. (We recommend you provide URIs to the location from which the code is derived.)

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR DOCUMENTATION.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to the software without specific, written prior permission. Title to copyright in this software and any associated documentation will at all times remain with copyright holders.

```
=====
Netty
=====
                The Netty Project
                =====
```

Please visit the Netty web site for more information:

* <http://netty.io/>

Copyright 2014 The Netty Project
Apache License, Version 2.0

```
=====
jackson-core
=====
Jackson Core
Copyright 20082019 FasterXML. All rights reserved.
```

This copy of Jackson JSON processor streaming parser/generator is licensed under the Apache (Software) License, version 2.0 ("the License"). See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

<http://www.apache.org/licenses/LICENSE-2.0>

NOTICE FILE:

```
=====
# Jackson JSON processor
```

Jackson is a high-performance, Free/Open Source JSON processing library. It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has been in development since 2007. It is currently developed by a community of developers, as well as supported commercially by FasterXML.com.

```
## Licensing
```

Jackson core and extension components may licensed under different licenses.

To find the details that apply to this artifact see the accompanying LICENSE file.
For more information, including possible other licensing options, contact
FasterXML.com (<http://fasterxml.com>).

Credits

A list of contributors may be found from CREDITS file, which is included
in some artifacts (usually source distributions); but is always available
from the source code management (SCM) system project uses.

=====

Apache License, Version 2.0

=====

jackson-databind

=====

Jackson Databind

Copyright (c) 2019 Tatu Saloranta <tatu.saloranta@iki.fi>

Apache License, Version 2.0

=====

SLF4J

=====

Simple Logging Facade for Java (SLF4J) 1.7.26

Copyright (c) 2004-2017 QOS.ch

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=====
Apache Ant
=====

Apache Ant
Copyright 1999-2019 The Apache Software Foundation
Apache License, Version 2.0

The <sync> task is based on code Copyright (c) 2002, Landmark Graphics Corp that has been kindly donated to the Apache Software Foundation.

W3C SOFTWARE NOTICE AND LICENSE

<http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231>

This work (and included software, documentation such as READMEs, or other related items) is being provided by the copyright holders under the following license. By obtaining, using and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions.

Permission to copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications:

1. The full text of this NOTICE in a location viewable to users of the redistributed or derivative work.
2. Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the W3C Software Short Notice should be included (hypertext is preferred, text is permitted) within the body of any redistributed or derivative code.
3. Notice of any changes or modifications to the files, including the date changes were made. (We recommend you provide URIs to the location from which the code is derived.)

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR DOCUMENTATION.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to the software without specific, written prior permission.

Title to copyright in this software and any associated documentation will at all times remain with copyright holders.

This formulation of W3C's notice and license became active on December 31 2002. This version removes the copyright ownership notice such that this license can be used with materials other than those owned by the W3C, reflects that ERCIM is now a host of the W3C, includes references to this specific dated version of the license, and removes the ambiguous grant of "use". Otherwise, this version is the same as the previous version and is written so as to preserve the Free Software Foundation's assessment of GPL compatibility and OSI's certification under the Open Source Definition. Please see our Copyright FAQ for common questions about using materials from our site, including specific terms and conditions for packages like libwww, Amaya, and Jigsaw. Other questions about this notice can be directed to site-policy@w3.org.

Joseph Reagle <site-policy@w3.org>

This license came from: <http://www.megginson.com/SAX/copying.html>
However please note future versions of SAX may be covered
under <http://saxproject.org/?selected=pd>

SAX2 is Free!

I hereby abandon any property rights to SAX 2.0 (the Simple API for XML), and release all of the SAX 2.0 source code, compiled code, and documentation contained in this distribution into the Public Domain. SAX comes with NO WARRANTY or guarantee of fitness for any purpose.

David Megginson, david@megginson.com
2000-05-05

=====
AQuote Bndlib

=====
AQuote Bndlib
Copyright: 2006-2010, OSGi Alliance
Apache License, Version 2.0

=====
Genson

=====
Genson
Copyright 2011-2014 Genson - Cepoi Eugen <cepoi.eugen@gmail.com>
Apache License, Version 2.0

=====
Jaeger Tracing Client

=====

Jaeger Tracing Client
Copyright (c) 2018, The Jaeger Authors
Apache License, Version 2.0

Fourth Party Dependencies

"SLF4J API Module" (org.slf4j:slf4j-api)
Copyright (c) 2004-2011 QOS.ch

The MIT License SPDX short identifier: MIT

Further resources on the MIT License Copyright <YEAR> <COPYRIGHT HOLDER>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

"Apache Thrift" (org.apache.thrift:libthrift)
Apache Thrift
Copyright (C) 2006 - 2019, The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).
Apache License Version 2.0

"okhttp" (com.squareup.okhttp3:okhttp)
Copyright (C) 2018 Square, Inc.
Apache License Version 2.0

"Okio" (com.squareup.okio:okio)
Copyright (C) 2019 Square, Inc.
Apache License Version 2.0

"org.jetbrains.kotlin:kotlin-stdlib" (org.jetbrains.kotlin:kotlin-stdlib)
Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
Apache License Version 2.0

"org.jetbrains.kotlin:kotlin-stdlib-common" (org.jetbrains.kotlin:kotlin-stdlib-common)
Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
Apache License Version 2.0

"OpenTracing API" (io.opentracing:opentracing-api)
Copyright 2016-2019 The OpenTracing Authors
Apache License Version 2.0

"OpenTracing-util" (io.opentracing:opentracing-util)
Copyright 2016-2019 The OpenTracing Authors
Apache License Version 2.0

"OpenTracing-noop" (io.opentracing:opentracing-noop)
Copyright 2016-2019 The OpenTracing Authors
Apache License Version 2.0

"Gson" (com.google.code.gson:gson)
Copyright (C) 2017,2018 The Gson authors
Copyright (C) 2008,2014 Google Inc.
Copyright (C) 2010 The Android Open Source Project
Apache License Version 2.0

"Tracer resolver" (io.opentracing.contrib:opentracing-tracerresolver)
Copyright 2017-2019 The OpenTracing Authors
Apache License Version 2.0

=====
JUnit
=====

JUnit
Copyright JUnit

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

- a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
- b) in the case of each subsequent Contributor:

- i) changes to the Program, and

- ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents " mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

- a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

- b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

- c) Recipient understands that although each Contributor grants the

licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

- a) it complies with the terms and conditions of this Agreement; and
- b) its license agreement:
 - i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
 - ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
 - iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
 - iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

- a) it must be made available under this Agreement; and
- b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the

Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of

data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated

in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

=====

4th party: hamcrest 2.2

BSD License

Copyright (c) 2000-2015 www.hamcrest.org
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of Hamcrest nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY

WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====
spymemcached

=====
spymemcached

couchbase/spymemcached is licensed under the MIT License.

Copyright (c) 2006-2009 Dustin Sallings

Copyright (c) 2009-2011 Couchbase, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=====
Apache Felix Framework

=====
Apache Felix Framework

Copyright 2019 Apache Software Foundation

Apache License, Version 2.0

NOTICE file contents:

Apache Felix Framework

Copyright 2006-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

This product includes software developed at
The OSGi Alliance (<http://www.osgi.org/>).
Copyright (c) OSGi Alliance (2000, 2015).
Licensed under the Apache License 2.0.

Fourth party dependencies

With Apache license:

org.apache.felix:org.apache.felix.resolver 1.14.0
org.osgi:org.osgi.core 5.0.0
org.osgi:org.osgi.annotation 6.0.0

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You

institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to you under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Copyright 2015 The Apache Software Foundation

This software was developed at the Apache Software Foundation (<http://www.apache.org>) and may have dependencies on other Apache software licensed under Apache License 2.0.

I. Included Third-Party Software

This product includes software developed at
The OSGi Alliance (<http://www.osgi.org/>).
Copyright (c) OSGi Alliance (2000, 2014).
Licensed under the Apache License 2.0.

II. Used Third-Party Software

This product uses software developed at
The OSGi Alliance (<http://www.osgi.org/>).
Copyright (c) OSGi Alliance (2000, 2014).
Licensed under the Apache License 2.0.

III. License Summary

- Apache License 2.0

License for Animal Sniffer Annotations:

The MIT License

Copyright (c) 2009 codehaus.org.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) OSGi Alliance (2008, 2018). All Rights Reserved.

=====
Jettison

=====
Jettison
Copyright 2006 Envoi Solutions LLC
Apache License, Version 2.0

=====
Hamcrest
=====
Hamcrest
BSD License

Copyright (c) 2000-2015 www.hamcrest.org
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of
conditions and the following disclaimer. Redistributions in binary form must reproduce
the above copyright notice, this list of conditions and the following disclaimer in
the documentation and/or other materials provided with the distribution.

Neither the name of Hamcrest nor the names of its contributors may be used to endorse
or promote products derived from this software without specific prior written
permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY
EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO
EVENT
SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED
TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN
ANY
WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
DAMAGE.

=====
JaCoCo
=====
JaCoCo
Copyright (c) 2009, 2018 Mountainminds GmbH & Co. KG and Contributors

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

- a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
- b) in the case of each subsequent Contributor:
 - i) changes to the Program, and
 - ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents " mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under

Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

- a) it complies with the terms and conditions of this Agreement; and
- b) its license agreement:
 - i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
 - ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
 - iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
 - iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on

or through a medium customarily used for software exchange.

When the Program is made available in source code form:

- a) it must be made available under this Agreement; and
- b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be

modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

=====

Trademarks

Java and all Java-based trademarks are trademarks of Oracle Corporation in the United States, other countries, or both. Eclipse and all Eclipse related trademarks and logos are trademarks of the Eclipse Foundation, Inc. OSGi is a trademark, registered trademark, or service mark of The OSGi Alliance in the US and other countries. Apache Ant and Apache Maven are trademarks of the Apache Software Foundation. Android and Dalvik are trademarks of Google Inc. All other trademarks are the property of their respective owners.

Third Party Content

The Content includes items that have been sourced from third parties as set out below.

--

Apache Ant 1.7.1

Licensed under the Apache License 2.0.

--

ASM 6.2.1

ASM is subject to the terms and conditions of the following license:

Copyright (c) 2012 France Tlcom

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions

are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

--

args4j 2.0.28

Copyright (c) 2013 Kohsuke Kawaguchi and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

--

Google Code Prettify 2010/07/21

Google Code Prettify is subject to the terms and conditions of the Apache License 2.0.

=====

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the

editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

=====

Mockito

=====

Mockito

The MIT License

Copyright (c) 2007 Mockito contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal

in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=====

FOURTH PARTY LIBRARIES

Byte Buddy (without dependencies) (net.bytebuddy:byte-buddy:1.9.10 - <http://bytebuddy.net/byte-buddy>)
Byte Buddy Java agent (net.bytebuddy:byte-buddy-agent:1.9.10 - <http://bytebuddy.net/byte-buddy-agent>)
Objenesis (org.objenesis:objenesis:2.6 - <http://objenesis.org>)
Apache License, Version 2.0

=====

junit-jupiter-api

=====

junit-jupiter-api

Eclipse Public License - v 2.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE (AGREEMENT). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. Definitions

Contribution means:

- a) in the case of the initial Contributor, the initial content Distributed under this Agreement, and
- b) in the case of each subsequent Contributor:
 - i) changes to the Program, and
 - ii) additions to the Program; where such changes and/or additions to the Program originate from and are Distributed by that particular Contributor. A Contribution originates from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include changes or additions to the Program that are not Modified Works.

Contributor means any person or entity that Distributes the Program.

Licensed Patents mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

Program means the Contributions Distributed in accordance with this Agreement.

Recipient means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

Derivative Works shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

Modified Works shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

Distribute means the acts of a) distributing or b) making available in any manner that enables the transfer of a copy.

Source Code means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

Secondary License means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.

2. Grant of Rights a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations

which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

e) Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

3. Requirements 3.1 If a Contributor Distributes the Program in any form, then:

a) the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and

b) the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:

i) effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose; ii) effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits; iii) does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and iv) requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.

3.2 When the Program is Distributed as Source Code:

a) it must be made available under this Agreement, or if the Program (i) is combined with other material in a separate file or files made available under a Secondary License, and (ii) the initial Contributor attached to the Source Code

the notice described in Exhibit A of this Agreement, then the Program may be made available under the terms of such Secondary Licenses, and b) a copy of this Agreement must be included with each copy of the Program. 3.3 Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability (notices) contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

4. Commercial Distribution Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor (Commercial Contributor) hereby agrees to defend and indemnify every other Contributor (Indemnified Contributor) against any losses, damages and costs (collectively Losses) arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. No Warranty EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN AS IS BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. Disclaimer of Liability EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. General If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights

in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.

Exhibit A - Form of Secondary Licenses Notice This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: {name license(s), version(s), and exceptions or additional permissions here}.

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

END OF TERMS AND CONDITIONS

4P Dependencies:

=====

POM FILE:

org.apiguardian apiguardian-api 1.1.0

LICENSE: Apache 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt>

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the

direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of

this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and

wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor

has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

org.junit.platform junit-platform-commons 1.6.0

LICENSE: Eclipse Public License - v 2.0

<http://www.eclipse.org/legal/epl-v20.html>

Eclipse Public License - v 2.0 THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE (AGREEMENT). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS Contribution means:

a) in the case of the initial Contributor, the initial content Distributed under this Agreement, and b) in the case of each subsequent Contributor: i) changes to the Program, and ii) additions to the Program; where such changes and/or additions to the Program originate from and are Distributed by that particular Contributor. A Contribution originates from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include changes or additions to the Program that are not Modified Works. Contributor means any person or entity that Distributes the Program.

Licensed Patents mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

Program means the Contributions Distributed in accordance with this Agreement.

Recipient means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

Derivative Works shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

Modified Works shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

Distribute means the acts of a) distributing or b) making available in any manner that enables the transfer of a copy.

Source Code means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

Secondary License means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.

2. GRANT OF RIGHTS a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright

license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works. b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder. c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program. d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement. e) Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

3. REQUIREMENTS 3.1 If a Contributor Distributes the Program in any form, then:

a) the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and b) the Contributor may Distribute the Program under a license different than this Agreement, provided that such license: i) effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose; ii) effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits; iii) does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and iv) requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.

3.2 When the Program is Distributed as Source Code:

a) it must be made available under this Agreement, or if the Program (i) is combined with other material in a separate file or files made available under a Secondary License, and (ii) the initial Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available under the terms of such Secondary Licenses, and b) a copy of this Agreement must be included with each copy of the Program. 3.3 Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability (notices) contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

4. COMMERCIAL DISTRIBUTION Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor (Commercial Contributor) hereby agrees to defend and indemnify every other Contributor (Indemnified Contributor) against any losses, damages and costs (collectively Losses) arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN AS IS BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of

rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.

Exhibit A Form of Secondary Licenses Notice This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied:
{name license(s), version(s), and exceptions or additional permissions here}.

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

org.opentest4j opentest4j 1.2.0

LICENSE: Apache 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt>

=====

JUnit

=====

JUnit

COPYRIGHT: Copyright 2015-2020 the original author or authors.

LICENSE: Eclipse Public License - v 2.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE (AGREEMENT). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT. 1. DEFINITIONS

Contribution means:

a) in the case of the initial Contributor, the initial content Distributed under this Agreement, and

b) in the case of each subsequent Contributor:

i) changes to the Program, and

ii) additions to the Program;

where such changes and/or additions to the Program originate from and are Distributed by that particular Contributor. A Contribution originates from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include

changes or additions to the Program that are not Modified Works.

Contributor means any person or entity that Distributes the Program.

Licensed Patents mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

Program means the Contributions Distributed in accordance with this Agreement.

Recipient means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

Derivative Works shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

Modified Works shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

Distribute means the acts of a) distributing or b) making available in any manner that enables the transfer of a copy.

Source Code means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

Secondary License means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the

Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

- c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.
- d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.
- e) Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

3. REQUIREMENTS

3.1 If a Contributor Distributes the Program in any form, then:

- a) the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and
- b) the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:
 - i) effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
 - ii) effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
 - iii) does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and
 - iv) requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.

3.2 When the Program is Distributed as Source Code:

- a) it must be made available under this Agreement, or if the Program (i) is combined with other material in a separate file or files made available under a Secondary License, and (ii) the initial Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available under the terms of such Secondary Licenses, and
- b) a copy of this Agreement must be included with each copy of the Program.

3.3 Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability (notices) contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor (Commercial Contributor) hereby agrees to defend and indemnify every other Contributor (Indemnified Contributor) against any losses, damages and costs (collectively Losses) arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN AS IS BASIS, WITHOUT WARRANTIES

OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable

separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement. Exhibit A Form of Secondary Licenses Notice

This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: {name license(s), version(s), and exceptions or additional permissions here}.

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

4P Dependencies:

=====

POM FILE:

org.apiguardian apiguardian-api 1.1.0

LICENSE: Apache 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt>

COPYRIGHT: Copyright 2002-2019 the original author or authors.

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,

and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the

Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,

whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

org.junit.platform junit-platform-commons 1.6.0

LICENSE: Eclipse Public License - v 2.0

<http://www.eclipse.org/legal/epl-v20.html>

COPYRIGHT: Copyright 2015-2019 the original author or authors.

Eclipse Public License - v 2.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE (AGREEMENT). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS Contribution means:

a) in the case of the initial Contributor, the initial content Distributed under this Agreement, and

b) in the case of each subsequent Contributor: i) changes to the Program, and ii) additions to the Program; where such changes and/or additions to the Program originate from and are Distributed by that particular Contributor. A

Contribution originates from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf.

Contributions do not include changes or additions to the Program that are not Modified Works. Contributor means any person or entity that Distributes the Program.

Licensed Patents mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

Program means the Contributions Distributed in accordance with this Agreement.

Recipient means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

Derivative Works shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

Modified Works shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

Distribute means the acts of a) distributing or b) making available in any manner that enables the transfer of a copy.

Source Code means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

Secondary License means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

e) Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

3. REQUIREMENTS

3.1 If a Contributor Distributes the Program in any form, then:

a) the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and

informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and

b) the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:

i) effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and

iv) requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.

3.2 When the Program is Distributed as Source Code:

a) it must be made available under this Agreement, or if the Program (i) is combined with other material in a separate file or files made available under a Secondary License, and (ii) the initial Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available under the terms of such Secondary Licenses, and

b) a copy of this Agreement must be included with each copy of the Program.

3.3 Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability (notices) contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor (Commercial Contributor) hereby agrees to defend and indemnify every other Contributor (Indemnified Contributor) against any losses, damages and costs (collectively Losses) arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN AS IS BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does

not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.

Exhibit A Form of Secondary Licenses Notice

This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: {name license(s), version(s), and exceptions or additional permissions here}.

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

org.opentest4j opentest4j 1.2.0

LICENSE: Apache 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt>

COPYRIGHT: Copyright 2016-2019 the original author or authors.

org.junit.platform:junit-platform-engine:jar:1.6.0

LICENSE: Eclipse Public License - v 2.0 <http://www.eclipse.org/legal/epl-v20.html>

COPYRIGHT: Copyright 2016-2020 the original author or authors.

END of DEPENDENCIES

=====

mockito-core

=====

mockito-core

The MIT License

Copyright (c) 2007 Mockito contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=====

FOURTH PARTY LIBRARIES

Byte Buddy (without dependencies) (net.bytebuddy:byte-buddy:1.10.5 -

<http://bytebuddy.net/byte-buddy>)

Byte Buddy Java agent (net.bytebuddy:byte-buddy-agent:1.10.5 -

<http://bytebuddy.net/byte-buddy-agent>)

Objenesis (org.objenesis:objenesis:2.6 - <http://objenesis.org>)

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent

to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work,

excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any

risks associated with Your exercise of permissions under this License.

8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

=====
REST-assured
=====

REST-assured

LICENSE: Apache 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt>

COPYRIGHT: Copyright 2010, 2020 the original author or authors.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a

copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct

or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of

this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following

boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

=====

Fourth Party Dependencies

org.codehaus.groovy:groovy:jar
LICENSE: Apache 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt>

COPYRIGHT: Copyright 2007-2020 The Apache Software Foundation

org.codehaus.groovy:groovy-xml:jar
LICENSE: Apache 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt>

COPYRIGHT: Copyright 2012-2020 The Apache Software Foundation

org.apache.httpcomponents:httpclient:jar
LICENSE: Apache 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt>

COPYRIGHT: Copyright 1999-2020 The Apache Software Foundation

org.apache.httpcomponents:httpcore:jar
LICENSE: Apache 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt>

COPYRIGHT: Copyright 2005-2020 The Apache Software Foundation

commons-logging:commons-logging:jar
LICENSE: Apache 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt>

COPYRIGHT: Copyright 2003-2014 The Apache Software Foundation

commons-codec:commons-codec:jar
LICENSE: Apache 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt>

COPYRIGHT: Copyright 2002-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

src/test/org/apache/commons/codec/language/DoubleMetaphoneTest.java
contains test data from <http://aspell.net/test/orig/batch0.tab>.
Copyright (C) 2002 Kevin Atkinson (kevina@gnu.org)

=====
=====

The content of package org.apache.commons.codec.language.bm has
been translated from the original php source code available at
<http://stevemorse.org/phoneticinfo.htm>
with permission from the original authors.
Original source copyright:
Copyright (c) 2008 Alexander Beider & Stephen P. Morse.
License File:
Apache License Version 2.0

org.apache.httpcomponents:httpmime:jar
LICENSE: Apache 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt>

COPYRIGHT: Copyright 1999-2017 The Apache Software Foundation

org.ccil.cowan.tagsoup:tagsoup:jar
LICENSE: Apache 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt>

COPYRIGHT: Copyright 2005-2011 the original author or authors.

io.rest-assured:json-path:jar
LICENSE: Apache 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt>

COPYRIGHT: Copyright 2016-2020 the original author or authors.

org.codehaus.groovy:groovy-json:jar

LICENSE: Apache 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt>

COPYRIGHT: Copyright 2012-2020 The Apache Software Foundation

io.rest-assured:rest-assured-common:jar

LICENSE: Apache 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt>

COPYRIGHT: Copyright 2016-2020 the original author or authors.

io.rest-assured:xml-path:jar

LICENSE: Apache 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt>

COPYRIGHT: Copyright 2016-2020 the original author or authors.

org.apache.commons:commons-lang3:jar

LICENSE: Apache 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt>

COPYRIGHT: Copyright 2011-2019 The Apache Software Foundation

jakarta.xml.bind:jakarta.xml.bind-api:jar

LICENSE: Eclipse Distribution License - v 1.0

COPYRIGHT: Copyright 2019-2020 The original author or authors.

Copyright (c) 2007, Eclipse Foundation, Inc. and its licensors.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer

in the documentation and/or other materials provided with the distribution.

Neither the name of the Eclipse Foundation, Inc. nor the names of its contributors may be used to endorse or promote products

derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT

NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

jakarta.activation:jakarta.activation-api:jar
LICENSE: Eclipse Distribution License - v 1.0

COPYRIGHT: Copyright 2018-2020 The original author or authors.

Copyright (c) 2007, Eclipse Foundation, Inc. and its licensors.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the Eclipse Foundation, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

org.apache.sling:org.apache.sling.javax.activation:jar:0.1.0:test
LICENSE: Apache 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt>

COPYRIGHT: Copyright 2013 The original author or authors.

=====
weld-junit5
=====

weld-junit5

JBoss, Home of Professional Open Source

Copyright 2017, 2019 Red Hat, Inc., and individual contributors

Apache License, Version 2.0

=====
Fourth Party Dependencies

org.jboss.weld:weld-junit-common:jar:2.0.1.Final

LICENSE: Apache 2.0 <http://www.apache.org/licenses/LICENSE-2.0.txt>

COPYRIGHT:

JBoss, Home of Professional Open Source

Copyright 2017, 2019 Red Hat, Inc., and individual contributors

org.hibernate.javax.persistence:hibernate-jpa-2.1-api:jar:1.0.0.Final

LICENSE: Eclipse Distribution License - v 1.0

COPYRIGHT:

Copyright (c) 2007, Eclipse Foundation, Inc. and its licensors.

All rights reserved.

Eclipse Distribution License - v 1.0

Copyright (c) 2007, Eclipse Foundation, Inc. and its licensors.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list
of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this
list of conditions and the following disclaimer in the documentation and/or
other materials provided with the distribution.

Neither the name of the Eclipse Foundation, Inc. nor the names of its
contributors may be used to endorse or promote products derived from this
software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

<http://www.eclipse.org/legal/epl-v10.html>

<http://www.eclipse.org/org/documents/edl-v10.php>

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and

b) in the case of each subsequent Contributor:

i) changes to the Program, and

ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:

i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and

iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and

b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to

control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under

Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

=====
rxjava
=====
rxjava
Apache License, Version 2.0

Below is 4th party dependency.
-----reactive-streams 1.0.2-----
CC0 1.0 Universal

CREATIVE COMMONS CORPORATION IS NOT A LAW FIRM AND DOES NOT PROVIDE LEGAL

SERVICES. DISTRIBUTION OF THIS DOCUMENT DOES NOT CREATE AN ATTORNEY-CLIENT RELATIONSHIP. CREATIVE COMMONS PROVIDES THIS INFORMATION ON AN "AS-IS" BASIS. CREATIVE COMMONS MAKES NO WARRANTIES REGARDING THE USE OF THIS DOCUMENT OR THE INFORMATION OR WORKS PROVIDED HEREUNDER, AND DISCLAIMS LIABILITY FOR DAMAGES RESULTING FROM THE USE OF THIS DOCUMENT OR THE INFORMATION OR WORKS PROVIDED HEREUNDER.

Statement of Purpose

The laws of most jurisdictions throughout the world automatically confer exclusive Copyright and Related Rights (defined below) upon the creator and subsequent owner(s) (each and all, an "owner") of an original work of authorship and/or a database (each, a "Work").

Certain owners wish to permanently relinquish those rights to a Work for the purpose of contributing to a commons of creative, cultural and scientific works ("Commons") that the public can reliably and without fear of later claims of infringement build upon, modify, incorporate in other works, reuse and redistribute as freely as possible in any form whatsoever and for any purposes, including without limitation commercial purposes. These owners may contribute to the Commons to promote the ideal of a free culture and the further production of creative, cultural and scientific works, or to gain reputation or greater distribution for their Work in part through the use and efforts of others.

For these and/or other purposes and motivations, and without any expectation of additional consideration or compensation, the person associating CC0 with a Work (the "Affirmer"), to the extent that he or she is an owner of Copyright and Related Rights in the Work, voluntarily elects to apply CC0 to the Work and publicly distribute the Work under its terms, with knowledge of his or her Copyright and Related Rights in the Work and the meaning and intended legal effect of CC0 on those rights.

1. Copyright and Related Rights. A Work made available under CC0 may be protected by copyright and related or neighboring rights ("Copyright and Related Rights"). Copyright and Related Rights include, but are not limited to, the following:

- the right to reproduce, adapt, distribute, perform, display, communicate, and translate a Work;
- moral rights retained by the original author(s) and/or performer(s);
- publicity and privacy rights pertaining to a person's image or likeness depicted in a Work;
- rights protecting against unfair competition in regards to a Work, subject to the limitations in paragraph 4(a), below;
- rights protecting the extraction, dissemination, use and reuse of data in a Work;
- database rights (such as those arising under Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal

protection of databases, and under any national implementation thereof, including any amended or successor version of such directive); and other similar, equivalent or corresponding rights throughout the world based on applicable law or treaty, and any national implementations thereof.

2. Waiver. To the greatest extent permitted by, but not in contravention of, applicable law, Affirmer hereby overtly, fully, permanently, irrevocably and unconditionally waives, abandons, and surrenders all of Affirmer's Copyright and Related Rights and associated claims and causes of action, whether now known or unknown (including existing as well as future claims and causes of action), in the Work (i) in all territories worldwide, (ii) for the maximum duration provided by applicable law or treaty (including future time extensions), (iii) in any current or future medium and for any number of copies, and (iv) for any purpose whatsoever, including without limitation commercial, advertising or promotional purposes (the "Waiver"). Affirmer makes the Waiver for the benefit of each member of the public at large and to the detriment of Affirmer's heirs and successors, fully intending that such Waiver shall not be subject to revocation, rescission, cancellation, termination, or any other legal or equitable action to disrupt the quiet enjoyment of the Work by the public as contemplated by Affirmer's express Statement of Purpose.

3. Public License Fallback. Should any part of the Waiver for any reason be judged legally invalid or ineffective under applicable law, then the Waiver shall be preserved to the maximum extent permitted taking into account Affirmer's express Statement of Purpose. In addition, to the extent the Waiver is so judged Affirmer hereby grants to each affected person a royalty-free, non transferable, non sublicensable, non exclusive, irrevocable and unconditional license to exercise Affirmer's Copyright and Related Rights in the Work (i) in all territories worldwide, (ii) for the maximum duration provided by applicable law or treaty (including future time extensions), (iii) in any current or future medium and for any number of copies, and (iv) for any purpose whatsoever, including without limitation commercial, advertising or promotional purposes (the "License"). The License shall be deemed effective as of the date CC0 was applied by Affirmer to the Work. Should any part of the License for any reason be judged legally invalid or ineffective under applicable law, such partial invalidity or ineffectiveness shall not invalidate the remainder of the License, and in such case Affirmer hereby affirms that he or she will not (i) exercise any of his or her remaining Copyright and Related Rights in the Work or (ii) assert any associated claims and causes of action with respect to the Work, in either case contrary to Affirmer's express Statement of Purpose.

4. Limitations and Disclaimers.

No trademark or patent rights held by Affirmer are waived, abandoned, surrendered, licensed or otherwise affected by this document. Affirmer offers the Work as-is and makes no representations or warranties of any kind concerning the Work, express, implied, statutory or otherwise, including without limitation warranties of title, merchantability, fitness for a

particular purpose, non infringement, or the absence of latent or other defects, accuracy, or the present or absence of errors, whether or not discoverable, all to the greatest extent permissible under applicable law. Affirmer disclaims responsibility for clearing rights of other persons that may apply to the Work or any use thereof, including without limitation any person's Copyright and Related Rights in the Work. Further, Affirmer disclaims responsibility for obtaining any necessary consents, permissions or other rights required for any use of the Work.

Affirmer understands and acknowledges that Creative Commons is not a party to this document and has no duty or obligation with respect to this CC0 or use of the Work.

=====

Helidon

=====

Helidon

Copyright (c) 2017, 2020 Oracle and/or its affiliates. All rights reserved.

Apache License, Version 2.0

NOTICE:

Project Helidon

=====

Copyright (c) 2017, 2020 Oracle and/or its affiliates. All rights reserved.

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

=====

Third Party Dependencies

=====

This project includes or depends on code from third party projects.

Attributions are contained in THIRD_PARTY_LICENSES.txt.

=====
Hibernate
=====

Hibernate ORM - hibernate-core (org.hibernate:hibernate-core)
Copyright (c) 2008, Red Hat Middleware LLC or third-party contributors
Copyright (c) 2010, Red Hat Middleware LLC or third-party contributors
Copyright (c) 2013, Red Hat Middleware LLC or third-party contributors
Copyright (c) 2015, Red Hat Middleware LLC or third-party contributors

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some
specially designated software packages--typically libraries--of the
Free Software Foundation and other authors who decide to use it. You
can use it too, but we suggest you first think carefully about whether
this license or the ordinary General Public License is the better
strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use,
not price. Our General Public Licenses are designed to make sure that
you have the freedom to distribute copies of free software (and charge
for this service if you wish); that you receive source code or can get
it if you want it; that you can change the software and use pieces of
it in new free programs; and that you are informed that you can do
these things.

To protect your rights, we need to make restrictions that forbid
distributors to deny you these rights or to ask you to surrender these

rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages

are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated

straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of

its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form

under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies

the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

- a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.
- b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and

conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is

safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

=====
Fourth Party Dependencies
=====

"Hibernate Commons Annotations" (org.hibernate.common:hibernate-commons-annotations)
Copyright 2001-2015 Red Hat, Inc. All Rights Reserved.
LGPL 2.1

"dom4j" (org.dom4j:dom4j)
Copyright 2001- 2005 (C) MetaStuff, Ltd. All Rights Reserved.
Copyright 2001-2016 (C) MetaStuff, Ltd. and DOM4J contributors. All Rights Reserved.

Redistribution and use of this software and associated documentation ("Software"), with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain copyright statements and notices. Redistributions must also contain a copy of this document.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name "DOM4J" must not be used to endorse or promote products derived from this Software without prior written permission of MetaStuff, Ltd. For written permission, please contact dom4j-info@metastuff.com.
4. Products derived from this Software may not be called "DOM4J" nor may "DOM4J" appear in their names without prior written permission of MetaStuff, Ltd. DOM4J is a registered trademark of MetaStuff, Ltd.
5. Due credit should be given to the DOM4J Project - <https://dom4j.github.io/>

THIS SOFTWARE IS PROVIDED BY METASTUFF, LTD. AND CONTRIBUTORS "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL METASTUFF, LTD. OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

"Javassist" (org.javassist:javassist)

Copyright (C) 1999- Shigeru Chiba. All Rights Reserved.

Copyright (C) 1999- Shigeru Chiba. All Rights Reserved.</i>]]></bottom>

Copyright (C) 2004 Bill Burke. All Rights Reserved.

Copyright (C) 1999-2019 by Shigeru Chiba, All rights reserved.

Javassist (JAVA programming ASSISTant) makes Java bytecode manipulation simple.

It is a class library for editing bytecodes in Java; it enables Java programs to

define a new class at runtime and to modify a class file when the JVM loads it. Unlike other similar bytecode editors, Javassist provides two levels of API: source level and bytecode level. If the users use the source-level API, they can edit a class file without knowledge of the specifications of the Java bytecode. The whole API is designed with only the vocabulary of the Java language. You can even specify inserted bytecode in the form of source text; Javassist compiles it on the fly. On the other hand, the bytecode-level API allows the users to directly edit a class file as other editors.

This software is distributed under the Mozilla Public License Version 1.1, the GNU Lesser General Public License Version 2.1 or later, or the Apache License Version 2.0.

Oracle elects the Apache License Version 2.0.

Apache License 2.0

antlr:antlr

[The "BSD 3-clause license"]

Copyright (c) 2012-2017 The ANTLR Project. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====

MIT License for codepointat.js from <https://git.io/codepointat>
MIT License for fromcodepoint.js from <https://git.io/vDW1m>

Copyright Mathias Bynens <<https://mathiasbynens.be/>>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=====

com.ibm.icu icu4j 61.1
<https://github.com/unicode-org/icu/tree/release-61-1>
COPYRIGHT: 2016 and later: Unicode, Inc. and others.
LICENSE: MIT

COPYRIGHT AND PERMISSION NOTICE

Copyright 1991-2018 Unicode, Inc. All rights reserved.
Distributed under the Terms of Use in <http://www.unicode.org/copyright.html>.

Permission is hereby granted, free of charge, to any person obtaining a copy of the Unicode data files and any associated documentation (the "Data Files") or Unicode software and any associated documentation (the "Software") to deal in the Data Files or Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Data Files or Software, and to permit persons to whom the Data Files or Software are furnished to do so, provided that either (a) this copyright and permission notice appear with all copies of the Data Files or Software, or

(b) this copyright and permission notice appear in associated Documentation.

THE DATA FILES AND SOFTWARE ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS.

IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THE DATA FILES OR SOFTWARE.

====

org.abego.treelayout.core 1.0.3

COPYRIGHT and LICENSE:

BSD 3-Clause License

Copyright (c) 2011, abego Software GmbH, Germany (<http://www.abego.org>)

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

====

antlr4-runtime

COPYRIGHT:

Copyright (c) 2003-2008 Terence Parr. All rights reserved.

Copyright (c) 2008, Yahoo! Inc. All rights reserved.

Code licensed under the BSD License:

<http://www.antlr.org/license.html>

[The BSD License]

Copyright (c) 2012 Terence Parr and Sam Harwell

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the author nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

====

antlr-runtime 3.5.2

COPYRIGHT:

Copyright (c) 2003-2008 Terence Parr. All rights reserved.

Copyright (c) 2008, Yahoo! Inc. All rights reserved.

Code licensed under the BSD License:

<http://www.antlr.org/license.html>

[The BSD License]

Copyright (c) 2012 Terence Parr and Sam Harwell
All rights reserved.

Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this
list of conditions and the following disclaimer in the documentation and/or
other materials provided with the distribution.

Neither the name of the author nor the names of its contributors may be used
to endorse or promote products derived from this software without specific
prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON
ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

org.antlr ST4 4.3

COPYRIGHT and LICENSE:

[The "BSD license"]

Copyright (c) 2011-2013 Terence Parr

All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES

OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

org.jboss.logging:jboss-logging
Copyright 2010 Red Hat, Inc., and individual contributors
Apache License 2.0

org.hibernate.javafx.persistence:hibernate-jpa-2.1-api
Copyright (c) 2008, 2009, 2011 Oracle, Inc. All rights reserved.

Eclipse Distribution License - v 1.0

Copyright (c) 2007, Eclipse Foundation, Inc. and its licensors.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list
of conditions and the following disclaimer. Redistributions in binary form must
reproduce the above copyright notice, this list of conditions and the following
disclaimer in the documentation and/or other materials provided with the
distribution. Neither the name of the Eclipse Foundation, Inc. nor the names of
its contributors may be used to endorse or promote products derived from this
software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY
THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT
SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE
OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF
ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

org.jboss.spec.javafx.transaction:jboss-transaction-api_1.2_spec

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

1. Definitions.

1.1. "Contributor" means each individual or entity that creates or contributes to the creation of Modifications.

1.2. "Contributor Version" means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. "Covered Software" means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. "Executable" means the Covered Software in any form other than Source Code.

1.5. "Initial Developer" means the individual or entity that first makes Original Software available under this License.

1.6. "Larger Work" means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. "License" means this document.

1.8. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. "Modifications" means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. "Original Software" means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims,

in any patent Licensable by grantor.

1.12. "Source Code" means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform,

sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party. (d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any

Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as "Participant") alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the

Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and "commercial computer software documentation" as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if

any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

org.jboss:jandex
Copyright 2013 Red Hat, Inc., and individual contributors
Apache License 2.0

com.fasterxml:classmate
Copyright: (C) FasterXML, LLC and Tatu Saloranta
Java ClassMate library was originally written by Tatu Saloranta (tatu.saloranta@iki.fi)

Other developers who have contributed code are:

* Brian Langel

Apache License 2.0

=====
HSQL DB License
=====

HSQL DB License

/*

* For work developed by the HSQL Development Group:

*

- * Copyright (c) 2001-2019, The HSQL Development Group
- * All rights reserved.
- *
- * Redistribution and use in source and binary forms, with or without
- * modification, are permitted provided that the following conditions are met:
- *
- * Redistributions of source code must retain the above copyright notice, this
- * list of conditions and the following disclaimer.
- *
- * Redistributions in binary form must reproduce the above copyright notice,
- * this list of conditions and the following disclaimer in the documentation
- * and/or other materials provided with the distribution.
- *
- * Neither the name of the HSQL Development Group nor the names of its
- * contributors may be used to endorse or promote products derived from this
- * software without specific prior written permission.
- *
- * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
- * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
- * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
- * ARE DISCLAIMED. IN NO EVENT SHALL HSQL DEVELOPMENT GROUP, HSQLDB.ORG,
- * OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
- * EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
- * PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
- * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
- * ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
- * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
- * SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
- *
- *
- *
- * For work originally developed by the Hypersonic SQL Group:
- *
- * Copyright (c) 1995-2000, The Hypersonic SQL Group.
- * All rights reserved.
- *
- * Redistribution and use in source and binary forms, with or without
- * modification, are permitted provided that the following conditions are met:
- *
- * Redistributions of source code must retain the above copyright notice, this
- * list of conditions and the following disclaimer.
- *
- * Redistributions in binary form must reproduce the above copyright notice,
- * this list of conditions and the following disclaimer in the documentation
- * and/or other materials provided with the distribution.
- *
- * Neither the name of the Hypersonic SQL Group nor the names of its
- * contributors may be used to endorse or promote products derived from this

* software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE HYPERSONIC SQL GROUP,
* OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
* EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
* PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
* ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
* SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*
* This software consists of voluntary contributions made by many individuals
* on behalf of the Hypersonic SQL Group.
*/

=====
Netty
=====
io.netty:netty-codec-http2
io.netty:netty-handler-proxy
io.netty:netty-all

Apache License, Version 2.0

The Netty Project
=====

Please visit the Netty web site for more information:

* <https://netty.io/>

Copyright 2014 The Netty Project

The Netty Project licenses this file to you under the Apache License, version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at:

<https://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Also, please refer to each LICENSE.<component>.txt file, which is located in the 'license' directory of the distribution file, for the license terms of the components that this product depends on.

This product contains the extensions to Java Collections Framework which has been derived from the works by JSR-166 EG, Doug Lea, and Jason T. Greene:

* LICENSE:

* license/LICENSE.jsr166y.txt (Public Domain)

* HOMEPAGE:

* <http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/>

* <http://viewvc.jboss.org/cgi-bin/viewvc.cgi/jboss/cache/experimental/jsr166/>

license/LICENSE.jsr166y.txt (Public Domain)

The person or persons who have associated work with this document (the "Dedicator" or "Certifier") hereby either (a) certifies that, to the best of his knowledge, the work of authorship identified is in the public domain of the country from which the work is published, or (b) hereby dedicates whatever copyright the dedicator holds in the work of authorship identified below (the "Work") to the public domain. A certifier, moreover, dedicates any copyright interest he may have in the associated work, and for these purposes, is described as a "dedicator" below.

A certifier has taken reasonable steps to verify the copyright status of this work. Certifier recognizes that his good faith efforts may not shield him from liability if in fact the work certified is not in the public domain.

Dedicator makes this dedication for the benefit of the public at large and to the detriment of the Dedicator's heirs and successors. Dedicator intends this dedication to be an overt act of relinquishment in perpetuity of all present and future rights under copyright law, whether vested or contingent, in the Work. Dedicator understands that such relinquishment of all rights includes the relinquishment of all rights to enforce (by lawsuit or otherwise) those copyrights in the Work.

Dedicator recognizes that, once placed in the public domain, the Work may be freely reproduced, distributed, transmitted, used, modified, built upon, or otherwise exploited by anyone for any purpose, commercial or non-commercial, and in any way, including by methods that have not yet been invented or conceived.

This product contains a modified version of Robert Harder's Public Domain Base64 Encoder and Decoder, which can be obtained at:

- * LICENSE:
- * license/LICENSE.base64.txt (Public Domain)
- * HOMEPAGE:
- * <http://iharder.sourceforge.net/current/java/base64/>

license/LICENSE.base64.txt (Public Domain)

The person or persons who have associated work with this document (the "Dedicator" or "Certifier") hereby either (a) certifies that, to the best of his knowledge, the work of authorship identified is in the public domain of the country from which the work is published, or (b) hereby dedicates whatever copyright the dedicators holds in the work of authorship identified below (the "Work") to the public domain. A certifier, moreover, dedicates any copyright interest he may have in the associated work, and for these purposes, is described as a "dedicator" below.

A certifier has taken reasonable steps to verify the copyright status of this work. Certifier recognizes that his good faith efforts may not shield him from liability if in fact the work certified is not in the public domain.

Dedicator makes this dedication for the benefit of the public at large and to the detriment of the Dedicator's heirs and successors. Dedicator intends this dedication to be an overt act of relinquishment in perpetuate of all present and future rights under copyright law, whether vested or contingent, in the Work. Dedicator understands that such relinquishment of all rights includes the relinquishment of all rights to enforce (by lawsuit or otherwise) those copyrights in the Work.

Dedicator recognizes that, once placed in the public domain, the Work may be freely reproduced, distributed, transmitted, used, modified, built upon, or otherwise exploited by anyone for any purpose, commercial or non-commercial, and in any way, including by methods that have not yet been invented or conceived.

This product contains a modified portion of 'Webbit', an event based WebSocket and HTTP server, which can be obtained at:

- * LICENSE:
- * license/LICENSE.webbit.txt (BSD License)
- * HOMEPAGE:
- * <https://github.com/joewalnes/webbit>

license/LICENSE.webbit.txt (BSD License)

(BSD License: <https://www.opensource.org/licenses/bsd-license>)

Copyright (c) 2011, Joe Walnes, Aslak Hellesøy and contributors

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the Webbit nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product contains a modified portion of 'SLF4J', a simple logging facade for Java, which can be obtained at:

- * LICENSE:
- * license/LICENSE.slf4j.txt (MIT License)
- * HOMEPAGE:
- * <https://www.slf4j.org/>

license/LICENSE.slf4j.txt (MIT License)

/*

* Copyright (c) 2004-2007 QOS.ch

* All rights reserved.
*
* Permission is hereby granted, free of charge, to any person obtaining
* a copy of this software and associated documentation files (the
* "Software"), to deal in the Software without restriction, including
* without limitation the rights to use, copy, modify, merge, publish,
* distribute, sublicense, and/or sell copies of the Software, and to
* permit persons to whom the Software is furnished to do so, subject to
* the following conditions:
*
* The above copyright notice and this permission notice shall be
* included in all copies or substantial portions of the Software.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
* NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
* LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
* OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
* WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
*/

This product contains a modified portion of 'Apache Harmony', an open source
Java SE, which can be obtained at:

* NOTICE:
* license/NOTICE.harmony.txt
* LICENSE:
* license/LICENSE.harmony.txt (Apache License 2.0)
* HOMEPAGE:
* <https://archive.apache.org/dist/harmony/>

This product contains a modified portion of 'jzip2', a Java bzip2 compression
and decompression library written by Matthew J. Francis. It can be obtained at:

* LICENSE:
* license/LICENSE.jzip2.txt (MIT License)
* HOMEPAGE:
* <https://code.google.com/p/jzip2/>

license/LICENSE.jzip2.txt (MIT License)

Copyright (c) 2010-2011 Matthew J. Francis and Contributors of the jzip2 Project

Permission is hereby granted, free of charge, to any person obtaining a copy

of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This product contains a modified portion of 'libdivsufsort', a C API library to construct the suffix array and the Burrows-Wheeler transformed string for any input string of a constant-size alphabet written by Yuta Mori. It can be obtained at:

- * LICENSE:
- * license/LICENSE.libdivsufsort.txt (MIT License)
- * HOMEPAGE:
- * <https://github.com/y-256/libdivsufsort>

license/LICENSE.libdivsufsort.txt (MIT License)

Copyright (c) 2003-2008 Yuta Mori All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT

HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This product contains a modified portion of Nitsan Wakart's 'JCTools', Java Concurrency Tools for the JVM, which can be obtained at:

- * LICENSE:
- * license/LICENSE.jctools.txt (ASL2 License)
- * HOMEPAGE:
- * <https://github.com/JCTools/JCTools>

This product optionally depends on 'JZlib', a re-implementation of zlib in pure Java, which can be obtained at:

- * LICENSE:
- * license/LICENSE.jzlib.txt (BSD style License)
- * HOMEPAGE:
- * <http://www.jcraft.com/jzlib/>

license/LICENSE.jzlib.txt (BSD style License)

Copyright (c) 2000,2001,2002,2003,2004 ymnk, JCraft,Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The names of the authors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL JCRAFT, INC. OR ANY CONTRIBUTORS TO THIS SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF

LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product optionally depends on 'Compress-LZF', a Java library for encoding and decoding data in LZF format, written by Tatu Saloranta. It can be obtained at:

- * LICENSE:
- * license/LICENSE.compress-lzf.txt (Apache License 2.0)
- * HOMEPAGE:
- * <https://github.com/ning/compress>

This product optionally depends on 'lz4', a LZ4 Java compression and decompression library written by Adrien Grand. It can be obtained at:

- * LICENSE:
- * license/LICENSE.lz4.txt (Apache License 2.0)
- * HOMEPAGE:
- * <https://github.com/jpountz/lz4-java>

This product optionally depends on 'lzma-java', a LZMA Java compression and decompression library, which can be obtained at:

- * LICENSE:
- * license/LICENSE.lzma-java.txt (Apache License 2.0)
- * HOMEPAGE:
- * <https://github.com/jponge/lzma-java>

This product contains a modified portion of 'jfastlz', a Java port of FastLZ compression and decompression library written by William Kinney. It can be obtained at:

- * LICENSE:
- * license/LICENSE.jfastlz.txt (MIT License)
- * HOMEPAGE:
- * <https://code.google.com/p/jfastlz/>

license/LICENSE.jfastlz.txt (MIT License)

The MIT License

Copyright (c) 2009 William Kinney

Permission is hereby granted, free of charge, to any person

obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This product contains a modified portion of and optionally depends on 'Protocol Buffers', Google's data interchange format, which can be obtained at:

- * LICENSE:
- * [license/LICENSE.protobuf.txt](#) (New BSD License)
- * HOMEPAGE:
- * <https://github.com/google/protobuf>

[license/LICENSE.protobuf.txt](#) (New BSD License)

Protocol Buffers - Google's data interchange format
Copyright 2013 Google Inc. All rights reserved.
<https://developers.google.com/protocol-buffers/>

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from

this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Code generated by the Protocol Buffer compiler is owned by the owner of the input file used when generating it. This code is not standalone and requires a support library to be linked with it. This support library is itself covered by the above license.

This product optionally depends on 'Bouncy Castle Crypto APIs' to generate a temporary self-signed X.509 certificate when the JVM does not provide the equivalent functionality. It can be obtained at:

- * LICENSE:
- * license/LICENSE.bouncycastle.txt (MIT License)
- * HOMEPAGE:
- * <https://www.bouncycastle.org/>

license/LICENSE.bouncycastle.txt (MIT License)

The MIT License (MIT)

Copyright (c) 2000 - 2013 The Legion of the Bouncy Castle Inc.
(<https://www.bouncycastle.org>)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This product optionally depends on 'Snappy', a compression library produced by Google Inc, which can be obtained at:

- * LICENSE:
- * license/LICENSE.snappy.txt (New BSD License)
- * HOMEPAGE:
- * <https://github.com/google/snappy>

license/LICENSE.snappy.txt (New BSD License)

Copyright 2011, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product optionally depends on 'JBoss Marshalling', an alternative Java serialization API, which can be obtained at:

- * LICENSE:
 - * license/LICENSE.jboss-marshalling.txt (Apache License 2.0)
 - * HOMEPAGE:
 - * <https://github.com/jboss-remoting/jboss-marshalling>
-

This product optionally depends on 'Caliper', Google's micro-benchmarking framework, which can be obtained at:

- * LICENSE:
 - * license/LICENSE.caliper.txt (Apache License 2.0)
 - * HOMEPAGE:
 - * <https://github.com/google/caliper>
-

This product optionally depends on 'Apache Commons Logging', a logging framework, which can be obtained at:

- * LICENSE:
 - * license/LICENSE.commons-logging.txt (Apache License 2.0)
 - * HOMEPAGE:
 - * <https://commons.apache.org/logging/>
-

This product optionally depends on 'Apache Log4J', a logging framework, which can be obtained at:

- * LICENSE:
 - * license/LICENSE.log4j.txt (Apache License 2.0)
 - * HOMEPAGE:
 - * <https://logging.apache.org/log4j/>
-

This product optionally depends on 'Aalto XML', an ultra-high performance non-blocking XML processor, which can be obtained at:

- * LICENSE:
 - * license/LICENSE.aalto-xml.txt (Apache License 2.0)
 - * HOMEPAGE:
 - * <http://wiki.fasterxml.com/AaltoHome>
-

This product contains a modified version of 'HPACK', a Java implementation of

the HTTP/2 HPACK algorithm written by Twitter. It can be obtained at:

- * LICENSE:
- * license/LICENSE.hpack.txt (Apache License 2.0)
- * HOMEPAGE:
- * <https://github.com/twitter/hpack>

This product contains a modified version of 'HPACK', a Java implementation of the HTTP/2 HPACK algorithm written by Cory Benfield. It can be obtained at:

- * LICENSE:
- * license/LICENSE.hyper-hpack.txt (MIT License)
- * HOMEPAGE:
- * <https://github.com/python-hyper/hpack/>

license/LICENSE.hyper-hpack.txt (MIT License)

The MIT License (MIT)

Copyright (c) 2014 Cory Benfield

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This product contains a modified version of 'HPACK', a Java implementation of the HTTP/2 HPACK algorithm written by Tatsuhiro Tsujikawa. It can be obtained at:

- * LICENSE:
- * license/LICENSE.nghttp2-hpack.txt (MIT License)
- * HOMEPAGE:

* <https://github.com/nghttp2/nghttp2/>

license/LICENSE.nghttp2-hpack.txt (MIT License)

The MIT License

Copyright (c) 2012, 2014, 2015, 2016 Tatsuhiro Tsujikawa

Copyright (c) 2012, 2014, 2015, 2016 nghttp2 contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This product contains a modified portion of 'Apache Commons Lang', a Java library provides utilities for the java.lang API, which can be obtained at:

- * LICENSE:
- * <license/LICENSE.commons-lang.txt> (Apache License 2.0)
- * HOMEPAGE:
- * <https://commons.apache.org/proper/commons-lang/>

This product contains the Maven wrapper scripts from 'Maven Wrapper', that provides an easy way to ensure a user has everything necessary to run the Maven build.

- * LICENSE:
- * <license/LICENSE.mvn-wrapper.txt> (Apache License 2.0)
- * HOMEPAGE:
- * <https://github.com/takari/maven-wrapper>

This product contains the dnsinfo.h header file, that provides a way to retrieve the system DNS configuration on MacOS.

This private header is also used by Apple's open source mDNSResponder (<https://opensource.apple.com/tarballs/mDNSResponder/>).

* LICENSE:

* license/LICENSE.dnsinfo.txt (Apple Public Source License 2.0)

* HOMEPAGE:

* <https://www.opensource.apple.com/source/configd/configd-453.19/dnsinfo/dnsinfo.h>

license/LICENSE.dnsinfo.txt (Apple Public Source License 2.0)

/*

* Copyright (c) 2004-2006, 2008, 2009, 2011 Apple Inc. All rights reserved.

*

* @APPLE_LICENSE_HEADER_START@

*

* This file contains Original Code and/or Modifications of Original Code

* as defined in and that are subject to the Apple Public Source License

* Version 2.0 (the 'License'). You may not use this file except in

* compliance with the License. Please obtain a copy of the License at

* <https://www.opensource.apple.com/apsl/> and read it before using this

* file.

*

* The Original Code and all software distributed under the License are

* distributed on an 'AS IS' basis, WITHOUT WARRANTY OF ANY KIND, EITHER

* EXPRESS OR IMPLIED, AND APPLE HEREBY DISCLAIMS ALL SUCH WARRANTIES,

* INCLUDING WITHOUT LIMITATION, ANY WARRANTIES OF MERCHANTABILITY,

* FITNESS FOR A PARTICULAR PURPOSE, QUIET ENJOYMENT OR NON-INFRINGEMENT.

* Please see the License for the specific language governing rights and

* limitations under the License.

*

* @APPLE_LICENSE_HEADER_END@

*/

=====

grpc-java

=====

io.grpc:grpc-protobuf

io.grpc:grpc-stub

io.grpc:grpc-netty

Copyright 2014,2020 The gRPC Authors

Copyright 2018, gRPC Authors All rights reserved.

Copyright 2014 The gRPC Authors

Apache License, Version 2.0

Copyright 2014 The gRPC Authors

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

This product contains a modified portion of 'OkHttp', an open source
HTTP & SPDY client for Android and Java applications, which can be obtained
at:

- * LICENSE:
 - * [okhttp/third_party/okhttp/LICENSE](#) (Apache License 2.0)
- * HOMEPAGE:
 - * <https://github.com/square/okhttp>
- * LOCATION_IN_GRPC:
 - * [okhttp/third_party/okhttp](#)

This product contains a modified portion of 'Envoy', an open source
cloud-native high-performance edge/middle/service proxy, which can be
obtained at:

- * LICENSE:
 - * [xds/third_party/envoy/LICENSE](#) (Apache License 2.0)
- * NOTICE:
 - * [xds/third_party/envoy/NOTICE](#)
- * HOMEPAGE:
 - * <https://www.envoyproxy.io>
- * LOCATION_IN_GRPC:
 - * [xds/third_party/envoy](#)

This product contains a modified portion of 'protoc-gen-validate (PGV)',
an open source protoc plugin to generate polyglot message validators,
which can be obtained at:

- * LICENSE:

* xds/third_party/protoc-gen-validate/LICENSE (Apache License 2.0)
* NOTICE:
* xds/third_party/protoc-gen-validate/NOTICE
* HOMEPAGE:
* <https://github.com/envoyproxy/protoc-gen-validate>
* LOCATION_IN_GRPC:
* xds/third_party/protoc-gen-validate

This product contains a modified portion of 'udpa',
an open source universal data plane API, which can be obtained at:

* LICENSE:
* xds/third_party/udpa/LICENSE (Apache License 2.0)
* HOMEPAGE:
* <https://github.com/cncf/udpa>
* LOCATION_IN_GRPC:
* xds/third_party/udpa

Fourth Party Dependencies

"perfmark:perfmark-api" (io.perfmark:perfmark-api)
Copyright 2019,2020 Google LLC
Apache License Version 2.0

"Gson" (com.google.code.gson:gson)
Copyright (C) 2017,2018 The Gson authors
Copyright (C) 2008,2014 Google Inc.
Copyright (C) 2010 The Android Open Source Project
Apache License Version 2.0

"Google Android Annotations Library" (com.google.android:annotations)
Copyright (C) 2012 The Android Open Source Project
Apache License Version 2.0

"error-prone annotations" (com.google.errorprone:error_prone_annotations)
Copyright 2014,2017 The Error Prone Authors.
Apache License Version 2.0

"FindBugs-jsr305" (com.google.code.findbugs:jsr305)
Copyright (c) JSR305 expert group
Apache License Version 2.0

"Guava: Google Libraries for Java" (com.google.guava:*)
Copyright (C) 2005,2020 The Guava Authors
Copyright (C) 2018 The Guava Authors
Apache License Version 2.0

"J2ObjC Annotations" (com.google.j2objc:j2objc-annotations)

Copyright 2012 Google Inc. All Rights Reserved.

Apache License Version 2.0

"Netty" (io.netty:netty-*)

Copyright 2011,2020 The Netty Project

Copyright 2014,2019 The Netty Project

Copyright 2014 Twitter, Inc.

Copyright (c) 2004-2011 QOS.ch

Copyright 2012,2017 The Netty Project

Copyright (c) 2011, Joe Walnes and contributors

Copyright 2012,2019 The Netty Project

Copyright (c) 2008-2009 Bjoern Hoehrmann <bjoern@hoehrmann.de>

Apache License Version 2.0

"proto-google-common-protos" (com.google.api.grpc:proto-google-common-protos)

Copyright 2020 Google LLC

Apache License Version 2.0

"Checker Qual" (org.checkerframework:checker-compat-qual)

Checker Framework qualifiers

Copyright 2004-present by the Checker Framework developers

MIT License:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

"Animal Sniffer Annotations" (org.codehaus.mojo:animal-sniffer-annotations)

Copyright (c) 2009 codehaus.org.

Copyright (c) 2008 Kohsuke Kawaguchi and codehaus.org.

MIT License:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

"Protocol Buffers [Core]" (com.google.protobuf:protobuf-java)
Copyright 2008 Google Inc. All rights reserved.

The 3-Clause BSD License SPDX short identifier: BSD-3-Clause

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====
r2dbc-h2

=====
io.r2dbc:r2dbc-h2

Copyright 2019 the original author or authors.

Apache License, Version 2.0

NOTICE:

=====
Reactive Relational Database Connectivity

Copyright 2017-2019 the original author or authors.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<https://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Fourth Party Dependencies

R2DBC SPI (io.r2dbc:r2dbc-spi:jar)

Copyright 2017-2019 the original author or authors.

Apache License, Version 2.0

SLF4J API (org.slf4j:slf4j-api:jar)

Copyright (c) 2004-2017 QOS.ch

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,

distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

H2 Database (com.h2database:h2:jar)
Copyright 2004-2021 H2 Group.

H2 is dual licensed and available under the MPL 2.0 (Mozilla Public License Version 2.0) or under the EPL 1.0 (Eclipse Public License).

Mozilla Public License, version 2.0

1. Definitions

1.1. Contributor

means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

1.2. Contributor Version

means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution.

1.3. Contribution

means Covered Software of a particular Contributor.

1.4. Covered Software

means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

1.5. Incompatible With Secondary Licenses

means

a. that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or

b. that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

1.6. Executable Form

means any form of the work other than Source Code Form.

1.7. Larger Work

means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

1.8. License

means this document.

1.9. Licensable

means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

1.10. Modifications

means any of the following:

a. any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or

b. any new file in Source Code Form that contains any Covered Software.

1.11. Patent Claims of a Contributor

means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

1.12. Secondary License

means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

1.13. Source Code Form

means the form of the work preferred for making modifications.

1.14. You (or Your)

means an individual or a legal entity exercising rights under this License. For legal entities, You includes any entity that controls,

is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

- a. under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and
- b. under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

- a. for any code that a Contributor has removed from Covered Software; or
- b. for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or
- c. under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

- a. such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and
- b. You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice,

provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an

ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

6. Disclaimer of Warranty

Covered Software is provided under this License on an as is basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

7. Limitation of Liability

Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may

not apply to You.

8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - Incompatible With Secondary Licenses Notice

This Source Code Form is Incompatible With Secondary Licenses, as defined by the Mozilla Public License, v. 2.0.

Eclipse Public License, Version 1.0 (EPL-1.0)

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

- a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
- b) in the case of each subsequent Contributor:
 - i) changes to the Program, and
 - ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents " mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or

when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

- a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.
- b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution.
No hardware per se is licensed hereunder.
- c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.
- d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

- a) it complies with the terms and conditions of this Agreement; and
- b) its license agreement:
 - i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
 - ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
 - iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
 - iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

- a) it must be made available under this Agreement; and
- b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the

Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under

applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

Reactor Core (io.projectreactor:reactor-core:jar)

/*

* Copyright (c) 2011-2017 Pivotal Software Inc, All Rights Reserved.

*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <https://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Reactor Addons (io.projectreactor.addons:reactor-extra:jar)

/*
* Copyright (c) 2011-2017 Pivotal Software Inc, All Rights Reserved.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <https://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Reactive Streams (org.reactivestreams:reactive-streams:jar)

/*
* Licensed under Public Domain (CC0) *
*
* To the extent possible under law, the person who associated CC0 with *
* this code has waived all copyright and related or neighboring *
* rights to this code. *
*
* You should have received a copy of the CC0 legalcode along with this *
* work. If not, see <<http://creativecommons.org/publicdomain/zero/1.0/>>.*
*/

CREATIVE COMMONS CORPORATION IS NOT A LAW FIRM AND DOES NOT
PROVIDE LEGAL SERVICES. DISTRIBUTION OF THIS DOCUMENT DOES NOT

CREATE AN ATTORNEY-CLIENT RELATIONSHIP. CREATIVE COMMONS PROVIDES THIS INFORMATION ON AN "AS-IS" BASIS. CREATIVE COMMONS MAKES NO WARRANTIES REGARDING THE USE OF THIS DOCUMENT OR THE INFORMATION OR WORKS PROVIDED HEREUNDER, AND DISCLAIMS LIABILITY FOR DAMAGES RESULTING FROM THE USE OF THIS DOCUMENT OR THE INFORMATION OR WORKS PROVIDED HEREUNDER.

Statement of Purpose

The laws of most jurisdictions throughout the world automatically confer exclusive Copyright and Related Rights (defined below) upon the creator and subsequent owner(s) (each and all, an "owner") of an original work of authorship and/or a database (each, a "Work").

Certain owners wish to permanently relinquish those rights to a Work for the purpose of contributing to a commons of creative, cultural and scientific works ("Commons") that the public can reliably and without fear of later claims of infringement build upon, modify, incorporate in other works, reuse and redistribute as freely as possible in any form whatsoever and for any purposes, including without limitation commercial purposes. These owners may contribute to the Commons to promote the ideal of a free culture and the further production of creative, cultural and scientific works, or to gain reputation or greater distribution for their Work in part through the use and efforts of others.

For these and/or other purposes and motivations, and without any expectation of additional consideration or compensation, the person associating CC0 with a Work (the "Affirmer"), to the extent that he or she is an owner of Copyright and Related Rights in the Work, voluntarily elects to apply CC0 to the Work and publicly distribute the Work under its terms, with knowledge of his or her Copyright and Related Rights in the Work and the meaning and intended legal effect of CC0 on those rights.

1. Copyright and Related Rights. A Work made available under CC0 may be protected by copyright and related or neighboring rights ("Copyright and Related Rights"). Copyright and Related Rights include, but are not limited to, the following:

the right to reproduce, adapt, distribute, perform, display, communicate, and translate a Work; moral rights retained by the original author(s) and/or performer(s); publicity and privacy rights pertaining to a person's image or likeness depicted in a Work; rights protecting against unfair competition in regards to a Work, subject to the limitations in paragraph 4(a), below; rights protecting the extraction, dissemination, use and reuse of data in a Work; database rights (such as those arising under

Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, and under any national implementation thereof, including any amended or successor version of such directive); and other similar, equivalent or corresponding rights throughout the world based on applicable law or treaty, and any national implementations thereof.

2. Waiver. To the greatest extent permitted by, but not in contravention of, applicable law, Affirmer hereby overtly, fully, permanently, irrevocably and unconditionally waives, abandons, and surrenders all of Affirmer's Copyright and Related Rights and associated claims and causes of action, whether now known or unknown (including existing as well as future claims and causes of action), in the Work (i) in all territories worldwide, (ii) for the maximum duration provided by applicable law or treaty (including future time extensions), (iii) in any current or future medium and for any number of copies, and (iv) for any purpose whatsoever, including without limitation commercial, advertising or promotional purposes (the "Waiver"). Affirmer makes the Waiver for the benefit of each member of the public at large and to the detriment of Affirmer's heirs and successors, fully intending that such Waiver shall not be subject to revocation, rescission, cancellation, termination, or any other legal or equitable action to disrupt the quiet enjoyment of the Work by the public as contemplated by Affirmer's express Statement of Purpose.

3. Public License Fallback. Should any part of the Waiver for any reason be judged legally invalid or ineffective under applicable law, then the Waiver shall be preserved to the maximum extent permitted taking into account Affirmer's express Statement of Purpose. In addition, to the extent the Waiver is so judged Affirmer hereby grants to each affected person a royalty-free, non transferable, non sublicensable, non exclusive, irrevocable and unconditional license to exercise Affirmer's Copyright and Related Rights in the Work (i) in all territories worldwide, (ii) for the maximum duration provided by applicable law or treaty (including future time extensions), (iii) in any current or future medium and for any number of copies, and (iv) for any purpose whatsoever, including without limitation commercial, advertising or promotional purposes (the "License"). The License shall be deemed effective as of the date CC0 was applied by Affirmer to the Work. Should any part of the License for any reason be judged legally invalid or ineffective under applicable law, such partial invalidity or ineffectiveness shall not invalidate the remainder of the License, and in such case Affirmer hereby affirms that he or she will not (i) exercise any of his or her remaining Copyright and Related Rights in the Work or (ii) assert any associated claims and causes of action with

respect to the Work, in either case contrary to Affirmer's express Statement of Purpose.

4. Limitations and Disclaimers.

No trademark or patent rights held by Affirmer are waived, abandoned, surrendered, licensed or otherwise affected by this document. Affirmer offers the Work as-is and makes no representations or warranties of any kind concerning the Work, express, implied, statutory or otherwise, including without limitation warranties of title, merchantability, fitness for a particular purpose, non infringement, or the absence of latent or other defects, accuracy, or the present or absence of errors, whether or not discoverable, all to the greatest extent permissible under applicable law. Affirmer disclaims responsibility for clearing rights of other persons that may apply to the Work or any use thereof, including without limitation any person's Copyright and Related Rights in the Work. Further, Affirmer disclaims responsibility for obtaining any necessary consents, permissions or other rights required for any use of the Work. Affirmer understands and acknowledges that Creative Commons is not a party to this document and has no duty or obligation with respect to this CC0 or use of the Work.

commons-lang3 (org.apache.commons:commons-lang3:jar)
Apache Commons Lang
Copyright 2001-2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<https://www.apache.org/>).

Apache License, Version 2.0

=====
Caffeine
=====

com.github.ben-manes.caffeine:caffeine:jar

/*

* Copyright 2015 Ben Manes. All Rights Reserved.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

- *
 - * Unless required by applicable law or agreed to in writing, software
 - * distributed under the License is distributed on an "AS IS" BASIS,
 - * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 - * See the License for the specific language governing permissions and
 - * limitations under the License.
- */

Apache License, Version 2.0

4th Party Dependencies

+++++

checker-qual, v3.21.3, MIT
 Checker Framework qualifiers
 Copyright 2004-present by the Checker Framework developers

MIT License:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

+++++

errorprone, v2.11.0, Apache 2.0
 /*
 * Copyright 2014 The Error Prone Authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * <http://www.apache.org/licenses/LICENSE-2.0>

*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

+++++

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.

*
* Copyright YYYY Sun Microsystems, Inc. All rights reserved.

*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can obtain
* a copy of the License at <https://glassfish.dev.java.net/public/CDDL+GPL.html>
* or [glassfish/bootstrap/legal/LICENSE.txt](https://glassfish.org/legal/LICENSE.txt). See the License for the specific
* language governing permissions and limitations under the License.

*
* When distributing the software, include this License Header Notice in each
* file and include the License file at [glassfish/bootstrap/legal/LICENSE.txt](https://glassfish.org/legal/LICENSE.txt).
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyrighted [year] [name of copyright owner]"

*
* Contributor(s):

*
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made subject to such option by the copyright
* holder.

*/

%file%

Copyright (c) 2000, 2022, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademarks of Oracle Corporation and/or its
affiliates.

```

#
# This software is the confidential and proprietary information of Oracle
# Corporation. You shall not disclose such confidential and proprietary
# information and shall use it only in accordance with the terms of the
# license agreement you entered into with Oracle.
#
# This notice may not be removed or altered.
#
/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright YYYY Sun Microsystems, Inc. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can obtain
* a copy of the License at https://glassfish.dev.java.net/public/CDDL+GPL.html
* or glassfish/bootstrap/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at glassfish/bootstrap/legal/LICENSE.txt.
* Sun designates this particular file as subject to the "Classpath" exception
* as provided by Sun in the GPL Version 2 section of the License file that
* accompanied this code. If applicable, add the following below the License
* Header, with the fields enclosed by brackets [] replaced by your own
* identifying information: "Portions Copyrighted [year]
* [name of copyright owner]"
*
* Contributor(s):
*
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made subject to such option by the copyright
* holder.
*/
<!--
| %file%
|
| Copyright (c) 2000, 2022, Oracle and/or its affiliates. All rights reserved.
|

```

| Oracle is a registered trademarks of Oracle Corporation and/or its
| affiliates.

|

| This software is the confidential and proprietary information of Oracle
| Corporation. You shall not disclose such confidential and proprietary
| information and shall use it only in accordance with the terms of the
| license agreement you entered into with Oracle.

|

| This notice may not be removed or altered.

-->

/*

* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.

*

* Copyright (c) YYYY Oracle and/or its affiliates. All rights reserved.

*

* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development

* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can

* obtain a copy of the License at

* <https://oss.oracle.com/licenses/CDDL+GPL-1.1>

* or LICENSE.txt. See the License for the specific

* language governing permissions and limitations under the License.

*

* When distributing the software, include this License Header Notice in each
* file and include the License file at LICENSE.txt.

*

* GPL Classpath Exception:

* Oracle designates this particular file as subject to the "Classpath"

* exception as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.

*

* Modifications:

* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:

* "Portions Copyright [year] [name of copyright owner]"

*

* Contributor(s):

* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a

* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made subject to such option by the copyright

* holder.

*/
/*
* Copyright (c) YYYY Oracle and/or its affiliates. All rights reserved.
*
* This program and the accompanying materials are made available under the
* terms of the Eclipse Distribution License v. 1.0, which is available at
* <http://www.eclipse.org/org/documents/edl-v10.php>.
*
* SPDX-License-Identifier: BSD-3-Clause
*/
/*
* Copyright (c) YYYY Oracle and/or its affiliates. All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* - Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
*
* - Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
*
* - Neither the name of Oracle nor the names of its
* contributors may be used to endorse or promote products derived
* from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
* IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
* THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
* CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
* EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
* PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
* PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
* LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
* NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
* SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to

communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of

the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

1.56 feign-slf4j 10.12

1.56.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/**
```

```
* Copyright 2012-2020 The Feign Authors
```

```
*
```

```
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
```

```
* in compliance with the License. You may obtain a copy of the License at
```

```
*
```

```
* http://www.apache.org/licenses/LICENSE-2.0
```

```
*
```

```
* Unless required by applicable law or agreed to in writing, software distributed under the License
```

```
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either  
express
```

```
* or implied. See the License for the specific language governing permissions and limitations under
```

```
* the License.
```

```
*/
```

Found in path(s):

```
* /opt/cola/permits/1418559528_1663298228.924368/0/feign-slf4j-10-12-sources-1-jar/feign/slf4j/Slf4jLogger.java
```

No license file was found, but licenses were detected in source scan.

```
<!--
```

```
Copyright 2012-2020 The Feign Authors
```

```
Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
```

```
in compliance with the License. You may obtain a copy of the License at
```

```
http://www.apache.org/licenses/LICENSE-2.0
```

```
Unless required by applicable law or agreed to in writing, software distributed under the License
```

```
is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either  
express
```

```
or implied. See the License for the specific language governing permissions and limitations under
```

```
the License.
```

```
-->
```

Found in path(s):

```
* /opt/cola/permits/1418559528_1663298228.924368/0/feign-slf4j-10-12-sources-1-jar/META-
```

```
INF/maven/io.github.openfeign/feign-slf4j/pom.xml
```

1.57 t-digest 3.2

1.57.1 Available under license :

```
/*
 * Licensed to Ted Dunning under one or more
 * contributor license agreements. See the NOTICE file distributed with
 * this work for additional information regarding copyright ownership.
 * The ASF licenses this file to You under the Apache License, Version 2.0
 * (the "License"); you may not use this file except in compliance with
 * the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

1.58 junit-platform-junit-platform-engine 1.9.1

1.58.1 Available under license :

Eclipse Public License - v 2.0

=====

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE (AGREEMENT). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. Definitions

Contribution means:

- * **a)** in the case of the initial Contributor, the initial content Distributed under this Agreement, and
- * **b)** in the case of each subsequent Contributor:
 - * **i)** changes to the Program, and
 - * **ii)** additions to the Program;

where such changes and/or additions to the Program originate from and are Distributed by that particular Contributor. A Contribution originates from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include changes or additions to the Program that are not Modified Works.

Contributor means any person or entity that Distributes the Program.

Licensed Patents mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale

of its Contribution alone or when combined with the Program.

Program means the Contributions Distributed in accordance with this Agreement.

Recipient means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

Derivative Works shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

Modified Works shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

Distribute means the acts of ****a)**** distributing or ****b)**** making available in any manner that enables the transfer of a copy.

Source Code means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

Secondary License means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.

2. Grant of Rights

****a)**** Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

****b)**** Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

****c)**** Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

****d)**** Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any,

to grant the copyright license set forth in this Agreement.

****e)**** Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

3. Requirements

****3.1)**** If a Contributor Distributes the Program in any form, then:

****a)**** the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and

****b)**** the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:

****i)**** effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

****ii)**** effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

****iii)**** does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and

****iv)**** requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.

****3.2)**** When the Program is Distributed as Source Code:

****a)**** it must be made available under this Agreement, or if the Program ****i)**** is combined with other material in a separate file or files made available under a Secondary License, and ****ii)**** the initial Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available under the terms of such Secondary Licenses, and

****b)**** a copy of this Agreement must be included with each copy of the Program.

****3.3)**** Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability (notices) contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

4. Commercial Distribution

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor (Commercial Contributor) hereby agrees to defend and indemnify every other Contributor (Indemnified Contributor) against any losses, damages and costs (collectively Losses) arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial

product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: ****a)**** promptly notify the Commercial Contributor in writing of such claim, and ****b)**** allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. No Warranty

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN AS IS BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. Disclaimer of Liability

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. General

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of

such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.

Exhibit A - Form of Secondary Licenses Notice

> This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: {name license(s), version(s), and exceptions or additional permissions here }.

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Open Source Licenses

=====

This product may include a number of subcomponents with separate copyright notices and license terms. Your use of the source code for these subcomponents is subject to the terms and conditions of the subcomponent's license, as noted in the LICENSE-<subcomponent>.md files.

1.59 joda-time 2.10.5

1.59.1 Available under license :

=====

= NOTICE file corresponding to section 4d of the Apache License Version 2.0 =

=====

This product includes software developed by
Joda.org (<https://www.joda.org/>).

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the

editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the

same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.60 unboundid-scim2-sdk-common 2.3.6

1.60.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2016-2020 Ping Identity Corporation
 *
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License (GPLv2 only)
 * or the terms of the GNU Lesser General Public License (LGPLv2.1 only)
 * as published by the Free Software Foundation.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, see <http://www.gnu.org/licenses>.
 */
```

Found in path(s):

```
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/utils/MapperFactory.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/utils/JsonDiff.java
```

No license file was found, but licenses were detected in source scan.

```
/*
```

* Copyright 2019-2020 Ping Identity Corporation
*
* This program is free software; you can redistribute it and/or modify
* it under the terms of the GNU General Public License (GPLv2 only)
* or the terms of the GNU Lesser General Public License (LGPLv2.1 only)
* as published by the Free Software Foundation.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, see <<http://www.gnu.org/licenses>>.
*/

Found in path(s):

* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-jar/com/unboundid/scim2/common/utils/DateSerializer.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-jar/com/unboundid/scim2/common/types/GroupResource.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-jar/com/unboundid/scim2/common/utils/DateTimeUtils.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-jar/com/unboundid/scim2/common/utils/DateDeserializer.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-jar/com/unboundid/scim2/common/utils/CalendarDeserializer.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-jar/com/unboundid/scim2/common/utils/CalendarSerializer.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-jar/com/unboundid/scim2/common/types/Member.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2015-2020 Ping Identity Corporation
*
* This program is free software; you can redistribute it and/or modify
* it under the terms of the GNU General Public License (GPLv2 only)
* or the terms of the GNU Lesser General Public License (LGPLv2.1 only)
* as published by the Free Software Foundation.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, see <<http://www.gnu.org/licenses>>.

*/

Found in path(s):

- * /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-jar/com/unboundid/scim2/common/utils/GenericScimObjectSerializer.java
- * /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-jar/com/unboundid/scim2/common/types/SchemaResource.java
- * /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-jar/com/unboundid/scim2/common/filters/package-info.java
- * /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-jar/com/unboundid/scim2/common/exceptions/ScimException.java
- * /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-jar/com/unboundid/scim2/common/types/BulkConfig.java
- * /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-jar/com/unboundid/scim2/common/types/ChangePasswordConfig.java
- * /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-jar/com/unboundid/scim2/common/exceptions/ForbiddenException.java
- * /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-jar/com/unboundid/scim2/common/messages/SortOrder.java
- * /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-jar/com/unboundid/scim2/common/filters/OrFilter.java
- * /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-jar/com/unboundid/scim2/common/messages/ErrorResponse.java
- * /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-jar/com/unboundid/scim2/common/filters/LessThanFilter.java
- * /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-jar/com/unboundid/scim2/common/utils/StaticUtils.java
- * /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-jar/com/unboundid/scim2/common/utils/ScimFilterJsonParser.java
- * /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-jar/com/unboundid/scim2/common/filters/NotEqualFilter.java
- * /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-jar/com/unboundid/scim2/common/types/ETagConfig.java
- * /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-jar/com/unboundid/scim2/common/messages/PatchOperation.java
- * /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-jar/com/unboundid/scim2/common/utils/JsonUtils.java
- * /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-jar/com/unboundid/scim2/common/utils/JsonRefBeanSerializer.java
- * /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-jar/com/unboundid/scim2/common/utils/ScimJsonFactory.java
- * /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-jar/com/unboundid/scim2/common/BaseScimResource.java
- * /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-jar/com/unboundid/scim2/common/filters/NotFilter.java
- * /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-jar/com/unboundid/scim2/common/utils/SchemaUtils.java
- * /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-

jar/com/unboundid/scim2/common/exceptions/NotModifiedException.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/exceptions/NotImplementedException.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/filters/GreaterThanFilter.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/types/X509Certificate.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/exceptions/UnauthorizedException.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/types/JsonReference.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/types/Meta.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/exceptions/ResourceNotFoundException.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/types/Role.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/utils/DebugType.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/types/PhoneNumber.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/filters/CombiningFilter.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/utils/GenericScimObjectDeserializer.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/exceptions/package-info.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/types/InstantMessagingAddress.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/filters/GreaterThanOrEqualToFilter.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/types/Manager.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/types/Photo.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/filters/LessThanOrEqualToFilter.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/types/EnterpriseUserExtension.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/GenericScimResource.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/package-info.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/types/FilterConfig.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/types/UserResource.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-

jar/com/unboundid/scim2/common/filters/StartsWithFilter.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/utils/StatusDeserializer.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/utils/CaseIgnoreMap.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/utils/ScimDateFormat.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/utils/StatusSerializer.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/exceptions/ServerErrorException.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/filters/Filter.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/exceptions/PreconditionFailedException.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/messages/package-info.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/types/package-info.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/utils/Parser.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/messages/PatchRequest.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/types/Name.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/types/ResourceTypeResource.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/messages/PatchOpType.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/filters/FilterVisitor.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/annotations/Schema.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/Path.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/ScimResource.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/exceptions/ResourceConflictException.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/types/Entitlement.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/types/Group.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/annotations/Attribute.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/filters/PresentFilter.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-

jar/com/unboundid/scim2/common/filters/FilterType.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/messages/ListResponse.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/exceptions/BadRequestException.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/utils/ApiConstants.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/annotations/package-info.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/filters/EqualFilter.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/utils/ScimJsonNodeFactory.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/filters/AndFilter.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/types/SortConfig.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/types/AuthenticationScheme.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/utils/Debug.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/messages/SearchRequest.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/types/AttributeDefinition.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/filters/ContainsFilter.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/filters/ComplexValueFilter.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/types/PatchConfig.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/utils/FilterEvaluator.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/utils/CaseIgnoreObjectNode.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/utils/package-info.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/types/Address.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/types/Email.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/types/ServiceProviderConfigResource.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/filters/ComparisonFilter.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-
jar/com/unboundid/scim2/common/exceptions/MethodNotAllowedException.java
* /opt/cola/permits/1340031433_1654671331.583195/0/scim2-sdk-common-2-3-6-sources-

1.61 jul-to-slf4j-bridge 1.7.32

1.61.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the

editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the

same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 1999-2005 The Apache Software Foundation

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Copyright (c) 2004-2007 QOS.ch
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2004-2013 QOS.ch
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate

as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify

the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

This product includes software developed by
The Apache Software Foundation (<http://www.apache.org/>).
Copyright (c) 2004-2017 QOS.ch
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining

a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.62 feign-core 10.12

1.62.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/**
 * Copyright 2012-2020 The Feign Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-
jar/feign/RetryableException.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/codec/Encoder.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/Headers.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-
jar/feign/template/BodyTemplate.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/RequestInterceptor.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/Request.java
```

* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/RequestTemplate.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/codec/EncodingException.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/stream/StreamDecoder.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/AsyncInvocation.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/Body.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/AsyncFeign.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/Response.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/template/QueryTemplate.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/FeignException.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/Logger.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/auth/BasicAuthRequestInterceptor.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/Client.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/InvocationHandlerFactory.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/codec/ErrorDecoder.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/template/Template.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/SynchronousMethodHandler.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/MethodInfo.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/querymap/BeanQueryMapEncoder.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/optionals/OptionalDecoder.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/QueryMapEncoder.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/Target.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/ReflectiveFeign.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/Capability.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/CollectionFormat.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/codec/Decoder.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/template/Expression.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/AsyncJoinException.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/template/TemplateChunk.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/QueryMap.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/ExceptionPropagationPolicy.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/ReflectiveAsyncFeign.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/codec/DecodeException.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/Contract.java

* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/Feign.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/ResponseMapper.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/template/UriUtils.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/DefaultMethodHandler.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/Types.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/DeclarativeContract.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/HeaderMap.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/Param.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/Util.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/MethodMetadata.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/RequestLine.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/Retryer.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/AsyncClient.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/template/HeaderTemplate.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/template/UriTemplate.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/codecs/StringDecoder.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/template/Literal.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/auth/Base64.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/Experimental.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/AsyncResponseHandler.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/querymap/FieldQueryMapEncoder.java
* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/feign/template/Expressions.java

No license file was found, but licenses were detected in source scan.

<!--

Copyright 2012-2020 The Feign Authors

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

-->

Found in path(s):

* /opt/cola/permits/1418559444_1663298203.93103/0/feign-core-10-12-sources-1-jar/META-INF/maven/io.github.openfeign/feign-core/pom.xml

1.63 micronaut-azure 3.5.0

1.63.1 Available under license :

Apache License

Version 2.0, January 2004

<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed

with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate

comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.64 apache-commons-collections 3.2.2

1.64.1 Available under license :

Apache Commons Collections
Copyright 2001-2015 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (<http://www.apache.org/>).

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or

otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents

of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.65 log4j-compatibility-api 2.17.1

1.65.1 Available under license :

Apache Log4j 1.x Compatibility API
Copyright 1999-1969 The Apache Software Foundation

This product includes software developed at

The Apache Software Foundation (<http://www.apache.org/>).

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or

Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.66 metrics-integration-for-apache- httpasyncclient 4.0.5

1.66.1 Available under license :

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0

Bnd-LastModified: 1545937994519

Build-Jdk: 1.8.0_191

Built-By: artem

Bundle-Description: An Apache HttpAsyncClient wrapper providing Metrics instrumentation of connection pools, request durations and rates, and other useful information.

Bundle-License: <http://www.apache.org/licenses/LICENSE-2.0.html>

Bundle-ManifestVersion: 2

Bundle-Name: Metrics Integration for Apache HttpAsyncClient

Bundle-SymbolicName: io.dropwizard.metrics.httpasyncclient

Bundle-Version: 4.0.5

Created-By: Apache Maven Bundle Plugin

Export-Package: com.codahale.metrics.httpasyncclient;uses:="com.codahale.metrics,com.codahale.metrics.httpclient,org.apache.http.config,org.apache.http.conn,org.apache.http.impl.nio.client,org.apache.http.impl.nio.conn,org.apache.http.nio.conn,org.apache.http.nio.reactor";version="4.0.5"

Implementation-Title: Metrics Integration for Apache HttpAsyncClient

Implementation-URL: <http://metrics.dropwizard.io/metrics-httpasyncclient>

Implementation-Vendor-Id: io.dropwizard.metrics

Implementation-Version: 4.0.5

Import-Package: com.codahale.metrics;version="[4.0,5)",com.codahale.metrics.httpclient;version="[4.0,5)",org.apache.http,org.apache.http.concurrent,org.apache.http.config,org.apache.http.conn,org.apache.http.impl.nio.client,org.apache.http.impl.nio.conn,org.apache.http.nio.conn,org.apache.http.nio.protocol,org.apache.http.nio.reactor,org.apache

.http.pool.org.apache.http.protocol
Require-Capability: osgi.ee;filter:="(&(osgi.ee=JavaSE)(version=1.8))"
Tool: Bnd-3.3.0.201609221906

Found in path(s):

* /opt/cola/permits/1340031649_1654689479.4970224/0/metrics-httplibasyncclient-4-0-5-jar/META-INF/MANIFEST.MF

1.67 zookeeper 3.8.0

1.67.1 Available under license :

No license file was found, but licenses were detected in source scan.

<!--

Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to You under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

-->

Found in path(s):

* /opt/cola/permits/1425206646_1663952692.6053333/0/camel-zookeeper-3-8-0-sources-2-jar/META-INF/maven/org.apache.camel/camel-zookeeper/pom.xml

No license file was found, but licenses were detected in source scan.

/*

* Licensed to the Apache Software Foundation (ASF) under one or more
* contributor license agreements. See the NOTICE file distributed with
* this work for additional information regarding copyright ownership.
* The ASF licenses this file to You under the Apache License, Version 2.0
* (the "License"); you may not use this file except in compliance with
* the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

- * Unless required by applicable law or agreed to in writing, software
- * distributed under the License is distributed on an "AS IS" BASIS,
- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.
- */

Found in path(s):

- * /opt/cola/permits/1425206646_1663952692.6053333/0/camel-zookeeper-3-8-0-sources-2-jar/org/apache/camel/component/zookeeper/ZooKeeperMessage.java
- * /opt/cola/permits/1425206646_1663952692.6053333/0/camel-zookeeper-3-8-0-sources-2-jar/org/apache/camel/component/zookeeper/NaturalSortComparator.java
- * /opt/cola/permits/1425206646_1663952692.6053333/0/camel-zookeeper-3-8-0-sources-2-jar/org/apache/camel/component/zookeeper/ZooKeeperUtils.java
- * /opt/cola/permits/1425206646_1663952692.6053333/0/camel-zookeeper-3-8-0-sources-2-jar/org/apache/camel/component/zookeeper/ZooKeeperEndpoint.java
- * /opt/cola/permits/1425206646_1663952692.6053333/0/camel-zookeeper-3-8-0-sources-2-jar/org/apache/camel/component/zookeeper/cloud/ZooKeeperServiceDiscoveryFactory.java
- * /opt/cola/permits/1425206646_1663952692.6053333/0/camel-zookeeper-3-8-0-sources-2-jar/org/apache/camel/component/zookeeper/ZooKeeperCuratorHelper.java
- * /opt/cola/permits/1425206646_1663952692.6053333/0/camel-zookeeper-3-8-0-sources-2-jar/org/apache/camel/component/zookeeper/ZooKeeperConnectionManager.java
- * /opt/cola/permits/1425206646_1663952692.6053333/0/camel-zookeeper-3-8-0-sources-2-jar/org/apache/camel/component/zookeeper/ZooKeeperCuratorConfiguration.java
- * /opt/cola/permits/1425206646_1663952692.6053333/0/camel-zookeeper-3-8-0-sources-2-jar/org/apache/camel/component/zookeeper/cluster/ZooKeeperClusterView.java
- * /opt/cola/permits/1425206646_1663952692.6053333/0/camel-zookeeper-3-8-0-sources-2-jar/org/apache/camel/component/zookeeper/operations/AnyOfOperations.java
- * /opt/cola/permits/1425206646_1663952692.6053333/0/camel-zookeeper-3-8-0-sources-2-jar/org/apache/camel/component/zookeeper/operations/CreateOperation.java
- * /opt/cola/permits/1425206646_1663952692.6053333/0/camel-zookeeper-3-8-0-sources-2-jar/org/apache/camel/component/zookeeper/ZooKeeperConsumer.java
- * /opt/cola/permits/1425206646_1663952692.6053333/0/camel-zookeeper-3-8-0-sources-2-jar/org/apache/camel/component/zookeeper/cloud/ZooKeeperServiceRegistryConfiguration.java
- * /opt/cola/permits/1425206646_1663952692.6053333/0/camel-zookeeper-3-8-0-sources-2-jar/org/apache/camel/component/zookeeper/operations/GetDataOperation.java
- * /opt/cola/permits/1425206646_1663952692.6053333/0/camel-zookeeper-3-8-0-sources-2-jar/org/apache/camel/component/zookeeper/cloud/ZooKeeperServiceDiscovery.java
- * /opt/cola/permits/1425206646_1663952692.6053333/0/camel-zookeeper-3-8-0-sources-2-jar/org/apache/camel/component/zookeeper/operations/DataChangedOperation.java
- * /opt/cola/permits/1425206646_1663952692.6053333/0/camel-zookeeper-3-8-0-sources-2-jar/org/apache/camel/component/zookeeper/operations/SetDataOperation.java
- * /opt/cola/permits/1425206646_1663952692.6053333/0/camel-zookeeper-3-8-0-sources-2-jar/org/apache/camel/component/zookeeper/operations/DeleteOperation.java
- * /opt/cola/permits/1425206646_1663952692.6053333/0/camel-zookeeper-3-8-0-sources-2-jar/org/apache/camel/component/zookeeper/operations/ChildrenChangedOperation.java
- * /opt/cola/permits/1425206646_1663952692.6053333/0/camel-zookeeper-3-8-0-sources-2-jar/org/apache/camel/component/zookeeper/cloud/ZooKeeperServiceRegistry.java

* /opt/cola/permits/1425206646_1663952692.6053333/0/camel-zookeeper-3-8-0-sources-2-jar/org/apache/camel/component/zookeeper/operations/OperationResult.java
* /opt/cola/permits/1425206646_1663952692.6053333/0/camel-zookeeper-3-8-0-sources-2-jar/org/apache/camel/component/zookeeper/operations/ZooKeeperOperation.java
* /opt/cola/permits/1425206646_1663952692.6053333/0/camel-zookeeper-3-8-0-sources-2-jar/org/apache/camel/component/zookeeper/operations/ExistenceChangedOperation.java
* /opt/cola/permits/1425206646_1663952692.6053333/0/camel-zookeeper-3-8-0-sources-2-jar/org/apache/camel/component/zookeeper/ZooKeeperProducer.java
* /opt/cola/permits/1425206646_1663952692.6053333/0/camel-zookeeper-3-8-0-sources-2-jar/org/apache/camel/component/zookeeper/operations/ZooKeeperHelper.java
* /opt/cola/permits/1425206646_1663952692.6053333/0/camel-zookeeper-3-8-0-sources-2-jar/org/apache/camel/component/zookeeper/SequenceComparator.java
* /opt/cola/permits/1425206646_1663952692.6053333/0/camel-zookeeper-3-8-0-sources-2-jar/org/apache/camel/component/zookeeper/ZooKeeperConfiguration.java
* /opt/cola/permits/1425206646_1663952692.6053333/0/camel-zookeeper-3-8-0-sources-2-jar/org/apache/camel/component/zookeeper/cluster/ZooKeeperClusterService.java
* /opt/cola/permits/1425206646_1663952692.6053333/0/camel-zookeeper-3-8-0-sources-2-jar/org/apache/camel/component/zookeeper/operations/FutureEventDrivenOperation.java
* /opt/cola/permits/1425206646_1663952692.6053333/0/camel-zookeeper-3-8-0-sources-2-jar/org/apache/camel/component/zookeeper/ZooKeeperComponent.java
* /opt/cola/permits/1425206646_1663952692.6053333/0/camel-zookeeper-3-8-0-sources-2-jar/org/apache/camel/component/zookeeper/operations/ExistsOperation.java
* /opt/cola/permits/1425206646_1663952692.6053333/0/camel-zookeeper-3-8-0-sources-2-jar/org/apache/camel/component/zookeeper/operations/GetChildrenOperation.java
* /opt/cola/permits/1425206646_1663952692.6053333/0/camel-zookeeper-3-8-0-sources-2-jar/org/apache/camel/component/zookeeper/ConnectionHolder.java
* /opt/cola/permits/1425206646_1663952692.6053333/0/camel-zookeeper-3-8-0-sources-2-jar/org/apache/camel/component/zookeeper/operations/WatchedEventProvider.java

1.68 metrics-integration-with-jmx 4.0.5

1.68.1 Available under license :

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0

Bnd-LastModified: 1545938260836

Build-Jdk: 1.8.0_191

Built-By: artem

Bundle-Description: A set of classes which allow you to report metrics via JMX.

Bundle-License: <http://www.apache.org/licenses/LICENSE-2.0.html>

Bundle-ManifestVersion: 2

Bundle-Name: Metrics Integration with JMX

Bundle-SymbolicName: io.dropwizard.metrics.jmx

Bundle-Version: 4.0.5

Created-By: Apache Maven Bundle Plugin

Export-Package: com.codahale.metrics.jmx;uses:="com.codahale.metrics,j

avax.management";version="4.0.5"
Implementation-Title: Metrics Integration with JMX
Implementation-URL: http://metrics.dropwizard.io/metrics-jmx
Implementation-Vendor-Id: io.dropwizard.metrics
Implementation-Version: 4.0.5
Import-Package: org.slf4j;version="[1.6.0,2.0.0)",com.codahale.metrics
;version="[4.0,5)",javax.management
Require-Capability: osgi.ee;filter="(&(osgi.ee=JavaSE)(version=1.8))"
Tool: Bnd-3.3.0.201609221906

Found in path(s):

* /opt/cola/permits/1274704779_1648835825.49/0/metrics-jmx-4-0-5-jar/META-INF/MANIFEST.MF

1.69 jose4j 0.6.3

1.69.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*  
* Licensed to the Apache Software Foundation (ASF) under one or more  
* contributor license agreements. See the NOTICE file distributed with  
* this work for additional information regarding copyright ownership.  
* The ASF licenses this file to You under the Apache License, Version 2.0  
* (the "License"); you may not use this file except in compliance with  
* the License. You may obtain a copy of the License at  
*  
* http://www.apache.org/licenses/LICENSE-2.0  
*  
* Unless required by applicable law or agreed to in writing, software  
* distributed under the License is distributed on an "AS IS" BASIS,  
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
* See the License for the specific language governing permissions and  
* limitations under the License.  
*/
```

Found in path(s):

* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-
jar/org/jose4j/base64url/internal/apache/commons/codec/binary/Base64.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-
jar/org/jose4j/base64url/internal/apache/commons/codec/binary/BaseNCodec.java
No license file was found, but licenses were detected in source scan.

```
/*  
* Copyright 2012-2017 Brian Campbell  
*  
* Licensed under the Apache License, Version 2.0 (the "License");  
* you may not use this file except in compliance with the License.
```

* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/lang/InvalidKeyException.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwk/SimpleJwkFilter.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwk/SelectorSupport.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwk/DecryptionJwkSelector.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2012-2017 Brian Campbell
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwk/OctJwkGenerator.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwt/consumer/JwsCustomizer.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwt/consumer/JwtConsumerBuilder.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwk/Use.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwe/kdf/PasswordBasedKeyDerivationFunction2.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwt/IntDate.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwt/consumer/ErrorCodes.java

- * /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwt/consumer/ErrorCodeValidator.java
- * /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jws/JsonWebSignatureAlgorithm.java
- * /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwe/ContentEncryptionKeys.java
- * /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/lang/ByteUtil.java
- * /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwe/AesGcmContentEncryptionAlgorithm.java
- * /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jca/ProviderContext.java
- * /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jws/JsonWebSignature.java
- * /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/keys/KeyPersuasion.java
- * /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/keys/resolvers/JwksVerificationKeyResolver.java
- * /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwe/WrappingKeyManagementAlgorithm.java
- * /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwt/consumer/InvalidJwtException.java
- * /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwt/consumer/JweCustomizer.java
- * /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwe/EcdhKeyAgreementWithAesKeyWrapAlgorithm.java
- * /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwe/ContentEncryptionParts.java
- * /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwe/SimpleAeadCipher.java
- * /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/base64url/Base64Url.java
- * /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwt/consumer/JwtConsumer.java
- * /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwe/ContentEncryptionAlgorithmIdentifiers.java
- * /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/keys/BigEndianBigInteger.java
- * /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwe/AesCbcHmacSha2ContentEncryptionAlgorithm.java
- * /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/keys/resolvers/X509VerificationKeyResolver.java
- * /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/lang/UnresolvableKeyException.java
- * /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwk/PublicJsonWebKey.java
- * /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/http/SimpleGet.java
- * /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwt/consumer/IssValidator.java
- * /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwk/VerificationJwkSelector.java
- * /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/keys/X509Util.java
- * /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwt/JwtClaims.java
- * /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/lang/InvalidAlgorithmException.java

* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/zip/CompressionAlgorithm.java

* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/keys/resolvers/DecryptionKeyResolver.java

* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwx/Headers.java

* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwe/JsonWebEncryption.java

* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwa/AlgorithmInfo.java

* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwe/kdf/ConcatKeyDerivationFunction.java

* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/lang/JsonHelp.java

* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/keys/KeyPairUtil.java

* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jws/AlgorithmIdentifiers.java

* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jws/HmacUsingShaAlgorithm.java

* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwk/OctetSequenceJsonWebKey.java

* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/lang/ByteGenerator.java

* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jws/PlaintextNoneAlgorithm.java

* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwe/EcdhKeyAgreementAlgorithm.java

* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwk/JsonWebKeySet.java

* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwe/DirectKeyManagementAlgorithm.java

* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwe/CipherStrengthSupport.java

* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwt/consumer/SubValidator.java

* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/lang/ExceptionHelp.java

* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwk/RsaJsonWebKey.java

* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwt/consumer/AudValidator.java

* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/keys/EllipticCurves.java

* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwe/KeyManagementAlgorithmIdentifiers.java

* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwk/HttpsJwks.java

* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/lang/JsonWebException.java

* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/lang/UncheckedJsonWebException.java

* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwt/GeneralJwtException.java

* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwx/CompactSerialization.java

* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwt/ReservedClaimNames.java

* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwt/consumer/SimpleKeyResolver.java

* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/mac/MacUtil.java

* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwe/CipherUtil.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/zip/CompressionAlgorithmIdentifiers.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/lang/HashUtil.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwx/JsonWebStructure.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwt/consumer/JwtContext.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/keys/RsaKeyUtil.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwe/Pbes2HmacShaWithAesKeyWrapAlgorithm.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/keys/PbkdfKey.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwt/consumer/ErrorCodeValidatorAdapter.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwt/NumericDate.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwt/consumer/JtiValidator.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/keys/resolvers/JwksDecryptionKeyResolver.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jws/RsaUsingShaAlgorithm.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwe/kdf/KdfUtil.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/lang/BouncyCastleProviderHelp.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwt/MalformedClaimException.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/lang/IntegrityException.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwa/Algorithm.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwx/HeaderParameterNames.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/http/Get.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwt/consumer/Validator.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/keys/resolvers/HttpsJwksVerificationKeyResolver.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwe/ContentEncryptionAlgorithm.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/keys/HmacKey.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwx/CompactSerializer.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwt/consumer/NumericDateValidator.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/json/JsonUtil.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwx/KeyValidationSupport.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwe/ContentEncryptionKeyDescriptor.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/keys/AesKey.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jws/EcdsaUsingShaAlgorithm.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-

jar/org/jose4j/jwa/AlgorithmAvailability.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/zip/DeflateRFC1951CompressionAlgorithm.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/lang/StringUtil.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwe/ContentEncryptionHelp.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/keys/EcKeyUtil.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwk/JsonWebKey.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwe/RsaKeyManagementAlgorithm.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwe/KeyManagementAlgorithm.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwa/AlgorithmFactory.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwa/AlgorithmConstraints.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/lang/DefaultByteGenerator.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwk/EcJwkGenerator.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwk/KeyOperations.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwk/EllipticCurveJsonWebKey.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwe/InitializationVectorHelp.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwe/AesKeyWrapManagementAlgorithm.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jws/BaseSignatureAlgorithm.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/base64url/SimplePEMEncoder.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/keys/resolvers/VerificationKeyResolver.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwa/AlgorithmFactoryFactory.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwe/AesGcmKeyEncryptionAlgorithm.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/jwk/RsaJwkGenerator.java
* /opt/cola/permits/1274698934_1645233552.09/0/jose4j-0-6-3-sources-jar/org/jose4j/json/JsonHeaderUtil.java

1.70 ktor-bom 1.6.8

1.70.1 Available under license :

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution.

You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions.

Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks.

This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty.

Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability.

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability.

While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.71 jetty-java-based-http-1-x-http-2-servlet-websocket-server 11.0.12

1.71.1 Available under license :

Notices for Eclipse Jetty

=====

This content is produced and maintained by the Eclipse Jetty project.

Project home: <https://www.eclipse.org/jetty/>

Trademarks

Eclipse Jetty, and Jetty are trademarks of the Eclipse Foundation.

Copyright

All contributions are the property of the respective authors or of entities to which copyright has been assigned by the authors (eg. employer).

Declared Project Licenses

This artifacts of this project are made available under the terms of:

- * the Eclipse Public License v2.0
<https://www.eclipse.org/legal/epl-2.0>
SPDX-License-Identifier: EPL-2.0

or

- * the Apache License, Version 2.0
<https://www.apache.org/licenses/LICENSE-2.0>
SPDX-License-Identifier: Apache-2.0

The following dependencies are EPL.

- * org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following dependencies are EPL and ASL2.

- * org.eclipse.jetty.orbit:javax.security.auth.message

The following dependencies are EPL and CDDL 1.0.

- * org.eclipse.jetty.orbit:javax.mail.glassfish

The following dependencies are CDDL + GPLv2 with classpath exception.

<https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html>

- * jakarta.servlet:jakarta.servlet-api
- * javax.annotation:javax.annotation-api
- * javax.transaction:javax.transaction-api
- * javax.websocket:javax.websocket-api

The following dependencies are licensed by the OW2 Foundation according to the terms of <http://asm.ow2.org/license.html>

- * org.ow2.asm:asm-commons
- * org.ow2.asm:asm

The following dependencies are ASL2 licensed.

- * org.apache.taglibs:taglibs-standard-spec
- * org.apache.taglibs:taglibs-standard-impl

The following dependencies are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

- * org.mortbay.jasper:apache-jsp
- * org.apache.tomcat:tomcat-jasper
- * org.apache.tomcat:tomcat-juli
- * org.apache.tomcat:tomcat-jsp-api
- * org.apache.tomcat:tomcat-el-api
- * org.apache.tomcat:tomcat-jasper-el
- * org.apache.tomcat:tomcat-api
- * org.apache.tomcat:tomcat-util-scan
- * org.apache.tomcat:tomcat-util
- * org.mortbay.jasper:apache-el
- * org.apache.tomcat:tomcat-jasper-el
- * org.apache.tomcat:tomcat-el-api

The following artifacts are CDDL + GPLv2 with classpath exception.

<https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html>

- * org.eclipse.jetty.toolchain:jetty-schemas

Cryptography

Content may contain encryption software. The country in which you are currently may have restrictions on the import, possession, and use, and/or re-export to another country, of encryption software. BEFORE using any encryption software, please check the country's laws, regulations and policies concerning the import, possession, or use, and re-export of encryption software, to see if this is permitted.

The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville.

Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies.

Eclipse Public License - v 2.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

- a) in the case of the initial Contributor, the initial content Distributed under this Agreement, and
- b) in the case of each subsequent Contributor:
 - i) changes to the Program, and

ii) additions to the Program;
where such changes and/or additions to the Program originate from and are Distributed by that particular Contributor. A Contribution "originates" from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include changes or additions to the Program that are not Modified Works.

"Contributor" means any person or entity that Distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions Distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

"Derivative Works" shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

"Modified Works" shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

"Distribute" means the acts of a) distributing or b) making available in any manner that enables the transfer of a copy.

"Source Code" means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Secondary License" means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

e) Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

3. REQUIREMENTS

3.1 If a Contributor Distributes the Program in any form, then:

a) the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany

the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and

b) the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:

i) effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and

iv) requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.

3.2 When the Program is Distributed as Source Code:

a) it must be made available under this Agreement, or if the Program (i) is combined with other material in a separate file or files made available under a Secondary License, and (ii) the initial Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available under the terms of such Secondary Licenses, and

b) a copy of this Agreement must be included with each copy of the Program.

3.3 Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability ("notices") contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product

offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,

EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any

Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.

Exhibit A - Form of Secondary Licenses Notice

"This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: {name license(s), version(s), and exceptions or additional permissions here}."

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed

as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this

License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

SPDX-License-Identifier: EPL-2.0 OR Apache-2.0

1.72 fabric8-:::-kubernetes-model-:::- scheduling 4.13.3

1.72.1 Available under license :

No license file was found, but licenses were detected in source scan.

Copyright (C) 2015 Red Hat, Inc.

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE>

2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Found in path(s):

* /opt/cola/permits/1288519787_1647861855.61/0/kubernetes-model-scheduling-4-13-3-jar/META-INF/maven/io.fabric8/kubernetes-model-scheduling/pom.xml

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0

Bnd-LastModified: 1619068650619

Build-Jdk-Spec: 1.8

Bundle-Description: Java client for Kubernetes and OpenShift

Bundle-DocURL: <http://redhat.com>

Bundle-License: <http://www.apache.org/licenses/LICENSE-2.0.txt>

Bundle-ManifestVersion: 2

Bundle-Name: Fabric8 :: Kubernetes Model :: Scheduling

Bundle-SymbolicName: io.fabric8/kubernetes-model-scheduling

Bundle-Vendor: Red Hat

Bundle-Version: 4.13.3

Created-By: Apache Maven Bundle Plugin

Export-Package: io.fabric8/kubernetes.api.model.scheduling;uses:="com.fasterxml.jackson.annotation,com.fasterxml.jackson.databind,com.fasterxml.jackson.databind.annotation,io.fabric8/kubernetes.api.builder,io.fabric8/kubernetes.api.model,io.fabric8/kubernetes.model.annotation";version="4.13.3"

Implementation-Title: Fabric8 :: Kubernetes Model :: Scheduling

Implementation-Vendor: Red Hat

Implementation-Version: 4.13.3

Import-Package: com.fasterxml.jackson.annotation;version="[2.11,3)",com.fasterxml.jackson.databind;version="[2.11,3)",com.fasterxml.jackson.databind.annotation;version="[2.11,3)",io.fabric8/kubernetes.api.builder;version="[4.13,5)",io.fabric8/kubernetes.api.model,io.fabric8/kubernetes.model.annotation;version="[4.13,5)"

Require-Capability: osgi.ee;filter="(&(osgi.ee=JavaSE)(version=1.8))"

Specification-Title: Fabric8 :: Kubernetes Model :: Scheduling

Specification-Vendor: Red Hat

Specification-Version: 4.13

Tool: Bnd-5.1.1.202006162103

Found in path(s):

* /opt/cola/permits/1288519787_1647861855.61/0/kubernetes-model-scheduling-4-13-3-jar/META-INF/MANIFEST.MF

No license file was found, but licenses were detected in source scan.

*

* Copyright (C) 2015 Red Hat, Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*

Found in path(s):

* /opt/cola/permits/1288519787_1647861855.61/0/kubernetes-model-scheduling-4-13-3-jar/manifest.vm

1.73 feign-jackson 10.12

1.73.1 Available under license :

No license file was found, but licenses were detected in source scan.

/**

* Copyright 2012-2020 The Feign Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except

* in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License

* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express

* or implied. See the License for the specific language governing permissions and limitations under

* the License.

*/

Found in path(s):

* /opt/cola/permits/1418559492_1663298225.1235242/0/feign-jackson-10-12-sources-1-

jar/feign/jackson/JacksonDecoder.java

```
* /opt/cola/permits/1418559492_1663298225.1235242/0/feign-jackson-10-12-sources-1-  
jar/feign/jackson/JacksonIteratorDecoder.java  
* /opt/cola/permits/1418559492_1663298225.1235242/0/feign-jackson-10-12-sources-1-  
jar/feign/jackson/JacksonEncoder.java  
No license file was found, but licenses were detected in source scan.
```

<!--

Copyright 2012-2020 The Feign Authors

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

-->

Found in path(s):

```
* /opt/cola/permits/1418559492_1663298225.1235242/0/feign-jackson-10-12-sources-1-jar/META-  
INF/maven/io.github.openfeign/feign-jackson/pom.xml
```

1.74 prometheus-java-span-context-supplier-common 0.15.0

1.74.1 Available under license :

No license file was found, but licenses were detected in source scan.

<url><http://www.apache.org/licenses/LICENSE-2.0.txt></url>

Found in path(s):

```
* /opt/cola/permits/1341561687_1654803029.1439095/0/simpleclient-tracer-common-0-15-0-sources-jar/META-  
INF/maven/io.prometheus/simpleclient_tracer_common/pom.xml
```

1.75 feign-okhttp 10.12

1.75.1 Available under license :

No license file was found, but licenses were detected in source scan.

/**

```
* Copyright 2012-2020 The Feign Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
```

Found in path(s):

```
* /opt/cola/permits/1418559504_1663298240.5238116/0/feign-okhttp-10-12-sources-1-
jar/feign-okhttp/OkHttpClient.java
```

No license file was found, but licenses were detected in source scan.

```
<!--
```

Copyright 2012-2020 The Feign Authors

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

```
-->
```

Found in path(s):

```
* /opt/cola/permits/1418559504_1663298240.5238116/0/feign-okhttp-10-12-sources-1-jar/META-
INF/maven/io.github.openfeign/feign-okhttp/pom.xml
```

1.76 jakarta-expression-language-3.0-api 5.0.0

1.76.1 Available under license :

Notices for Jakarta Expression Language

This content is produced and maintained by the Jakarta Expression Language project.

* Project home: <https://projects.eclipse.org/projects/ee4j.el>

Trademarks

Jakarta Expression Language is a trademark of the Eclipse Foundation.

Copyright

All content is the property of the respective authors or their employers. For more information regarding authorship of content, please consult the listed source code repository logs.

Declared Project Licenses

This program and the accompanying materials are made available under the terms of the Eclipse Public License v. 2.0 which is available at <http://www.eclipse.org/legal/epl-2.0>. This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License v. 2.0 are satisfied: GNU General Public License, version 2 with the GNU Classpath Exception which is available at <https://www.gnu.org/software/classpath/license.html>.

SPDX-License-Identifier: EPL-2.0 OR GPL-2.0 WITH Classpath-exception-2.0

Source Code

The project maintains the following source code repositories:

* <https://github.com/eclipse-ee4j/el-ri>

Third-party Content

Cryptography

Content may contain encryption software. The country in which you are currently may have restrictions on the import, possession, and use, and/or re-export to another country, of encryption software. BEFORE using any encryption software, please check the country's laws, regulations and policies concerning the import, possession, or use, and re-export of encryption software, to see if this is permitted.

Eclipse Public License - v 2.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial content Distributed under this Agreement, and

b) in the case of each subsequent Contributor:

i) changes to the Program, and

ii) additions to the Program;

where such changes and/or additions to the Program originate from and are Distributed by that particular Contributor. A Contribution "originates" from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include changes or additions to the Program that are not Modified Works.

"Contributor" means any person or entity that Distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions Distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

"Derivative Works" shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

"Modified Works" shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

"Distribute" means the acts of a) distributing or b) making available in any manner that enables the transfer of a copy.

"Source Code" means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Secondary License" means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

e) Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License

(if permitted under the terms of Section 3).

3. REQUIREMENTS

3.1 If a Contributor Distributes the Program in any form, then:

a) the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and

b) the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:

i) effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and

iv) requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.

3.2 When the Program is Distributed as Source Code:

a) it must be made available under this Agreement, or if the Program (i) is combined with other material in a separate file or files made available under a Secondary License, and (ii) the initial Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available under the terms of such Secondary Licenses, and

b) a copy of this Agreement must be included with each copy of the Program.

3.3 Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability ("notices") contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add

their own appropriate notices.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors,

compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing

version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.

Exhibit A - Form of Secondary Licenses Notice

"This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: {name license(s), version(s), and exceptions or additional permissions here}."

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor
Boston, MA 02110-1335
USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to

make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers

to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major

components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under

any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software

Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1335 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type
`show w'. This is free software, and you are welcome to redistribute
it under certain conditions; type `show c' for details.
```

The hypothetical commands ``show w'` and ``show c'` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than ``show w'` and ``show c'`; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the
program `Gnomovision' (which makes passes at compilers) written by
James Hacker.
```

```
signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

CLASSPATH EXCEPTION

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked

independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

1.77 latencyutils 2.0.3

1.77.1 Available under license :

```
PK0P=spring-cloud-starter-netflix-ribbon-2.1.2.RELEASE-sources.jarg4G7C{ F-`DutCD !QG7Dwy>9u:g?{]ic`pqq
_Uh*C5!Jz`p4w{HIE`1z>O<(X
IRq8>H+Y/^*xBZ8}LX1~11_fg7s{{`n`[W_.;2|csqpu9x2IETH=a~v<\.0'i>/d1!4F&U5xV~VBEGA?)\O7pk"i)K3 )
.#RScs7,_D/_z,:*^+;J~1XE3.<%wt_3[5h]v7S,Dw5bZLI=C3tMDVrJLn9?!,Sthp
'sn}S`o [~ {:8@SI7R$
1f+U7L,`,$Ts,VIow^Mk0p_5S#[n.A **h"v\NK:qvO< jRT2'eC@Br-KuY.czg]-\&VwiJ6W]Z>0E;zjs~OG_
g+8`?<$O{7@-A'QR+&Uz.+GIL8UODo9wTsV"yCYjI iWmz^'pkJ pd^V{NjW)x|U#
&V~/G|{Jj`ySR}gag{M}OL>0av
"nJ9${,VdS} .~E8[MNbW/
kN.Emkhuo~)SY_[3p^Pu#,`c
]Jcoig;C^CxJ6LY!sN~|~B3TKFxAi#g@Z@%glo:L$~A }rno$Opv^$C&=Ut+AU FF[7U-17{a]Mc
bDcK*B*^HC:>kfIV5fZu,Tt<j
JsmXl49y[iu"6;`UNf&9OnF9i6h;tXhM())wb_UJqiZ_:)1&|,)P(~1wV,^dU_V+m9kmKw\Lj c:47:@U3^>/-
lu58cxw_?},|2^Re+ Ao\%IAIeJ02vB;iMASvF}
Ht|a8Q>}g6%2%~k{2##Hs-.7ZY%k*= 0{:_?d:#"8m"sq(DEZR#T?N%kx~IHNvJK
[Kz2n'JI.I8sX)U4vXcej),4{tEgiF9J DL$J,(Onb)cJY)A2N&-dRrE/3D2OLAHRR=*+bWQgbE_UG{Y-
$&6rX@_HA"K /H(VO9C9O%FZEn6{F BtKDK1zD%u28J<kI+p%HJvQH|q
S.YCm<83=Fn8p $S2EDP$"u4x"ym
ZN!cS<a8,Iyz7KIw{{*xO#B\-*}mD $RGR;Fp0o0V00B#c=4skVTY04P@c4gqO*82myazgQ<_/,*Zm_M|e.
nKbdE ^O^yTd/$Vu|=Q HJJ_k0YzH+;+Wb=u`xm
FFbkCs<dJ%_b ht&Sq:)?*TfEe{)E% @mUxT8W}y3 Z8BEFM}tEa7b,gKMUAU'07wm8AQ-+)
D\y:=GlmiN.rh;o$Tsqmv}j^m}C>qT?<+^3g5&'Ac%/ei3hq9R = Npo<.Guc{^LmM9^o=!%71%?^4cd9Hv!-
HN{2+t2Un{&sIBsfw<ov_I<vcfF
$XZE5zAocaf!"RUy6QRabLDkT)
Ce0K_ JW@"cEo,Z~Oru#U0NNymq}y56@HXl<_D 6s9:/`EZ91JFR"u:tu }/ksKanG`X2
Nk>!wn!J'wu+,liM7 e&V|b
XQU{3},6:BZa852ES$fk|s'F"*w<R47YXv'Z+DXsE1[5Kxcps`aK1dm!p~kSsct,Xo~j$Yn]RYA;D1y-
q/_\UP<ujCQ9/V@OiXj>].+ZV_]Mo/^)?d*NPX Ns![zEVB{P0"5".?H3N1}QZ{,4X{.Lvn=<n5t;1
'}7e/@q\
EF2@wr'Hs>|J{h2hI,-oIfU_bF;60EaGd/uV./+WJ(?>ttxuJhTK^G}Hxwd*DGi"mU(QVN-
NXWUGJaSy3^JSTh#M^ngogsJ k+org]Thcr{+vIDE=kL3>Fr~"VH4~(^ P/dSRA\H-mbzdckf{`-
3amB~eQ4$WKg5F#*=Qx&% W`d
7=
2&Ka L6klntKG3'{hOX[D@"6??&$AZKjVBYIN.||6~k}5{W~onI=RjsvXa|YF1+~a+.Yhhj^8 A
({#.][v!sR'5<#.^Q~8zUft#>]l0fvDyTV_v#,x|W
Q;
mmWfw|&+G'IR_VJ: ~B9_P6>oFX!)~u,?zPKDcPK9ZP6spring-cloud-netflix-sidecar-2.2.2.RELEASE-
```

sources.jarzex]9c1ffff333C113cq3_` }GVRTz\$A`r}
AW4uo]ZDI #J/- QTU`er,SHrbQ%Qb;p7d
acm6}0o/Fc_H`__S!b0_lu;0K?E~s'8~D-x3sTV\@jP=AKlgNi
}bIjw:FhPf:HJ/+rTo*WDqnG(LJ\$JLj`eE#5_qfCvO)Tdn7L`T`=Brl4;yr.}#Ebsu<Eke@
O'Q#=#TQ DKg!foY~i"AA,#"m2F[C~#pj0~%K'ik^RQpUvPdWQG[Uu5Y)K(RT@yq`\$-4a LIHgfBRe&9h-
e1\Fa~i%+\MtW*+Blzi-p7&Q1v'4U{roR d[~g~65bSS{-dBU"7)7Enmi{K&6Lum(^)c7KU
a2mMbW+C\$(>~e8@@llpu+%vgNhDxX%
Tf*"\$8f*f(a[Xn@\$ {YQa/=HHHb,WAG
p>a66ff,iz0D)d|_7}i+{TFjCp}WX&k1v-I tst]B <)cri!!;D4+
%!F9\DaP7KF*mn >`
HOM-_/;ta?]w^JzC] @!VaN=\$ODc%*CsI
,wc#HH-
`Z7>3CrZR\$qwCm'/p]R!/)r% ?Z]Wg^zm{. {!HR@:.,D}&k'1cU6/h}(5VtAVeE4vM.<%&<|+93UUIc2t~Ig#,YiA9H
/-h`p^i8.UIVnF|x-&e<O?dAtHCDj"
c`c<seJMS\$*jKr?q&cW AY: XsU/JB!5(yADY)@
e@Cj
J/J)-x53mNxHZi
7rxGwczU`mWl{T4CZd"zpphU`jU
=3X[UtteFhvE+]U+V;]cK?RM(,9W5A(m9jBkQ~sN'yqg"}j9K)thHJ1NDo\$z3n:IiAjh%ACWj;
-mj^
axurii1k/VVtPY]0|<
w\5g.lj-N)V<:y%t4c9 RNS]3(`#dcmlfh`fc{ \,W_NTDo(%EQD2i=W_QH" _N%|IC~F+S
zj7i(AvyPrIe)7*fNf4VwJuhA2,rS~\$q\$ywuWX(#B`D==>*/9Nt*1d{ OApPd/pdPb@##@8dUd)Y-xQht-
RCU@^5'3NW{A8R f"0]=Eu\C]S\$aIQ=<v8I80jHOeO.l/Y\$E;Eh8%6J\~Q!:9_#h`h@/
4G&mZX3H>v. X98"MBb9 [,\B]tUhMu@EldpXM*;#C[{>n + t;l`<SjPzWnSHeV@o:peR^!@:olx?K]
t
>tc>ILD2C2H+W,D<g5E|N-Ve5 &M<Pc*80SB7Ox)Dg3#/G0f/e5
@ntau1;n{]L
.-&jzSy*q^1G!g&S
q"v{Bltz4od^ZrIUq`fbo3GCp^:8x+FRC,uQriTL[xSO2'Z[0+s|4*#>SGzM]I].->\$905E]3b)8Us9y1MjLk}Gq-
W|">%V
Snk87]U\$Mv{m1 {<.,z;17}W8\$8UCs&SN4mc.F(@BH\$K #8G#g9GD@T3#?LYoD9
h;GyW%>5pNk;S0iyFbrS-]p[N8Aq]E3d37gkbnOy3 Q_v9\q&2{9`H*,/c]F6dyT"x>Nwb
kP\h,ho7r|7h:bR.)o=U
3jpdpjBS>6;)-k50|!
dym:=& !#p[hz=^?*w
:bLWc-C (z\$SK8OLjJ4:\e|4*U*YIEI,?"u'mKQ)|C7)bg{-sv
2#~YSR "QJ]
LZ]dJ5y;6?6Y)Q+EkW#TN--Q3dxmg|PIIO0m"ZG%>btV\%?J,MsL^5N31):^f&05CD
8eDgKt" _U#H?E-)YE
4'v61gStI7k\$vw^dQ
g9*B,RYEC0}6I" ,NY^yl/W;ji7t1-GdTyK~1mQa.#1X)ayorAMJK}#Q9M~J8\$*+LOIm#
{scW7F/[l0xxB?!(Kt?qpkuh\$
o7[4.@rDQ{Bc(U[-b;zw" |(\$x=0@[xq:b#V65L!IEk<qOuQO%m
gDDwY#x+]C=odD?W8Hp+o'noI&?I#,{ "m}z*A4/O>z5J@4Wf,h\twk3D#CM1O&ND{blCPV|P)^Ds'A&lom^TC7
C>e,O]K@.olsoR@cNR~~YKuYV~*\$cW2dUCtdap/TzF

LN[pY-bFv9UJ3nbVzd8-3/:dhgqF5)#G/1>fkuV{[Hp&2Vf4NOw%FiU5.0}|4oBc Wn;'&SU2HT@Tg*42` }=mW Yb

;hKs~MU<2 eIbaU:Lpgyc"svh>*bDW2TxiHrcBQU1;&QA`>l8
/KT?pAX;^[hmK y>7??4C[,ecg)`kkifY:
{<,mH>of>Pv9%EH4_VSaqx3S.s;lji2?Ci1-IH9N~F)bH8`p`|IKZ(;GB0^xt
RDS23v+iCL#F>Cvkqeq\$
gcj%
<Tze"&-7\$qm E6Z##.|#k>|6S9/c3Wu+Mx44Rbl9dJ6)T=8yDI2g
fSL>5M_<{:3S9Y%zmvRrhde;)T>[+|=|O_-liv-65 o#wm8'c"7/cGjs(9yO5QkD(!9{/0
{m\NXAtJCI!uc"<B<IP.DAaY/+;y9MmTJ
i>SPB2CuTR{4K\$P8?V 7}"
/jdD;^ c^KI|+}?Y?T0hHDj*J{U\$VA&#^N w%*MWA
XIoy9-aR.[1ep&(4nK%EEu_Yk!Y^*TuOy0 &iJ9a]?\$
mUJrlf1gd<-T~~tz{^bbxTRZbTv~xNrx~JvHv|BrDV0,CE
+6=^}NrF\Rzjzjt<L7\h6RfX=;XZ2mYt
FrE|]k(!m;D;xtAK9#w<=@*>S\$R*^:qD8,~3A=G*Bj(6ePy~R%
0\$^I]2]>/"(sE-E:~xNEx)
;YZtk%jTp[XO5*p4,b@n)gye+S[&GF}zD!|\2r5|xR9x^XDvL xH%e(pExM}b-(4jWY#;>.r8&MJy#hD+|sF"2uQ
|H"bX9KZz<6J?~liW* Y_o|O|Fp2NEQ)c\$)0D9mIhW{`[h x(v0_a)5Tqn:\$k|LxPWO
P~@lh>Zz~KE9U0sU* }l>qdF@ :%
\$kf phCU#:%aohb5h
sm7iCY/[64::w/aRM.-c4szto@`onq?J?zQANROBtppz8qp5H-g+]
+mVg-@W4MHWnv\$}kCxm `qB[8eLiU0mnMy4g&`
^Qx
vd*yC5S~--+ |2 _>\$Z5:xz/i.]{p8fM78m}kVO8PHeMGTi:~\$0JHcFyaq2&<~
jg4ylkX"+*z9{L
o(B|7|{|h-ICc!s6zJVE.3RK`pDIL.RG6#swX+GBuRg5mlM;|IF}DJ#QZuU=G m\06WaLh3SZh _`SA8va\$/c)`6
@h1]C
.!=
"wj\5^XW6|OV0|};^k3i^
9ws_Iq:D=g '!PX8{6Rh8*I|[Tv1^8!c/K1g3Nzj}_pSER"%F_B7F}mr"oS/]PPE:)VFXNk'47p4ZUK4

CA=6JjVHj2Lv[Gy(? "x
Mi0p=M=r)wO[YPUD1% rkX4yMc
k`WR/ABV+qKP?4Z~&`J5F?./;QjxYg7!"q8qMk&p-.Lq3aqolASrz#b"Koq3:7Ky0WZ6D+IB\UPU2j;\$LVs^m-
+ETHfF

A@i{nVl\BCS!srY rq25>rgYS3\$`?kLjI>{GJwwqS<s9thIWR8_aHD]Zs },[1`uE1H4zG5.2khKPx]/g{?
k1B(P^%*Vmo?X,*r;*\$_~9[B\jr~0a`F1P0ry\$^Qa^?VL{ljdPzKf: v?=oqM~@RTob(aO3i
[GJ.gNWA6La7.QcBz1" 451'cliDC`i|8[Ay0(.8{_)xO`5
4R,r]QNo >UYT<
r?=3LaAnfO"Zo;1mu?.5z+bYhX %
/"7J|C|8qoHjW7Ju~.{ fP 'ad mw7OiG|nsi
u~2r@_3v [Fg@.Nmjino\$'+q\$}4Wae3r\ZK
[Z6QXf+.s~Y9Wz`e9%)OnI.-VZHM^t:r*ppkMd# Y!Nn6fxs/
a23pT]sO,rpF;&gq|dL)BsZ{mtwciPt=[5hmUSA@ 9t*bSI?@IbO_!`?! G/qU'w><00` aEw`?|Nlu ;~%~
WFwm0G;:_
w>Q6W>
0?yU|fI7W.~w\$8oh~*TPK'&2PK0PDC=spring-cloud-starter-netflix-ribbon-2.1.2.RELEASE-
sources.jarPK9ZP'&26spring-cloud-netflix-sidecar-2.2.2.RELEASE-sources.jarPK=

1.78 micronaut-test 3.1.1

1.78.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2017-2020 original authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1331474237_1653506865.917328/0/micronaut-test-junit5-3-1-1-sources-
jar/io/micronaut/test/extensions/junit5/annotation/MicronautTest.java
* /opt/cola/permits/1331474237_1653506865.917328/0/micronaut-test-junit5-3-1-1-sources-
jar/io/micronaut/test/extensions/junit5/MicronautJUnit5Extension.java
* /opt/cola/permits/1331474237_1653506865.917328/0/micronaut-test-junit5-3-1-1-sources-
jar/io/micronaut/test/extensions/junit5/graal/MicronautJUnit5Gaal.java
```

1.79 curator-client 5.2.1

1.79.1 Available under license :

Curator Client
Copyright 2011-2022 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent

to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work,

excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any

risks associated with Your exercise of permissions under this License.

8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.80 google-guice 4.2.3

1.80.1 Available under license :

Google Guice - Core Library
Copyright 2006-2020 Google, Inc.

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You

institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.81 cloudevents---kafka-transport-binding

2.2.0

1.81.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
<!--
~ Copyright 2018-Present The CloudEvents Authors
~ <p>
~ Licensed under the Apache License, Version 2.0 (the "License");
~ you may not use this file except in compliance with the License.
~ You may obtain a copy of the License at
~ <p>
~ http://www.apache.org/licenses/LICENSE-2.0
~ <p>
~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS,
~ WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
~ See the License for the specific language governing permissions and
~ limitations under the License.
~
```

-->

Found in path(s):

* /opt/cola/permits/1340816289_1654861242.1326292/0/cloudevents-kafka-2-2-0-sources-jar/META-INF/maven/io.cloudevents/cloudevents-kafka/pom.xml

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2018-Present The CloudEvents Authors

* <p>

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

* <p>

* <http://www.apache.org/licenses/LICENSE-2.0>

* <p>

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*

*/

Found in path(s):

* /opt/cola/permits/1340816289_1654861242.1326292/0/cloudevents-kafka-2-2-0-sources-jar/io/cloudevents/kafka/CloudEventMessageDeserializer.java

* /opt/cola/permits/1340816289_1654861242.1326292/0/cloudevents-kafka-2-2-0-sources-jar/io/cloudevents/kafka/CloudEventMessageSerializer.java

* /opt/cola/permits/1340816289_1654861242.1326292/0/cloudevents-kafka-2-2-0-sources-jar/io/cloudevents/kafka/CloudEventSerializer.java

* /opt/cola/permits/1340816289_1654861242.1326292/0/cloudevents-kafka-2-2-0-sources-jar/io/cloudevents/kafka/CloudEventDeserializer.java

* /opt/cola/permits/1340816289_1654861242.1326292/0/cloudevents-kafka-2-2-0-sources-jar/io/cloudevents/kafka/impl/KafkaBinaryMessageReaderImpl.java

* /opt/cola/permits/1340816289_1654861242.1326292/0/cloudevents-kafka-2-2-0-sources-jar/io/cloudevents/kafka/impl/KafkaProducerMessageWriterImpl.java

* /opt/cola/permits/1340816289_1654861242.1326292/0/cloudevents-kafka-2-2-0-sources-jar/io/cloudevents/kafka/KafkaMessageFactory.java

* /opt/cola/permits/1340816289_1654861242.1326292/0/cloudevents-kafka-2-2-0-sources-jar/io/cloudevents/kafka/impl/KafkaHeaders.java

* /opt/cola/permits/1340816289_1654861242.1326292/0/cloudevents-kafka-2-2-0-sources-jar/io/cloudevents/kafka/impl/KafkaSerializerMessageWriterImpl.java

* /opt/cola/permits/1340816289_1654861242.1326292/0/cloudevents-kafka-2-2-0-sources-jar/io/cloudevents/kafka/impl/BaseKafkaMessageWriterImpl.java

1.82 micronaut-reactor 2.4.1

1.82.1 Available under license :

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain

separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the

origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.83 elasticsearch-kubernetes-cloud-plugin

4.13.3

1.83.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
#  
# Copyright (C) 2015 Red Hat, Inc.  
#  
# Licensed under the Apache License, Version 2.0 (the "License");  
# you may not use this file except in compliance with the License.  
# You may obtain a copy of the License at  
#  
# http://www.apache.org/licenses/LICENSE-2.0  
#  
# Unless required by applicable law or agreed to in writing, software  
# distributed under the License is distributed on an "AS IS" BASIS,  
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
# See the License for the specific language governing permissions and  
# limitations under the License.  
#
```

```
io.fabric8.kubernetes.client.URLFromEnvVarsImpl  
io.fabric8.kubernetes.client.URLFromIngressImpl  
io.fabric8.kubernetes.client.URLFromNodePortImpl  
io.fabric8.kubernetes.client.URLFromClusterIPImpl
```

Found in path(s):

```
* /opt/cola/permits/1288519866_1647863658.86/0/kubernetes-client-4-13-3-jar/META-  
INF/services/io.fabric8.kubernetes.client.ServiceToURLProvider  
No license file was found, but licenses were detected in source scan.
```

Copyright (C) 2015 Red Hat, Inc.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE>

2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Found in path(s):

* /opt/cola/permits/1288519866_1647863658.86/0/kubernetes-client-4-13-3-jar/META-
INF/maven/io.fabric8/kubernetes-client/pom.xml

No license file was found, but licenses were detected in source scan.

#

Copyright (C) 2015 Red Hat, Inc.

#

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

#

<http://www.apache.org/licenses/LICENSE-2.0>

#

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

#

io.fabric8.kubernetes.client.AppsAPIGroupExtensionAdapter

io.fabric8.kubernetes.client.AdmissionRegistrationAPIGroupExtensionAdapter

io.fabric8.kubernetes.client.V1AdmissionRegistrationAPIGroupExtensionAdapter

io.fabric8.kubernetes.client.V1beta1AdmissionRegistrationAPIGroupExtensionAdapter

io.fabric8.kubernetes.client.AutoscalingAPIGroupExtensionAdapter

io.fabric8.kubernetes.client.ApiextensionsAPIGroupExtensionAdapter

io.fabric8.kubernetes.client.AuthorizationAPIGroupExtensionAdapter

io.fabric8.kubernetes.client.V1AutoscalingAPIGroupExtensionAdapter

io.fabric8.kubernetes.client.V2beta1AutoscalingAPIGroupExtensionAdapter

io.fabric8.kubernetes.client.V2beta2AutoscalingAPIGroupExtensionAdapter

io.fabric8.kubernetes.client.BatchAPIGroupExtensionAdapter

io.fabric8.kubernetes.client.ExtensionsAPIGroupExtensionAdapter

io.fabric8.kubernetes.client.MetricAPIGroupExtensionAdapter
io.fabric8.kubernetes.client.NetworkAPIGroupExtensionAdapter
io.fabric8.kubernetes.client.PolicyAPIGroupExtensionAdapter
io.fabric8.kubernetes.client.RbacAPIGroupExtensionAdapter
io.fabric8.kubernetes.client.SchedulingAPIGroupExtensionAdapter
io.fabric8.kubernetes.client.SettingsAPIGroupExtensionAdapter
io.fabric8.kubernetes.client.StorageAPIGroupExtensionAdapter
io.fabric8.kubernetes.client.V1APIGroupExtensionAdapter
io.fabric8.kubernetes.client.V1ApiextensionsAPIGroupExtensionAdapter
io.fabric8.kubernetes.client.V1beta1ApiextensionsAPIGroupExtensionAdapter
io.fabric8.kubernetes.client.V1AuthorizationAPIGroupExtensionAdapter
io.fabric8.kubernetes.client.V1beta1AuthorizationAPIGroupExtensionAdapter
io.fabric8.kubernetes.client.V1NetworkAPIGroupExtensionAdapter
io.fabric8.kubernetes.client.V1beta1NetworkAPIGroupExtensionAdapter

Found in path(s):

* /opt/cola/permits/1288519866_1647863658.86/0/kubernetes-client-4-13-3-jar/META-INF/services/io.fabric8.kubernetes.client.ExtensionAdapter

1.84 netty-tomcatnative-openssl-dynamic

2.0.54.Final

1.84.1 Available under license :

LICENSE ISSUES

=====

The OpenSSL toolkit stays under a double license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. See below for the actual license texts.

OpenSSL License

/* =====

* Copyright (c) 1998-2019 The OpenSSL Project. All rights reserved.

*

* Redistribution and use in source and binary forms, with or without

* modification, are permitted provided that the following conditions

* are met:

*

* 1. Redistributions of source code must retain the above copyright

* notice, this list of conditions and the following disclaimer.

*

* 2. Redistributions in binary form must reproduce the above copyright

* notice, this list of conditions and the following disclaimer in

* the documentation and/or other materials provided with the
 * distribution.
 *

* 3. All advertising materials mentioning features or use of this
 * software must display the following acknowledgment:
 * "This product includes software developed by the OpenSSL Project
 * for use in the OpenSSL Toolkit. (<http://www.openssl.org/>)"
 *

* 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
 * endorse or promote products derived from this software without
 * prior written permission. For written permission, please contact
 * openssl-core@openssl.org.
 *

* 5. Products derived from this software may not be called "OpenSSL"
 * nor may "OpenSSL" appear in their names without prior written
 * permission of the OpenSSL Project.
 *

* 6. Redistributions of any form whatsoever must retain the following
 * acknowledgment:
 * "This product includes software developed by the OpenSSL Project
 * for use in the OpenSSL Toolkit (<http://www.openssl.org/>)"
 *

* THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS" AND ANY
 * EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
 * PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR
 * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
 * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
 * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
 * OF THE POSSIBILITY OF SUCH DAMAGE.
 * =====
 *

* This product includes cryptographic software written by Eric Young
 * (ey@cryptsoft.com). This product includes software written by Tim
 * Hudson (tjh@cryptsoft.com).
 *
 */

Original SSLeay License

/* Copyright (C) 1995-1998 Eric Young (ey@cryptsoft.com)
 * All rights reserved.
 *

* This package is an SSL implementation written
 * by Eric Young (eay@cryptsoft.com).
 * The implementation was written so as to conform with Netscapes SSL.
 *
 * This library is free for commercial and non-commercial use as long as
 * the following conditions are aheared to. The following conditions
 * apply to all code found in this distribution, be it the RC4, RSA,
 * lhash, DES, etc., code; not just the SSL code. The SSL documentation
 * included with this distribution is covered by the same copyright terms
 * except that the holder is Tim Hudson (tjh@cryptsoft.com).
 *
 * Copyright remains Eric Young's, and as such any Copyright notices in
 * the code are not to be removed.
 * If this package is used in a product, Eric Young should be given attribution
 * as the author of the parts of the library used.
 * This can be in the form of a textual message at program startup or
 * in documentation (online or textual) provided with the package.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * 3. All advertising materials mentioning features or use of this software
 * must display the following acknowledgement:
 * "This product includes cryptographic software written by
 * Eric Young (eay@cryptsoft.com)"
 * The word 'cryptographic' can be left out if the rouines from the library
 * being used are not cryptographic related :-).
 * 4. If you include any Windows specific code (or a derivative thereof) from
 * the apps directory (application code) you must include an acknowledgement:
 * "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
 *
 * THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS" AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 *

- * The licence and distribution terms for any publically available version or
- * derivative of this code cannot be changed. i.e. this code cannot simply be
- * copied and put under another distribution licence
- * [including the GNU Public Licence.]
- */

BoringSSL is a fork of OpenSSL. As such, large parts of it fall under OpenSSL licensing. Files that are completely new have a Google copyright and an ISC license. This license is reproduced at the bottom of this file.

Contributors to BoringSSL are required to follow the CLA rules for Chromium:
<https://cla.developers.google.com/cla>

Files in third_party/ have their own licenses, as described therein. The MIT license, for third_party/ fiat, which, unlike other third_party directories, is compiled into non-test libraries, is included below.

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. See below for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL please contact openssl-core@openssl.org.

The following are Google-internal bug numbers where explicit permission from some authors is recorded for use of their work. (This is purely for our own record keeping.)

27287199

27287880

27287883

OpenSSL License

/* =====

* Copyright (c) 1998-2011 The OpenSSL Project. All rights reserved.

*

* Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:

*

* 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.

*

* 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in
 * the documentation and/or other materials provided with the
 * distribution.

*

* 3. All advertising materials mentioning features or use of this
 * software must display the following acknowledgment:
 * "This product includes software developed by the OpenSSL Project
 * for use in the OpenSSL Toolkit. (<http://www.openssl.org/>)"

*

* 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
 * endorse or promote products derived from this software without

* prior written permission. For written permission, please contact
 * openssl-core@openssl.org.
 *

* 5. Products derived from this software may not be called "OpenSSL"
 * nor may "OpenSSL" appear in their names without prior written
 * permission of the OpenSSL Project.
 *

* 6. Redistributions of any form whatsoever must retain the following
 * acknowledgment:
 * "This product includes software developed by the OpenSSL Project
 * for use in the OpenSSL Toolkit (<http://www.openssl.org/>)"
 *

* THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS" AND ANY
 * EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
 * PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR
 * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
 * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
 * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
 * OF THE POSSIBILITY OF SUCH DAMAGE.
 * =====
 *

* This product includes cryptographic software written by Eric Young
 * (eay@cryptsoft.com). This product includes software written by Tim
 * Hudson (tjh@cryptsoft.com).
 *

*/
 Original SSLeay License

 /* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
 * All rights reserved.
 *

* This package is an SSL implementation written
 * by Eric Young (eay@cryptsoft.com).
 * The implementation was written so as to conform with Netscapes SSL.
 *

* This library is free for commercial and non-commercial use as long as
 * the following conditions are aheared to. The following conditions
 * apply to all code found in this distribution, be it the RC4, RSA,
 * lhash, DES, etc., code; not just the SSL code. The SSL documentation
 * included with this distribution is covered by the same copyright terms
 * except that the holder is Tim Hudson (tjh@cryptsoft.com).
 *

* Copyright remains Eric Young's, and as such any Copyright notices in

* the code are not to be removed.

* If this package is used in a product, Eric Young should be given attribution

* as the author of the parts of the library used.

* This can be in the form of a textual message at program startup or

* in documentation (online or textual) provided with the package.

*

* Redistribution and use in source and binary forms, with or without

* modification, are permitted provided that the following conditions

* are met:

* 1. Redistributions of source code must retain the copyright

* notice, this list of conditions and the following disclaimer.

* 2. Redistributions in binary form must reproduce the above copyright

* notice, this list of conditions and the following disclaimer in the

* documentation and/or other materials provided with the distribution.

* 3. All advertising materials mentioning features or use of this software

* must display the following acknowledgement:

* "This product includes cryptographic software written by

* Eric Young (eay@cryptsoft.com)"

* The word 'cryptographic' can be left out if the routines from the library

* being used are not cryptographic related :-).

* 4. If you include any Windows specific code (or a derivative thereof) from

* the apps directory (application code) you must include an acknowledgement:

* "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"

*

* THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS" AND

* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE

* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE

* ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE

* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL

* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS

* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)

* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT

* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY

* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF

* SUCH DAMAGE.

*

* The licence and distribution terms for any publically available version or

* derivative of this code cannot be changed. i.e. this code cannot simply be

* copied and put under another distribution licence

* [including the GNU Public Licence.]

*/

ISC license used for completely new code in BoringSSL:

/* Copyright (c) 2015, Google Inc.

*

* Permission to use, copy, modify, and/or distribute this software for any

* purpose with or without fee is hereby granted, provided that the above

* copyright notice and this permission notice appear in all copies.

*

* THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
* WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
* MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY
* SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
* WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION
* OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN
* CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. */

The code in third_party/ fiat carries the MIT license:

Copyright (c) 2015-2016 the fiat-crypto authors (see
<https://github.com/mit-plv/fiat-crypto/blob/master/AUTHORS>).

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

Licenses for support code

Parts of the TLS test suite are under the Go license. This code is not included
in BoringSSL (i.e. libcrypto and libssl) when compiled, however, so
distributing code linked against BoringSSL does not trigger this license:

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.

* Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,

SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

BoringSSL uses the Chromium test infrastructure to run a continuous build, trybots etc. The scripts which manage this, and the script for generating build metadata, are under the Chromium license. Distributing code linked against BoringSSL does not trigger this license.

Copyright 2015 The Chromium Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity

on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one

of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a

result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

The Netty Project

=====

Please visit the Netty web site for more information:

* <http://netty.io/>

Copyright 2016 The Netty Project

The Netty Project licenses this file to you under the Apache License, version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at:

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

This product contains a forked and modified version of Tomcat Native

- * LICENSE:
 - * [license/LICENSE.tomcat-native.txt](#) (Apache License 2.0)
- * HOMEPAGE:
 - * <http://tomcat.apache.org/native-doc/>
 - * <https://svn.apache.org/repos/asf/tomcat/native/>

This product contains the Maven wrapper scripts from 'Maven Wrapper', that provides an easy way to ensure a user has everything necessary to run the Maven build.

- * LICENSE:
 - * [license/LICENSE.mvn-wrapper.txt](#) (Apache License 2.0)
- * HOMEPAGE:
 - * <https://github.com/takari/maven-wrapper>

This product contains small piece of code to support AIX, taken from netbsd.

- * LICENSE:
 - * [license/LICENSE.aix-netbsd.txt](#) (OpenSSL License)
- * HOMEPAGE:
 - * <https://ftp.netbsd.org/pub/NetBSD/NetBSD-current/src/crypto/external/bsd/openssl/dist>

This product contains code from boringssl.

- * LICENSE (Combination ISC and OpenSSL license)
 - * [license/LICENSE.boringssl.txt](#) (Combination ISC and OpenSSL license)
- * HOMEPAGE:
 - * <https://boringssl.googlesource.com/boringssl/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or

agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.85 apache-yetus-audience-annotations

0.12.0

1.85.1 Available under license :

Apache Yetus
Copyright 2008-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<https://www.apache.org/>).

Additional licenses for the Apache Yetus Source/Website:

See LICENSE for terms.

```
# Licensed to the Apache Software Foundation (ASF) under one or more
# contributor license agreements. See the NOTICE file distributed with
# this work for additional information regarding copyright ownership.
# The ASF licenses this file to You under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License. You may obtain a copy of the License at
#
# http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
```

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of

the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works

that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A

PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Additional licenses for the Apache Yetus Source/Website:

This project incorporates portions of the Bootstrap project available under the MIT license:

The MIT License (MIT)

Copyright (c) 2011-2015 Twitter, Inc

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This project incorporates NORMALIZE.css as bundled with the Twitter Bootstrap project which is released under the same license as Bootstrap.

Copyright Nicolas Gallagher and Jonathan Neal

This project incorporates GLYPHICONS FREE as bundled with the Twitter Bootstrap project which are released under the same license as Bootstrap.

Copyright (c) 2010 - 2015 Jan Kovarik

This project incorporates portions of the Font Awesome project available under the MIT license and SIL OFL 1.1 .

Copyright (c) 2015 Dave Gandy

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

SIL OPEN FONT LICENSE Version 1.1 - 26 February 2007

PREAMBLE

The goals of the Open Font License (OFL) are to stimulate worldwide development of collaborative font projects, to support the font creation efforts of academic and linguistic communities, and to provide a free and open framework in which fonts may be shared and improved in partnership with others.

The OFL allows the licensed fonts to be used, studied, modified and redistributed freely as long as they are not sold by themselves. The fonts, including any derivative works, can be bundled, embedded, redistributed and/or sold with any software provided that any reserved names are not used by derivative works. The fonts and derivatives, however, cannot be released under any other type of license. The requirement for fonts to remain under this license does not apply to any document created using the fonts or their derivatives.

DEFINITIONS

"Font Software" refers to the set of files released by the Copyright Holder(s) under this license and clearly marked as such. This may include source files, build scripts and documentation.

"Reserved Font Name" refers to any names specified as such after the copyright statement(s).

"Original Version" refers to the collection of Font Software components as distributed by the Copyright Holder(s).

"Modified Version" refers to any derivative made by adding to, deleting, or substituting -- in part or in whole -- any of the components of the Original Version, by changing formats or by porting the Font Software to a new environment.

"Author" refers to any designer, engineer, programmer, technical writer or other person who contributed to the Font Software.

PERMISSION & CONDITIONS

Permission is hereby granted, free of charge, to any person obtaining a copy of the Font Software, to use, study, copy, merge, embed, modify, redistribute, and sell modified and unmodified copies of the Font Software, subject to the following conditions:

- 1) Neither the Font Software nor any of its individual components, in Original or Modified Versions, may be sold by itself.
- 2) Original or Modified Versions of the Font Software may be bundled, redistributed and/or sold with any software, provided that each copy contains the above copyright notice and this license. These can be included either as stand-alone text files, human-readable headers or in the appropriate machine-readable metadata fields within text or binary files as long as those fields can be easily viewed by the user.
- 3) No Modified Version of the Font Software may use the Reserved Font Name(s) unless explicit written permission is granted by the corresponding Copyright Holder. This restriction only applies to the primary font name as presented to the users.
- 4) The name(s) of the Copyright Holder(s) or the Author(s) of the Font Software shall not be used to promote, endorse or advertise any Modified Version, except to acknowledge the contribution(s) of the Copyright Holder(s) and the Author(s) or with their explicit written permission.
- 5) The Font Software, modified or unmodified, in part or in whole, must be distributed entirely under this license, and must not be

distributed under any other license. The requirement for fonts to remain under this license does not apply to any document created using the Font Software.

TERMINATION

This license becomes null and void if any of the above conditions are not met.

DISCLAIMER

THE FONT SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF COPYRIGHT, PATENT, TRADEMARK, OR OTHER RIGHT. IN NO EVENT SHALL THE COPYRIGHT HOLDER BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, INCLUDING ANY GENERAL, SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF THE USE OR INABILITY TO USE THE FONT SOFTWARE OR FROM OTHER DEALINGS IN THE FONT SOFTWARE.

This project incorporates portions of the JQuery project available under the MIT license:

Copyright jQuery Foundation and other contributors, <https://jquery.org/>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This project incorporates via jQuery portions of the Sizzle project

available under the MIT license:

Copyright JS Foundation and other contributors, <https://js.foundation/>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at <https://github.com/jquery/sizzle>

The following license applies to all parts of this software except as documented below:

====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This project utilizes Jython 2.7 for running Python code on JVMs. It is available under the Python Software Foundation License v2:

PYTHON SOFTWARE FOUNDATION LICENSE VERSION 2

1. This LICENSE AGREEMENT is between the Python Software Foundation ("PSF"), and the Individual or Organization ("Licensee") accessing and otherwise using this software ("Jython") in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, PSF hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly,

prepare derivative works, distribute, and otherwise use Jython alone or in any derivative version, provided, however, that PSF's License Agreement and PSF's notice of copyright, i.e., "Copyright (c) 2007 Python Software Foundation; All Rights Reserved" are retained in Jython alone or in any derivative version prepared by Licensee.

3. In the event Licensee prepares a derivative work that is based on or incorporates Jython or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Jython.

4. PSF is making Jython available to Licensee on an "AS IS" basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF JYTHON WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF JYTHON FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING JYTHON, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between PSF and Licensee. This License Agreement does not grant permission to use PSF trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By copying, installing or otherwise using Jython, Licensee agrees to be bound by the terms and conditions of this License Agreement.

1.86 micronaut-for-spring 4.3.1

1.86.1 Available under license :

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to

communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of

the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

1.87 aws-java-sdk-:::bill-of-materials 2.17.271

1.87.1 Available under license :

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the

editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the

same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.88 fabric8-::-kubernetes-model-::-storage-class 4.13.3

1.88.1 Available under license :

No license file was found, but licenses were detected in source scan.

*

* Copyright (C) 2015 Red Hat, Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*

Found in path(s):

* /opt/cola/permits/1288519903_1647861506.61/0/kubernetes-model-storageclass-4-13-3-jar/manifest.vm

No license file was found, but licenses were detected in source scan.

Copyright (C) 2015 Red Hat, Inc.

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE>

2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

Found in path(s):

* /opt/cola/permits/1288519903_1647861506.61/0/kubernetes-model-storageclass-4-13-3-jar/META-INF/maven/io.fabric8/kubernetes-model-storageclass/pom.xml

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0

Bnd-LastModified: 1619068667397

Build-Jdk-Spec: 1.8

Bundle-Description: Java client for Kubernetes and OpenShift

Bundle-DocURL: <http://redhat.com>

Bundle-License: <http://www.apache.org/licenses/LICENSE-2.0.txt>

Bundle-ManifestVersion: 2

Bundle-Name: Fabric8 :: Kubernetes Model :: Storage Class

Bundle-SymbolicName: io.fabric8.kubernetes-model-storageclass

Bundle-Vendor: Red Hat

Bundle-Version: 4.13.3

Created-By: Apache Maven Bundle Plugin

Export-Package: io.fabric8.kubernetes.api.model.storage;uses:="com.fasterxml.jackson.annotation,com.fasterxml.jackson.databind,com.fasterxml.jackson.databind.annotation,io.fabric8.kubernetes.api.builder,io.fabric8.kubernetes.api.model,io.fabric8.kubernetes.model.annotation";version="4.13.3",io.fabric8.kubernetes.api.model.storage.v1beta1;uses:="com.fasterxml.jackson.annotation,com.fasterxml.jackson.databind,com.fasterxml.jackson.databind.annotation,io.fabric8.kubernetes.api.builder,io.fabric8.kubernetes.api.model,io.fabric8.kubernetes.model.annotation";version="4.13.3"

Implementation-Title: Fabric8 :: Kubernetes Model :: Storage Class

Implementation-Vendor: Red Hat

Implementation-Version: 4.13.3

Import-Package: com.fasterxml.jackson.annotation;version="[2.11,3)",com.fasterxml.jackson.databind;version="[2.11,3)",com.fasterxml.jackson.databind.annotation;version="[2.11,3)",io.fabric8.kubernetes.api.builder;version="[4.13,5)",io.fabric8.kubernetes.api.model,io.fabric8.kubernetes.model.annotation;version="[4.13,5)"

Require-Capability: osgi.ee;filter="(&(osgi.ee=JavaSE)(version=1.8))"

Specification-Title: Fabric8 :: Kubernetes Model :: Storage Class

Specification-Vendor: Red Hat

Specification-Version: 4.13

Tool: Bnd-5.1.1.202006162103

Found in path(s):

* /opt/cola/permits/1288519903_1647861506.61/0/kubernetes-model-storageclass-4-13-3-jar/META-INF/MANIFEST.MF

1.89 kafka-streams-avro-serde 5.5.5

1.89.1 Available under license :

No license file was found, but licenses were detected in source scan.

<name>Apache License 2.0</name>

<url><http://www.apache.org/licenses/LICENSE-2.0.html></url>

Found in path(s):

* /opt/cola/permits/1473459798_1668563265.3256707/0/kafka-streams-avro-serde-5-5-5-jar/META-INF/maven/io.confluent/kafka-streams-avro-serde/pom.xml

1.90 netty-transport-native-unix-common

4.1.84.Final

1.90.1 Available under license :

No license file was found, but licenses were detected in source scan.

<!--

~ Copyright 2016 The Netty Project

~

~ The Netty Project licenses this file to you under the Apache License,

~ version 2.0 (the "License"); you may not use this file except in compliance

~ with the License. You may obtain a copy of the License at:

~

~ <https://www.apache.org/licenses/LICENSE-2.0>

~

~ Unless required by applicable law or agreed to in writing, software

~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT

~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the

~ License for the specific language governing permissions and limitations

~ under the License.

-->

Found in path(s):

* /opt/cola/permits/1470278818_1668107860.126395/0/netty-transport-native-unix-common-4-1-84-final-sources-2-jar/META-INF/maven/io.netty/netty-transport-native-unix-common/pom.xml

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2020 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1470278818_1668107860.126395/0/netty-transport-native-unix-common-4-1-84-final-sources-2-jar/netty_unix.h
* /opt/cola/permits/1470278818_1668107860.126395/0/netty-transport-native-unix-common-4-1-84-final-sources-2-jar/netty_unix.c
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2018 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1470278818_1668107860.126395/0/netty-transport-native-unix-common-4-1-84-final-sources-2-jar/netty_unix_buffer.h
* /opt/cola/permits/1470278818_1668107860.126395/0/netty-transport-native-unix-common-4-1-84-final-sources-2-jar/netty_unix_buffer.c
* /opt/cola/permits/1470278818_1668107860.126395/0/netty-transport-native-unix-common-4-1-84-final-sources-2-jar/io/netty/channel/unix/PreferredDirectByteBufferAllocator.java
```

* /opt/cola/permits/1470278818_1668107860.126395/0/netty-transport-native-unix-common-4-1-84-final-sources-2-jar/io/netty/channel/unix/Buffer.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2021 The Netty Project

*

* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:

*

* <https://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.

*/

Found in path(s):

* /opt/cola/permits/1470278818_1668107860.126395/0/netty-transport-native-unix-common-4-1-84-final-sources-2-jar/io/netty/channel/unix/DomainDatagramChannelConfig.java

* /opt/cola/permits/1470278818_1668107860.126395/0/netty-transport-native-unix-common-4-1-84-final-sources-2-jar/io/netty/channel/unix/SegmentedDatagramPacket.java

* /opt/cola/permits/1470278818_1668107860.126395/0/netty-transport-native-unix-common-4-1-84-final-sources-2-jar/io/netty/channel/unix/DomainDatagramPacket.java

* /opt/cola/permits/1470278818_1668107860.126395/0/netty-transport-native-unix-common-4-1-84-final-sources-2-jar/io/netty/channel/unix/DomainDatagramChannel.java

* /opt/cola/permits/1470278818_1668107860.126395/0/netty-transport-native-unix-common-4-1-84-final-sources-2-jar/io/netty/channel/unix/DomainDatagramSocketAddress.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2015 The Netty Project

*

* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:

*

* <https://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.

*/

Found in path(s):

- * /opt/cola/permits/1470278818_1668107860.126395/0/netty-transport-native-unix-common-4-1-84-final-sources-2-jar/io/netty/channel/unix/Socket.java
- * /opt/cola/permits/1470278818_1668107860.126395/0/netty-transport-native-unix-common-4-1-84-final-sources-2-jar/io/netty/channel/unix/FileDescriptor.java
- * /opt/cola/permits/1470278818_1668107860.126395/0/netty-transport-native-unix-common-4-1-84-final-sources-2-jar/netty_unix_filedescriptor.c
- * /opt/cola/permits/1470278818_1668107860.126395/0/netty-transport-native-unix-common-4-1-84-final-sources-2-jar/netty_unix_socket.c
- * /opt/cola/permits/1470278818_1668107860.126395/0/netty-transport-native-unix-common-4-1-84-final-sources-2-jar/netty_unix_errors.c
- * /opt/cola/permits/1470278818_1668107860.126395/0/netty-transport-native-unix-common-4-1-84-final-sources-2-jar/io/netty/channel/unix/InetAddress.java
- * /opt/cola/permits/1470278818_1668107860.126395/0/netty-transport-native-unix-common-4-1-84-final-sources-2-jar/io/netty/channel/unix/UnixChannel.java
- * /opt/cola/permits/1470278818_1668107860.126395/0/netty-transport-native-unix-common-4-1-84-final-sources-2-jar/io/netty/channel/unix/DatagramSocketAddress.java
- * /opt/cola/permits/1470278818_1668107860.126395/0/netty-transport-native-unix-common-4-1-84-final-sources-2-jar/io/netty/channel/unix/ServerDomainSocketChannel.java
- * /opt/cola/permits/1470278818_1668107860.126395/0/netty-transport-native-unix-common-4-1-84-final-sources-2-jar/netty_unix_errors.h
- * /opt/cola/permits/1470278818_1668107860.126395/0/netty-transport-native-unix-common-4-1-84-final-sources-2-jar/netty_unix_socket.h
- * /opt/cola/permits/1470278818_1668107860.126395/0/netty-transport-native-unix-common-4-1-84-final-sources-2-jar/io/netty/channel/unix/DomainSocketChannel.java
- * /opt/cola/permits/1470278818_1668107860.126395/0/netty-transport-native-unix-common-4-1-84-final-sources-2-jar/io/netty/channel/unix/Errors.java
- * /opt/cola/permits/1470278818_1668107860.126395/0/netty-transport-native-unix-common-4-1-84-final-sources-2-jar/netty_unix_filedescriptor.h
- * /opt/cola/permits/1470278818_1668107860.126395/0/netty-transport-native-unix-common-4-1-84-final-sources-2-jar/io/netty/channel/unix/DomainSocketReadMode.java
- * /opt/cola/permits/1470278818_1668107860.126395/0/netty-transport-native-unix-common-4-1-84-final-sources-2-jar/io/netty/channel/unix/DomainSocketAddress.java
- * /opt/cola/permits/1470278818_1668107860.126395/0/netty-transport-native-unix-common-4-1-84-final-sources-2-jar/io/netty/channel/unix/DomainSocketChannelConfig.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2022 The Netty Project

*

* The Netty Project licenses this file to you under the Apache License,

* version 2.0 (the "License"); you may not use this file except in compliance

* with the License. You may obtain a copy of the License at:

*

* <https://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):

* /opt/cola/permits/1470278818_1668107860.126395/0/netty-transport-native-unix-common-4-1-84-final-sources-2-jar/io/netty/channel/unix/GenericUnixChannelOption.java
* /opt/cola/permits/1470278818_1668107860.126395/0/netty-transport-native-unix-common-4-1-84-final-sources-2-jar/io/netty/channel/unix/IntegerUnixChannelOption.java
* /opt/cola/permits/1470278818_1668107860.126395/0/netty-transport-native-unix-common-4-1-84-final-sources-2-jar/io/netty/channel/unix/RawUnixChannelOption.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2014 The Netty Project

*

* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:

*

* <https://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.

*/

Found in path(s):

* /opt/cola/permits/1470278818_1668107860.126395/0/netty-transport-native-unix-common-4-1-84-final-sources-2-jar/io/netty/channel/unix/UnixChannelOption.java
* /opt/cola/permits/1470278818_1668107860.126395/0/netty-transport-native-unix-common-4-1-84-final-sources-2-jar/io/netty/channel/unix/Unix.java
* /opt/cola/permits/1470278818_1668107860.126395/0/netty-transport-native-unix-common-4-1-84-final-sources-2-jar/io/netty/channel/unix/package-info.java
* /opt/cola/permits/1470278818_1668107860.126395/0/netty-transport-native-unix-common-4-1-84-final-sources-2-jar/io/netty/channel/unix/IovArray.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2016 The Netty Project

*

* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:

*
* <https://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):

* /opt/cola/permits/1470278818_1668107860.126395/0/netty-transport-native-unix-common-4-1-84-final-sources-2-jar/io/netty/channel/unix/LimitsStaticallyReferencedJniMethods.java
* /opt/cola/permits/1470278818_1668107860.126395/0/netty-transport-native-unix-common-4-1-84-final-sources-2-jar/netty_unix_util.c
* /opt/cola/permits/1470278818_1668107860.126395/0/netty-transport-native-unix-common-4-1-84-final-sources-2-jar/io/netty/channel/unix/PeerCredentials.java
* /opt/cola/permits/1470278818_1668107860.126395/0/netty-transport-native-unix-common-4-1-84-final-sources-2-jar/io/netty/channel/unix/ErrorsStaticallyReferencedJniMethods.java
* /opt/cola/permits/1470278818_1668107860.126395/0/netty-transport-native-unix-common-4-1-84-final-sources-2-jar/netty_unix_util.h
* /opt/cola/permits/1470278818_1668107860.126395/0/netty-transport-native-unix-common-4-1-84-final-sources-2-jar/netty_unix_limits.c
* /opt/cola/permits/1470278818_1668107860.126395/0/netty-transport-native-unix-common-4-1-84-final-sources-2-jar/io/netty/channel/unix/Limits.java
* /opt/cola/permits/1470278818_1668107860.126395/0/netty-transport-native-unix-common-4-1-84-final-sources-2-jar/netty_unix_limits.h
* /opt/cola/permits/1470278818_1668107860.126395/0/netty-transport-native-unix-common-4-1-84-final-sources-2-jar/io/netty/channel/unix/SocketWritableByteChannel.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2017 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
* <https://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):

```
* /opt/cola/permits/1470278818_1668107860.126395/0/netty-transport-native-unix-common-4-1-84-final-sources-2-jar/io/netty/channel/unix/UnixChannelUtil.java
* /opt/cola/permits/1470278818_1668107860.126395/0/netty-transport-native-unix-common-4-1-84-final-sources-2-jar/netty_unix_jni.h
```

1.91 testng 7.0.0

1.91.1 Available under license :

Apache-2.0

1.92 asm-tree 7.2

1.92.1 Available under license :

<OWNER> = Regents of the University of California

<ORGANIZATION> = University of California, Berkeley

<YEAR> = 1998

In the original BSD license, both occurrences of the phrase "COPYRIGHT HOLDERS AND CONTRIBUTORS" in the disclaimer read "REGENTS AND CONTRIBUTORS".

Here is the license template:

Copyright (c) <YEAR>, <OWNER>

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the <ORGANIZATION> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.93 apache-kafka 3.3.1

1.93.1 Available under license :

Apache Kafka

Copyright 2022 The Apache Software Foundation.

This product includes software developed at
The Apache Software Foundation (<https://www.apache.org/>).

This distribution has a binary dependency on jersey, which is available under the CDDL License. The source code of jersey can be found at <https://github.com/jersey/jersey/>.

This distribution has a binary test dependency on jqwik, which is available under the Eclipse Public License 2.0. The source code can be found at <https://github.com/jlink/jqwik>.

The streams-scala (streams/streams-scala) module was donated by Lightbend and the original code was copyrighted by them:

Copyright (C) 2018 Lightbend Inc. <<https://www.lightbend.com>>

Copyright (C) 2017-2018 Alexis Seigneurin.

This project contains the following code copied from Apache Hadoop:

clients/src/main/java/org/apache/kafka/common/utils/PureJavaCrc32C.java

Some portions of this file Copyright (c) 2004-2006 Intel Corporation and licensed under the BSD license.

This project contains the following code copied from Apache Hive:

streams/src/main/java/org/apache/kafka/streams/state/internals/Murmur3.java

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,

and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the

Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,

whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.94 guava 31.0.1-jre

1.94.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2020 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except

* in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License

* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express

* or implied. See the License for the specific language governing permissions and limitations under

* the License.

*/

/**

* Holder for web specializations of methods of { @code Floats }. Intended to be empty for regular

* version.

*/

Found in path(s):

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/primitives/FloatsMethodsForWeb.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2007 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except

* in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License

* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express

* or implied. See the License for the specific language governing permissions and limitations under

* the License.

*/

/*

* This following method is a modified version of one found in

* <http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/src/test/tck/AbstractExecutorServiceTest.java?revision=1.30>

* which contained the following notice:
*
* Written by Doug Lea with assistance from members of JCP JSR-166 Expert Group and released to
* the public domain, as explained at <http://creativecommons.org/publicdomain/zero/1.0/>
*
* Other contributors include Andrew Wright, Jeffrey Hayes, Pat Fisher, Mike Judd.
*/

Found in path(s):

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/util/concurrent/MoreExecutors.java

No license file was found, but licenses were detected in source scan.

/*

* Written by Doug Lea with assistance from members of JCP JSR-166
* Expert Group and released to the public domain, as explained at
* <http://creativecommons.org/publicdomain/zero/1.0/>
*/

Found in path(s):

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/cache/LongAdder.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/util/concurrent/AtomicDoubleArray.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/hash/Striped64.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/cache/Striped64.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/hash/LongAdder.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2020 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express

* or implied. See the License for the specific language governing permissions and limitations under
* the License.

*/

/**

* Holder for web specializations of methods of { @code Doubles }. Intended to be empty for regular

* version.

*/

Found in path(s):

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/primitives/DoublesMethodsForWeb.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2011 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/util/concurrent/AtomicLongMap.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/GwtTransient.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2013 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/MultimapBuilder.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/ImmutableMapEntry.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/io/MoreFiles.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2014 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express

* or implied. See the License for the specific language governing permissions and limitations under
* the License.

*/

Found in path(s):

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/eventbus/SubscriberRegistry.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/base/MoreObjects.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/eventbus/Subscriber.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/util/concurrent/ListenerCallQueue.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/eventbus/Dispatcher.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/util/concurrent/TrustedListenableFutureTask.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/math/Quantiles.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2007 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/base/Functions.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/base/Objects.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/base/FinalizableSoftReference.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/eventbus/AllowConcurrentEvents.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/util/concurrent/AbstractFuture.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/eventbus/DeadEvent.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/util/concurrent/package-info.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/util/concurrent/ExecutionList.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/io/MultiInputStream.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/HashBiMap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/io/LittleEndianDataInputStream.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/util/concurrent/ListenableFuture.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/base/Charsets.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/eventbus/EventBus.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/base/Defaults.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/eventbus/package-info.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/base/Predicates.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/io/Closeables.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/eventbus/Subscribe.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/io/CharStreams.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/eventbus/AsyncEventBus.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/base/Supplier.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/io/LineBuffer.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/base/Preconditions.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/io/Resources.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/io/LineReader.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/io/LittleEndianDataOutputStream.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/base/Throwables.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/base/Predicate.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/io/ByteStreams.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/base/FinalizableReferenceQueue.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/io/Files.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/base/FinalizableWeakReference.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/base/AbstractIterator.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/primitives/Primitives.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/base/FinalizableReference.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/io/CountingOutputStream.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/base/FinalizablePhantomReference.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/EnumMultiset.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/io/package-info.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/io/CountingInputStream.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/io/Flushables.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/base/package-info.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/util/concurrent/DirectExecutor.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-

jar/com/google/common/collect/Interners.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/base/Suppliers.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/base/Function.java
No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2012 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-  
jar/com/google/common/io/ByteSource.java  
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-  
jar/com/google/common/hash/AbstractByteHasher.java  
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-  
jar/com/google/common/reflect/AbstractInvocationHandler.java  
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-  
jar/com/google/common/reflect/Parameter.java  
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-  
jar/com/google/common/util/concurrent/ServiceManager.java  
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-  
jar/com/google/common/base/StandardSystemProperty.java  
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-  
jar/com/google/common/math/StatsAccumulator.java  
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-  
jar/com/google/common/collect/CartesianList.java  
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-  
jar/com/google/common/cache/LongAddable.java  
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-  
jar/com/google/common/reflect/TypeToInstanceMap.java  
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-  
jar/com/google/common/html/package-info.java  
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-  
jar/com/google/common/hash/ChecksumHashFunction.java  
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
```

jar/com/google/common/io/ByteSink.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/reflect/TypeCapture.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/math/LinearTransformation.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/FilteredKeyMultimap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/reflect/ClassPath.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/hash/LongAddables.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/xml/package-
info.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/io/BaseEncoding.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/escape/package-info.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/util/concurrent/SmoothRateLimiter.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/hash/SipHashFunction.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/reflect/MutableTypeToInstanceMap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/io/CharSource.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/reflect/package-info.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/math/Stats.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/ImmutableRangeMap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/math/PairedStatsAccumulator.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/io/FileWriteMode.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/ImmutableRangeSet.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/io/CharSink.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/hash/LongAddable.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/cache/LongAddables.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/util/concurrent/RateLimiter.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/math/PairedStats.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-

jar/com/google/common/reflect/Invokable.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/reflect/ImmutableTypeToInstanceMap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/io/Closer.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/util/concurrent/ListenableScheduledFuture.java
No license file was found, but licenses were detected in source scan.

```
/*  
* Copyright (C) 2010 The Guava Authors  
*  
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except  
* in compliance with the License. You may obtain a copy of the License at  
*  
* http://www.apache.org/licenses/LICENSE-2.0  
*  
* Unless required by applicable law or agreed to in writing, software distributed under the License  
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either  
* express  
* or implied. See the License for the specific language governing permissions and limitations under  
* the License.  
*/
```

Found in path(s):

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/primitives/package-info.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/annotations/Beta.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/util/concurrent/ListeningExecutorService.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/util/concurrent/ThreadFactoryBuilder.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/base/Ascii.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/base/Equivalence.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/base/Strings.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/util/concurrent/UncaughtExceptionHandler.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/util/concurrent/Atomics.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/ContiguousSet.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/SortedLists.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/net/package-
info.java

```
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/util/concurrent/ForwardingBlockingQueue.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/util/concurrent/Monitor.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/annotations/package-info.java
No license file was found, but licenses were detected in source scan.
```

```
/*
* Copyright (C) 2007 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):

```
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/Multiset.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/Iterables.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/SortedSetMultimap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/MapDifference.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/Sets.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/HashMultiset.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/TreeMultimap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/AbstractIterator.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/package-info.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/MutableClassToInstanceMap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/SetMultimap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
```

jar/com/google/common/collect/Multisets.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/BiMap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/ForwardingSortedMap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/ReverseOrdering.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/AbstractMapEntry.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/ByFunctionOrdering.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/Ordering.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/ImmutableSet.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/ForwardingMapEntry.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/EnumHashBiMap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/Interner.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/ForwardingObject.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/Multimap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/ForwardingListIterator.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/LinkedListMultimap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/ExplicitOrdering.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/ForwardingMultiset.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/ComparatorOrdering.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/ImmutableList.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/AbstractMapBasedMultimap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/AbstractMultiset.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/ForwardingSortedSet.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/Lists.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/RegularImmutableSet.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-

jar/com/google/common/collect/AbstractBiMap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/ForwardingQueue.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/EnumBiMap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/ForwardingMultimap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/LinkedHashMultimap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/Iterators.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/NaturalOrdering.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/Multimaps.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/ForwardingMap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/NullsFirstOrdering.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/NullsLastOrdering.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/UsingToStringOrdering.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/Maps.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/ClassToInstanceMap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/ListMultimap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/ConcurrentHashMultiset.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/ArrayListMultimap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/ForwardingList.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/TreeMultiset.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/ForwardingSet.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/ForwardingConcurrentMap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/AbstractListMultimap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/AbstractMapBasedMultiset.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/ForwardingCollection.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-

jar/com/google/common/collect/SingletonImmutableSet.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/AbstractSortedSetMultimap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/HashMultimap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/LexicographicalOrdering.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/CompoundOrdering.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/AbstractSetMultimap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/ForwardingIterator.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/Synchronized.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/ReverseNaturalOrdering.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/LinkedHashMultiset.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2017 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express

* or implied. See the License for the specific language governing permissions and limitations under
* the License.

*/

Found in path(s):

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/primitives/ImmutableIntArray.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/util/concurrent/ForwardingCondition.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/util/concurrent/ForwardingLock.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/primitives/ImmutableDoubleArray.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/hash/AbstractHashFunction.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-

jar/com/google/common/primitives/ImmutableLongArray.java
No license file was found, but licenses were detected in source scan.

```
/*  
 * Copyright (C) 2016 The Guava Authors  
 *  
 * Licensed under the Apache License, Version 2.0 (the "License");  
 * you may not use this file except in compliance with the License.  
 * You may obtain a copy of the License at  
 *  
 * http://www.apache.org/licenses/LICENSE-2.0  
 *  
 * Unless required by applicable law or agreed to in writing, software  
 * distributed under the License is distributed on an "AS IS" BASIS,  
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
 * See the License for the specific language governing permissions and  
 * limitations under the License.  
 */
```

Found in path(s):

```
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-  
jar/com/google/common/collect/LinkedHashMapMultimapGwtSerializationDependencies.java  
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-  
jar/com/google/common/graph/DirectedNetworkConnections.java  
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-  
jar/com/google/common/graph/ElementOrder.java  
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-  
jar/com/google/common/graph/DirectedMultiNetworkConnections.java  
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-  
jar/com/google/common/graph/ImmutableValueGraph.java  
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-  
jar/com/google/common/collect/CollectCollectors.java  
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-  
jar/com/google/common/graph/MutableValueGraph.java  
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-  
jar/com/google/common/graph/AbstractGraphBuilder.java  
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-  
jar/com/google/common/graph/StandardMutableGraph.java  
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-  
jar/com/google/common/collect/ImmutableMultisetGwtSerializationDependencies.java  
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-  
jar/com/google/common/collect/ArrayListMultimapGwtSerializationDependencies.java  
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-  
jar/com/google/common/graph/GraphBuilder.java  
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-  
jar/com/google/common/graph/StandardMutableValueGraph.java  
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-  
jar/com/google/common/collect/Comparators.java
```

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/graph/EndpointPairIterator.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/graph/ValueGraphBuilder.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/graph/MapRetrievalCache.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/graph/ForwardingGraph.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/graph/AbstractValueGraph.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/graph/ValueGraph.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/MoreCollectors.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/graph/UndirectedMultiNetworkConnections.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/graph/GraphConstants.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/graph/ForwardingValueGraph.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/graph/StandardMutableNetwork.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/graph/StandardNetwork.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/RangeGwtSerializationDependencies.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/graph/StandardValueGraph.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/graph/MapIteratorCache.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/graph/AbstractUndirectedNetworkConnections.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/graph/NetworkConnections.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/HashMultimapGwtSerializationDependencies.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/graph/AbstractGraph.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/graph/MultiEdgesConnecting.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/graph/AbstractNetwork.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/graph/UndirectedNetworkConnections.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/graph/NetworkBuilder.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/graph/ForwardingNetwork.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/graph/EdgesConnecting.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/graph/AbstractDirectedNetworkConnections.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/graph/GraphConnections.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/graph/DirectedGraphConnections.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/graph/UndirectedGraphConnections.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/graph/EndpointPair.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2019 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express

* or implied. See the License for the specific language governing permissions and limitations under
* the License.

*/

Found in path(s):

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/util/concurrent/Internal.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/primitives/Platform.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2007 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

```
* See the License for the specific language governing permissions and
* limitations under the License.
*/
/**
 * Returns an array containing all of the elements in the specified collection. This method
 * returns the elements in the order they are returned by the collection's iterator. The returned
 * array is "safe" in that no references to it are maintained by the collection. The caller is
 * thus free to modify the returned array.
 *
 * <p>This method assumes that the collection size doesn't change while the method is running.
 *
 * <p>TODO(kevinb): support concurrently modified collections?
 *
 * @param c the collection for which to return an array of elements
 */
```

Found in path(s):

```
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/ObjectArrays.java
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2012 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */
/**
 * This method was rewritten in Java from an intermediate step of the Murmur hash function in
 * http://code.google.com/p/smhasher/source/browse/trunk/MurmurHash3.cpp, which contained the
 * following header:
 *
 * MurmurHash3 was written by Austin Appleby, and is placed in the public domain. The author
 * hereby disclaims copyright to this source code.
 */
```

Found in path(s):

```
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/base/SmallCharMatcher.java
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2015 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you
 * may not use this file except in compliance with the License. You may
 * obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
 * implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/Streams.java
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2012 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/CompactHashSet.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/FilteredMultimap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/DescendingImmutableSortedSet.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/AbstractNavigableMap.java
```

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/FilteredSetMultimap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/FilteredKeySetMultimap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/ForwardingImmutableList.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/ForwardingImmutableSet.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/CompactLinkedHashSet.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/UnmodifiableSortedMultiset.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/EvictingQueue.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/CompactHashMap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/TreeTraverser.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/DescendingMultiset.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/ImmutableEnumMap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/FilteredEntryMultimap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/RegularImmutableAsList.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/ForwardingNavigableMap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/TransformedListIterator.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/FilteredKeyListMultimap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/TreeRangeMap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/CompactLinkedHashMap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/ForwardingBlockingDeque.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/AbstractMultimap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/FilteredEntrySetMultimap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/RangeMap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/ForwardingDeque.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/AbstractSortedKeySortedSetMultimap.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/AllEqualOrdering.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/ForwardingNavigableSet.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/util/concurrent/ForwardingBlockingDeque.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/TransformedIterator.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/SortedMultisetBridge.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/ForwardingImmutableMap.java
No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2013 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express

* or implied. See the License for the specific language governing permissions and limitations under
* the License.

*/

Found in path(s):

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/reflect/TypeVisitor.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/eventbus/SubscriberExceptionHandler.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/AbstractTable.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/base/VerifyException.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/base/Verify.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/thirdparty/publicsuffix/PublicSuffixType.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/FilteredMultimapValues.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/eventbus/SubscriberExceptionContext.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/io/CharSequenceReader.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/util/concurrent/WrappingScheduledExecutorService.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/hash/HashingInputStream.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/base/Utf8.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/util/concurrent/Runnables.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2011 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the
* License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND,
either

* express or implied. See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/GeneralRange.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/RegularImmutableSortedMultiset.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/ImmutableSortedMultisetFauxverideShim.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/AbstractRangeSet.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/SortedIterable.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/Count.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/ImmutableSortedMultiset.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/SortedIterables.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/ForwardingSortedMultiset.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/RangeSet.java

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2015 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/util/concurrent/Platform.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/io/ReaderInputStream.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/util/concurrent/InterruptibleTask.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/util/concurrent/AsyncCallable.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/hash/LittleEndianByteArray.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/ConsumingQueueIterator.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/hash/MacHashFunction.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/util/concurrent/CombinedFuture.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/util/concurrent/AggregateFutureState.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/hash/FarmHashFingerprint64.java
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2018 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
```

express

* or implied. See the License for the specific language governing permissions and limitations under

* the License.

*/

Found in path(s):

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/hash/ImmutableSupplier.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/JdkBackedImmutableMultiset.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/util/concurrent/ExecutionSequencer.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/JdkBackedImmutableSet.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2011 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except

* in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License

* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either

express

* or implied. See the License for the specific language governing permissions and limitations under

* the License.

*/

Found in path(s):

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/hash/AbstractStreamingHasher.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/cache/Cache.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/cache/AbstractLoadingCache.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/cache/CacheBuilderSpec.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/hash/BloomFilter.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/primitives/UnsignedLong.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/cache/ForwardingLoadingCache.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/DescendingImmutableSortedMultiset.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/net/HostAndPort.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/hash/PrimitiveSink.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/math/IntMath.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/hash/Hasher.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/base/PairwiseEquivalence.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/cache/LoadingCache.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/primitives/ParseRequest.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/hash/Hashing.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/math/DoubleUtils.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/cache/Weigher.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/primitives/UnsignedInteger.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/util/concurrent/AbstractScheduledService.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/util/concurrent/AbstractListeningExecutorService.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/primitives/UnsignedInts.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/util/concurrent/ExecutionError.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/hash/Funnel.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/reflect/TypeParameter.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/hash/BloomFilterStrategies.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/util/concurrent/CycleDetectingLockFactory.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/net/MediaType.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/util/concurrent/Uninterruptibles.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/net/HttpHeaders.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/cache/RemovalCause.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/cache/package-info.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/math/MathPreconditions.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/base/Ticker.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/base/Absent.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/util/concurrent/ForwardingExecutorService.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/EmptyContiguousSet.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/math/DoubleMath.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/cache/RemovalListener.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/AbstractSortedMultiset.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/base/Enums.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/hash/AbstractCompositeHashFunction.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/base/Present.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/hash/Funnels.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/primitives/UnsignedLongs.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/hash/HashingOutputStream.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/base/Optional.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/math/BigIntegerMath.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/hash/AbstractHasher.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/BoundType.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/TreeRangeSet.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/cache/RemovalListeners.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/cache/ForwardingCache.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/util/concurrent/FutureCallback.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/hash/HashCode.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/math/package-info.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/hash/Murmur3_128HashFunction.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/math/LongMath.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/hash/AbstractNonStreamingHashFunction.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/hash/MessageDigestHashFunction.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/hash/Murmur3_32HashFunction.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/Queues.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/cache/AbstractCache.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/util/concurrent/ForwardingListeningExecutorService.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/RegularImmutableMultiset.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/hash/Crc32cHashFunction.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/util/concurrent/WrappingExecutorService.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/util/concurrent/UncheckedExecutionException.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/hash/HashFunction.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/cache/CacheStats.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/cache/RemovalNotification.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/util/concurrent/AsyncFunction.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/RegularContiguousSet.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/util/concurrent/ListeningScheduledExecutorService.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/base/FunctionalEquivalence.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/reflect/Types.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/cache/CacheLoader.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2011 The Guava Authors.

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except

* in compliance with the License. You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/hash/package-info.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2014 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/TopKSelector.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/graph/Graphs.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/graph/PredecessorsFunction.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/graph/ImmutableNetwork.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/graph/Graph.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/graph/MutableGraph.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/graph/Network.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-

jar/com/google/common/graph/MutableNetwork.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/graph/SuccessorsFunction.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/io/InsecureRecursiveDeleteException.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/graph/ImmutableGraph.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/io/RecursiveDeleteOption.java
No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2009 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
/**
 * Not supported. <b>You are attempting to create a map that may contain a non-{@code Comparable}
 * key.</b> Proper calls will resolve to the version in {@code ImmutableSortedMap}, not this dummy
 * version.
 *
 * @throws UnsupportedOperationException always
 * @deprecated <b>Pass a key of type {@code Comparable} to use {@link
 *     ImmutableSortedMap#of(Comparable, Object)}.</b>
 */
```

Found in path(s):
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/ImmutableSortedMapFauxverideShim.java
No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2016 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
```

*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/base/JdkPattern.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/base/CommonPattern.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/base/CommonMatcher.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/base/PatternCompiler.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2020 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.

*/

/**

* Holder for web specializations of methods of { @code Ints }. Intended to be empty for regular
* version.

*/

Found in path(s):

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/primitives/IntsMethodsForWeb.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2006 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at

*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/util/concurrent/FuturesGetChecked.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/util/concurrent/UncheckedTimeoutException.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/util/concurrent/TimeLimiter.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/util/concurrent/FluentFuture.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/util/concurrent/ImmediateFuture.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/escape/CharEscaper.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/reflect/TypeToken.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/io/PatternFilenameFilter.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/annotations/VisibleForTesting.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/util/concurrent/SimpleTimeLimiter.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/util/concurrent/FakeTimeLimiter.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/io/AppendableWriter.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/util/concurrent/Futures.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/util/concurrent/AbstractCatchingFuture.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/util/concurrent/TimeoutFuture.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/util/concurrent/GwtFuturesCatchingSpecialization.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/escape/CharEscaperBuilder.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/util/concurrent/AbstractTransformFuture.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-

jar/com/google/common/util/concurrent/CollectionFuture.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/base/CaseFormat.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/util/concurrent/AggregateFuture.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/util/concurrent/GwtFluentFutureCatchingSpecialization.java
No license file was found, but licenses were detected in source scan.

```
/*  
* Copyright (C) 2009 The Guava Authors  
*  
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except  
* in compliance with the License. You may obtain a copy of the License at  
*  
* http://www.apache.org/licenses/LICENSE-2.0  
*  
* Unless required by applicable law or agreed to in writing, software distributed under the  
* License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND,  
either  
* express or implied. See the License for the specific language governing permissions and  
* limitations under the License.  
*/
```

Found in path(s):
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/ImmutableSortedAsList.java
No license file was found, but licenses were detected in source scan.

```
/*  
* Copyright (C) 2005 The Guava Authors  
*  
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except  
* in compliance with the License. You may obtain a copy of the License at  
*  
* http://www.apache.org/licenses/LICENSE-2.0  
*  
* Unless required by applicable law or agreed to in writing, software distributed under the License  
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either  
express  
* or implied. See the License for the specific language governing permissions and limitations under  
* the License.  
*/
```

Found in path(s):
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/reflect/Reflection.java

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2009 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/util/concurrent/ForwardingListenableFuture.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/util/concurrent/Service.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/cache/CacheBuilder.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/util/concurrent/ForwardingFuture.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/MapMakerInternalMap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/annotations/GwtCompatible.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/annotations/GwtIncompatible.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/cache/LocalCache.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/util/concurrent/AbstractExecutionThreadService.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/util/concurrent/AbstractIdleService.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/cache/ReferenceEntry.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/net/InternetDomainName.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/html/HtmlEscapers.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/util/concurrent/AbstractService.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/Cut.java
```

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/base/Platform.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/util/concurrent/ForwardingFluentFuture.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/escape/Escapers.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/primitives/SignedBytes.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/primitives/UnsignedBytes.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/net/HostSpecifier.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/escape/ArrayBasedCharEscaper.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/base/Splitter.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/escape/Platform.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/MapMaker.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/DenseImmutableTable.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/net/UrlEscapers.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/io/ByteArrayDataOutput.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/RegularImmutableTable.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/util/concurrent/Callables.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/escape/ArrayBasedEscaperMap.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/reflect/TypeResolver.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/io/ByteArrayDataInput.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/escape/ArrayBasedUnicodeEscaper.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/SparseImmutableTable.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/xml/XmlEscapers.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/io/LineProcessor.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/util/concurrent/JdkFutureAdapters.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/util/concurrent/SettableFuture.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/io/ByteProcessor.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2015 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/graph/package-info.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/ImmutableBiMapFauxverideShim.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/CollectSpliterators.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2018 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/JdkBackedImmutableBiMap.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/IndexedImmutableSet.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/BaseImmutableMultimap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/JdkBackedImmutableMap.java
No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2010 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/ForwardingSetMultimap.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/MinMaxPriorityQueue.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/ForwardingImmutableCollection.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/AbstractSequentialIterator.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/ForwardingListMultimap.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/UnmodifiableListIterator.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/ForwardingSortedSetMultimap.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/RowSortedTable.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/SortedMapDifference.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2009 The Guava Authors

*

- * Licensed under the Apache License, Version 2.0 (the "License");
- * you may not use this file except in compliance with the License.
- * You may obtain a copy of the License at
- *
- * <http://www.apache.org/licenses/LICENSE-2.0>
- *
- * Unless required by applicable law or agreed to in writing, software
- * distributed under the License is distributed on an "AS IS" BASIS,
- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.
- */

Found in path(s):

- * /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/ImmutableClassToInstanceMap.java
- * /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/DiscreteDomain.java
- * /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/ImmutableSetMultimap.java
- * /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/RegularImmutableSortedSet.java
- * /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/ImmutableSortedSetFauxverideShim.java
- * /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/ForwardingTable.java
- * /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/ImmutableEnumSet.java
- * /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/ImmutableAsList.java
- * /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/AbstractIndexedListAdapter.java
- * /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/EmptyImmutableSetMultimap.java
- * /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/SingletonImmutableTable.java
- * /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/TableCollectors.java
- * /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/RegularImmutableList.java
- * /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/ComparisonChain.java
- * /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/ImmutableSortedMap.java
- * /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/ImmutableTable.java
- * /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/ArrayTable.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/ComputationException.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/SingletonImmutableList.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2021 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/escape/ParametricNullness.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/io/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/base/ParametricNullness.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/net/ParametricNullness.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/xml/ParametricNullness.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/util/concurrent/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/io/ParametricNullness.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/escape/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/eventbus/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/reflect/ParametricNullness.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/primitives/ParametricNullness.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/graph/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-

jar/com/google/common/primitives/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/xml/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/hash/ParametricNullness.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/reflect/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/eventbus/ParametricNullness.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/math/ParametricNullness.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/net/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/cache/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/math/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/base/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/cache/ParametricNullness.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/html/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/ParametricNullness.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/util/concurrent/ParametricNullness.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/graph/ParametricNullness.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/hash/ElementTypesAreNonnullByDefault.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/html/ParametricNullness.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2020 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License

* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express

* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/**
* Holder for web specializations of methods of { @code Shorts }. Intended to be empty for regular
* version.
*/

Found in path(s):

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/primitives/ShortsMethodsForWeb.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2011 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/*
* This method was written by Doug Lea with assistance from members of JCP JSR-166 Expert Group
* and released to the public domain, as explained at
* <http://creativecommons.org/licenses/publicdomain>
*
* As of 2010/06/11, this method is identical to the (package private) hash method in OpenJDK 7's
* java.util.HashMap class.
*/

Found in path(s):

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/util/concurrent/Striped.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2017 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*
*/

* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/graph/AbstractBaseGraph.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/util/concurrent/ClosingFuture.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/graph/BaseGraph.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/graph/Traverser.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2016 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.

*/

/**

* Holder for extra methods of {@code Objects} only in web. Intended to be empty for regular
* version.

*/

Found in path(s):

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/base/ExtraObjectsMethodsForWeb.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2020 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except

* in compliance with the License. You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/math/BigDecimalMath.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/util/concurrent/OverflowAvoidingLockSupport.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/hash/Java8Compatibility.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/math/ToDoubleRounder.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/base/Java8Compatibility.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/io/Java8Compatibility.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2009 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express

* or implied. See the License for the specific language governing permissions and limitations under
* the License.

*/

/**

* Outer class that exists solely to let us write { @code Partially.GwtIncompatible } instead of plain
* { @code GwtIncompatible }. This is more accurate for { @link Futures#catching }, which is available
* under GWT but with a slightly different signature.

*

* <p>We can't use { @code PartiallyGwtIncompatible } because then the GWT compiler wouldn't recognize
* it as a { @code GwtIncompatible } annotation. And for { @code Futures.catching }, we need the GWT
* compiler to autostrip the normal server method in order to expose the special, inherited GWT

* version.

*/

Found in path(s):

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/util/concurrent/Partially.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2020 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/util/concurrent/ServiceManagerBridge.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2021 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except

* in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License

* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express

* or implied. See the License for the specific language governing permissions and limitations under

* the License.

*/

Found in path(s):

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/base/NullnessCasts.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-

jar/com/google/common/util/concurrent/NullnessCasts.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/NullnessCasts.java
No license file was found, but licenses were detected in source scan.

```
/*  
* Copyright (C) 2008 The Guava Authors  
*  
* Licensed under the Apache License, Version 2.0 (the "License");  
* you may not use this file except in compliance with the License.  
* You may obtain a copy of the License at  
*  
* http://www.apache.org/licenses/LICENSE-2.0  
*  
* Unless required by applicable law or agreed to in writing, software  
* distributed under the License is distributed on an "AS IS" BASIS,  
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
* See the License for the specific language governing permissions and  
* limitations under the License.  
*/
```

Found in path(s):

```
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-  
jar/com/google/common/collect/Platform.java  
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-  
jar/com/google/common/collect/ImmutableMapEntrySet.java  
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-  
jar/com/google/common/collect/ImmutableMapValues.java  
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-  
jar/com/google/common/collect/Tables.java  
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-  
jar/com/google/common/collect/Collections2.java  
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-  
jar/com/google/common/collect/ImmutableMultiset.java  
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-  
jar/com/google/common/collect/ImmutableMap.java  
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-  
jar/com/google/common/collect/HashBasedTable.java  
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-  
jar/com/google/common/collect/ImmutableMultimap.java  
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-  
jar/com/google/common/collect/CollectPreconditions.java  
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-  
jar/com/google/thirdparty/publicsuffix/PublicSuffixPatterns.java  
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-  
jar/com/google/common/collect/ImmutableMapKeySet.java  
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-  
jar/com/google/common/collect/StandardRowSortedTable.java
```

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/ImmutableListMultimap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/RegularImmutableMap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/PeekingIterator.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/Table.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/ImmutableEntry.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/RegularImmutableBiMap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/SingletonImmutableBiMap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/Serialization.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/ImmutableSortedSet.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/ImmutableBiMap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/StandardTable.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/EmptyImmutableListMultimap.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/ImmutableCollection.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/Range.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/TreeBasedTable.java
* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/UnmodifiableIterator.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2008 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except

* in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License

* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express

* or implied. See the License for the specific language governing permissions and limitations under

* the License.

*/

Found in path(s):

- * /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/base/internal/Finalizer.java
- * /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/primitives/Booleans.java
- * /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/thirdparty/publicsuffix/TrieParser.java
- * /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/primitives/Ints.java
- * /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/util/concurrent/ListenableFutureTask.java
- * /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/primitives/Chars.java
- * /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/primitives/Shorts.java
- * /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/escape/Escaper.java
- * /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/primitives/Floats.java
- * /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/base/Stopwatch.java
- * /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/io/FileBackedOutputStream.java
- * /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/base/CharMatcher.java
- * /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/escape/UnicodeEscaper.java
- * /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/collect/FluentIterable.java
- * /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/io/MultiReader.java
- * /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/base/Converter.java
- * /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/primitives/Doubles.java
- * /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/net/InetAddresses.java
- * /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/primitives/Longs.java
- * /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/primitives/Bytes.java
- * /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/base/Joiner.java
- * /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/util/concurrent/SequentialExecutor.java
- * /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-jar/com/google/common/net/PercentEscaper.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2011 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not
* use this file except in compliance with the License. You may obtain a copy of
* the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations under
* the License.

*/

Found in path(s):

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/SortedMultiset.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/SortedMultisets.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2019 The Guava Authors

*

* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/graph/IncidentEdgeSet.java

* /opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/CompactHashing.java

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2008 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
/*
 * This method was rewritten in Java from an intermediate step of the Murmur hash function in
 * http://code.google.com/p/smhasher/source/browse/trunk/MurmurHash3.cpp, which contained the
 * following header:
 *
 * MurmurHash3 was written by Austin Appleby, and is placed in the public domain. The author
 * hereby disclaims copyright to this source code.
 */
```

Found in path(s):

```
*/opt/cola/permits/1288520115_1647351767.93/0/guava-31-0-1-jre-sources-
jar/com/google/common/collect/Hashing.java
```

1.95 micronaut-rxjava-2 1.3.0

1.95.1 Available under license :

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common

control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or

documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill,

work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.96 io-grpc-grpc-core 1.47.0

1.96.1 Available under license :

Envoy

Copyright The Envoy Project Authors

Licensed under Apache License 2.0. See LICENSE for terms.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications

represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without

modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. **Trademarks.** This License does not grant permission to use the trade

names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier

identification within third-party archives.

Copyright [yyyy] [name of copyright owner].

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

protoc-gen-validate

Copyright 2019 Envoy Project Authors

Licensed under Apache License 2.0. See LICENSE for terms.

zero-allocation-hashing

Copyright 2015 Higher Frequency Trading <http://www.higherfrequencytrading.com>

Licensed under Apache License 2.0. See LICENSE for terms.

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of

this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only

on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

/*

* Copyright 2015 The gRPC Authors

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Copyright 2014 The gRPC Authors

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

This product contains a modified portion of 'OkHttp', an open source
HTTP & SPDY client for Android and Java applications, which can be obtained
at:

- * LICENSE:
 - * [okhttp/third_party/okhttp/LICENSE](http://okhttp.org/third_party/okhttp/LICENSE) (Apache License 2.0)
- * HOMEPAGE:
 - * <https://github.com/square/okhttp>
- * LOCATION_IN_GRPC:
 - * [okhttp/third_party/okhttp](http://okhttp.org/third_party/okhttp)

This product contains a modified portion of 'Envoy', an open source
cloud-native high-performance edge/middle/service proxy, which can be
obtained at:

- * LICENSE:
 - * [xds/third_party/envoy/LICENSE](http://xds.org/third_party/envoy/LICENSE) (Apache License 2.0)
- * NOTICE:
 - * [xds/third_party/envoy/NOTICE](http://xds.org/third_party/envoy/NOTICE)
- * HOMEPAGE:
 - * <https://www.envoyproxy.io>
- * LOCATION_IN_GRPC:
 - * [xds/third_party/envoy](http://xds.org/third_party/envoy)

This product contains a modified portion of 'protoc-gen-validate (PGV)',
an open source protoc plugin to generate polyglot message validators,
which can be obtained at:

- * LICENSE:
 - * [xds/third_party/protoc-gen-validate/LICENSE](http://xds.org/third_party/protoc-gen-validate/LICENSE) (Apache License 2.0)
- * NOTICE:
 - * [xds/third_party/protoc-gen-validate/NOTICE](http://xds.org/third_party/protoc-gen-validate/NOTICE)

- * HOMEPAGE:
 - * <https://github.com/envoyproxy/protoc-gen-validate>
- * LOCATION_IN_GRPC:
 - * `xds/third_party/protoc-gen-validate`

This product contains a modified portion of 'udpa',
an open source universal data plane API, which can be obtained at:

- * LICENSE:
 - * `xds/third_party/udpa/LICENSE` (Apache License 2.0)
- * HOMEPAGE:
 - * <https://github.com/cncf/udpa>
- * LOCATION_IN_GRPC:
 - * `xds/third_party/udpa`

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You

institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.97 google-guice 4.1.0

1.97.1 Available under license :

Google Guice - Core Library
Copyright 2006-2016 Google, Inc.

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by

the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained

within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be

liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.98 commons-compress 1.21

1.98.1 Available under license :

Apache Commons Compress

Copyright 2002-2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<https://www.apache.org/>).

The files in the package `org.apache.commons.compress.archivers.sevenz`
were derived from the LZMA SDK, version 9.20 (C/ and CPP/7zip/),
which has been placed in the public domain:

"LZMA SDK is placed in the public domain." (<http://www.7-zip.org/sdk.html>)

The test file `lbzip2_32767.bz2` has been copied from `libbzip2`'s source
repository:

This program, "bzip2", the associated library "libbzip2", and all
documentation, are copyright (C) 1996-2019 Julian R Seward. All
rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
2. The origin of this software must not be misrepresented; you must
not claim that you wrote the original software. If you use this
software in a product, an acknowledgment in the product
documentation would be appreciated but is not required.
3. Altered source versions must be plainly marked as such, and must
not be misrepresented as being the original software.
4. The name of the author may not be used to endorse or promote
products derived from this software without specific prior written
permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS
OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED

WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Julian Seward, jseward@acm.org

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work

(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses

granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]"

replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.99 jersey's-jersey 2.34

1.99.1 Available under license :

Notice for Jersey Core Server module

This content is produced and maintained by the Eclipse Jersey project.

* <https://projects.eclipse.org/projects/ee4j.jersey>

Trademarks

Eclipse Jersey is a trademark of the Eclipse Foundation.

Copyright

All content is the property of the respective authors or their employers. For more information regarding authorship of content, please consult the listed source code repository logs.

Declared Project Licenses

This program and the accompanying materials are made available under the terms of the Eclipse Public License v. 2.0 which is available at <http://www.eclipse.org/legal/epl-2.0>. This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License v. 2.0 are satisfied: GNU General Public License, version 2 with the GNU Classpath Exception which is available at <https://www.gnu.org/software/classpath/license.html>.

SPDX-License-Identifier: EPL-2.0 OR GPL-2.0 WITH Classpath-exception-2.0

Source Code

The project maintains the following source code repositories:

* <https://github.com/eclipse-ee4j/jersey>

Third-party Content

org.glassfish.jersey.server.internal.monitoring.core

* License: Apache License, 2.0

* Copyright (c) 2015-2018 Oracle and/or its affiliates. All rights reserved.

* Copyright 2010-2013 Coda Hale and Yammer, Inc.

org.objectweb.asm Version 8.0

* License: Modified BSD (<http://asm.objectweb.org/license.html>)

* Copyright: (c) 2000-2011 INRIA, France Telecom. All rights reserved.

W3.org documents

* License: W3C License

* Copyright: Copyright (c) 1994-2001 World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University). All Rights Reserved.

<http://www.w3.org/Consortium/Legal/>

Copyright (c) 2018 Oracle and/or its affiliates. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the Eclipse Foundation, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,

PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Notice for Jersey Core Common module

This content is produced and maintained by the Eclipse Jersey project.

* <https://projects.eclipse.org/projects/ee4j.jersey>

Trademarks

Eclipse Jersey is a trademark of the Eclipse Foundation.

Copyright

All content is the property of the respective authors or their employers. For more information regarding authorship of content, please consult the listed source code repository logs.

Declared Project Licenses

This program and the accompanying materials are made available under the terms of the Eclipse Public License v. 2.0 which is available at <http://www.eclipse.org/legal/epl-2.0>. This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License v. 2.0 are satisfied: GNU General Public License, version 2 with the GNU Classpath Exception which is available at <https://www.gnu.org/software/classpath/license.html>.

SPDX-License-Identifier: EPL-2.0 OR GPL-2.0 WITH Classpath-exception-2.0

Source Code

The project maintains the following source code repositories:

* <https://github.com/eclipse-ee4j/jersey>

Third-party Content

Google Guava Version 18.0

* License: Apache License, 2.0

* Copyright: (C) 2009 The Guava Authors

JSR-166 Extension - JEP 266

* License: Creative Commons 1.0 (CC0)

* No copyright

* Written by Doug Lea with assistance from members of JCP JSR-166

* Expert Group and released to the public domain, as explained at

* <http://creativecommons.org/publicdomain/zero/1.0/>
Notice for Jersey Json Jackson module
This content is produced and maintained by the Eclipse Jersey project.

* <https://projects.eclipse.org/projects/ee4j.jersey>

Trademarks

Eclipse Jersey is a trademark of the Eclipse Foundation.

Copyright

All content is the property of the respective authors or their employers. For more information regarding authorship of content, please consult the listed source code repository logs.

Declared Project Licenses

This program and the accompanying materials are made available under the terms of the Eclipse Public License v. 2.0 which is available at <http://www.eclipse.org/legal/epl-2.0>. This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License v. 2.0 are satisfied: GNU General Public License, version 2 with the GNU Classpath Exception which is available at <https://www.gnu.org/software/classpath/license.html>.

SPDX-License-Identifier: EPL-2.0 OR GPL-2.0 WITH Classpath-exception-2.0

Source Code

The project maintains the following source code repositories:

* <https://github.com/eclipse-ee4j/jersey>

Third-party Content

Jackson JAX-RS Providers version 2.10.1

* License: Apache License, 2.0

* Project: <https://github.com/FasterXML/jackson-jaxrs-providers>

* Copyright: (c) 2009-2011 FasterXML, LLC. All rights reserved unless otherwise indicated.

/*

* Copyright (c) YYYY Oracle and/or its affiliates. All rights reserved.

*

* This program and the accompanying materials are made available under the

* terms of the Eclipse Public License v. 2.0, which is available at

* <http://www.eclipse.org/legal/epl-2.0>.

*

* This Source Code may also be made available under the following Secondary

* Licenses when the conditions for such availability set forth in the

* Eclipse Public License v. 2.0 are satisfied: GNU General Public License,

* version 2 with the GNU Classpath Exception, which is available at
* <https://www.gnu.org/software/classpath/license.html>.
*
* SPDX-License-Identifier: EPL-2.0 OR GPL-2.0 WITH Classpath-exception-2.0
*/
Notice for Jersey
This content is produced and maintained by the Eclipse Jersey project.

* Project home: <https://projects.eclipse.org/projects/ee4j.jersey>

Trademarks

Eclipse Jersey is a trademark of the Eclipse Foundation.

Copyright

All content is the property of the respective authors or their employers. For more information regarding authorship of content, please consult the listed source code repository logs.

Declared Project Licenses

This program and the accompanying materials are made available under the terms of the Eclipse Distribution License v. 1.0 which is available at <https://www.eclipse.org/org/documents/edl-v10.php>.

SPDX-License-Identifier: BSD-3-Clause

Source Code

The project maintains the following source code repositories:

* <https://github.com/eclipse-ee4j/jersey/examples>

Third-party Content

Angular JS, v1.6.6

* License MIT (<http://www.opensource.org/licenses/mit-license.php>)

* Project: <http://angularjs.org>

* Copyright: (c) 2010-2017 Google, Inc.

aopalliance Version 1

* License: all the source code provided by AOP Alliance is Public Domain.

* Project: <http://aopalliance.sourceforge.net>

* Copyright: Material in the public domain is not protected by copyright

Bean Validation API 2.0.2

* License: Apache License, 2.0

* Project: <http://beanvalidation.org/1.1/>

* Copyright: 2009, Red Hat, Inc. and/or its affiliates, and individual contributors

* by the @authors tag.

Bootstrap v3.3.7

- * License: MIT license (<https://github.com/twbs/bootstrap/blob/master/LICENSE>)
- * Project: <http://getbootstrap.com>
- * Copyright: 2011-2016 Twitter, Inc

Google Guava Version 18.0

- * License: Apache License, 2.0
- * Copyright (C) 2009 The Guava Authors

javax.inject Version: 1

- * License: Apache License, 2.0
- * Copyright (C) 2009 The JSR-330 Expert Group

Javassist Version 3.25.0-GA

- * License: Apache License, 2.0
- * Project: <http://www.javassist.org/>
- * Copyright (C) 1999- Shigeru Chiba. All Rights Reserved.

Jackson JAX-RS Providers Version 2.9.9

- * License: Apache License, 2.0
- * Project: <https://github.com/FasterXML/jackson-jaxrs-providers>
- * Copyright: (c) 2009-2011 FasterXML, LLC. All rights reserved unless otherwise indicated.

jQuery v1.12.4

- * License: jquery.org/license
- * Project: jquery.org
- * Copyright: (c) jQuery Foundation

jQuery Barcode plugin 0.3

- * License: MIT & GPL (<http://www.opensource.org/licenses/mit-license.php> & <http://www.gnu.org/licenses/gpl.html>)
- * Project: <http://www.pasella.it/projects/jquery/barcode>
- * Copyright: (c) 2009 Antonello Pasella antonello.pasella@gmail.com

JSR-166 Extension - JEP 266

- * License: CC0
- * No copyright
- * Written by Doug Lea with assistance from members of JCP JSR-166 Expert Group and released to the public domain, as explained at <http://creativecommons.org/publicdomain/zero/1.0/>

KineticJS, v4.7.1

- * License: MIT license (<http://www.opensource.org/licenses/mit-license.php>)
- * Project: <http://www.kineticjs.com>, <https://github.com/ericdrowell/KineticJS>
- * Copyright: Eric Rowell

org.objectweb.asm Version 7.2

- * License: Modified BSD (<http://asm.objectweb.org/license.html>)
- * Copyright (c) 2000-2011 INRIA, France Telecom. All rights reserved.

org.osgi.core version 6.0.0

- * License: Apache License, 2.0
- * Copyright (c) OSGi Alliance (2005, 2008). All Rights Reserved.

org.glassfish.jersey.server.internal.monitoring.core

- * License: Apache License, 2.0
- * Copyright (c) 2015-2018 Oracle and/or its affiliates. All rights reserved.
- * Copyright 2010-2013 Coda Hale and Yammer, Inc.

W3.org documents

- * License: W3C License
- * Copyright: Copyright (c) 1994-2001 World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University). All Rights Reserved.

<http://www.w3.org/Consortium/Legal/>

Notice for Jersey

This content is produced and maintained by the Eclipse Jersey project.

- * Project home: <https://projects.eclipse.org/projects/ee4j.jersey>

Trademarks

Eclipse Jersey is a trademark of the Eclipse Foundation.

Copyright

All content is the property of the respective authors or their employers. For more information regarding authorship of content, please consult the listed source code repository logs.

Declared Project Licenses

This program and the accompanying materials are made available under the terms of the Eclipse Public License v. 2.0 which is available at <http://www.eclipse.org/legal/epl-2.0>. This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License v. 2.0 are satisfied: GNU General Public License, version 2 with the GNU Classpath Exception which is available at <https://www.gnu.org/software/classpath/license.html>.

SPDX-License-Identifier: EPL-2.0 OR GPL-2.0 WITH Classpath-exception-2.0

Source Code

The project maintains the following source code repositories:

- * <https://github.com/eclipse-ee4j/jersey>

Third-party Content

Angular JS, v1.6.6

- * License MIT (<http://www.opensource.org/licenses/mit-license.php>)
- * Project: <http://angularjs.org>
- * Coyright: (c) 2010-2017 Google, Inc.

aopalliance Version 1

- * License: all the source code provided by AOP Alliance is Public Domain.
- * Project: <http://aopalliance.sourceforge.net>
- * Copyright: Material in the public domain is not protected by copyright

Bean Validation API 2.0.2

- * License: Apache License, 2.0
- * Project: <http://beanvalidation.org/1.1/>
- * Copyright: 2009, Red Hat, Inc. and/or its affiliates, and individual contributors
- * by the @authors tag.

Hibernate Validator CDI, 6.1.2.Final

- * License: Apache License, 2.0
- * Project: <https://beanvalidation.org/>
- * Repackaged in org.glassfish.jersey.server.validation.internal.hibernate

Bootstrap v3.3.7

- * License: MIT license (<https://github.com/twbs/bootstrap/blob/master/LICENSE>)
- * Project: <http://getbootstrap.com>
- * Copyright: 2011-2016 Twitter, Inc

Google Guava Version 18.0

- * License: Apache License, 2.0
- * Copyright (C) 2009 The Guava Authors

javax.inject Version: 1

- * License: Apache License, 2.0
- * Copyright (C) 2009 The JSR-330 Expert Group

Javassist Version 3.25.0-GA

- * License: Apache License, 2.0
- * Project: <http://www.javassist.org/>
- * Copyright (C) 1999- Shigeru Chiba. All Rights Reserved.

Jackson JAX-RS Providers Version 2.10.1

- * License: Apache License, 2.0
- * Project: <https://github.com/FasterXML/jackson-jaxrs-providers>
- * Copyright: (c) 2009-2011 FasterXML, LLC. All rights reserved unless otherwise indicated.

jQuery v1.12.4

- * License: jquery.org/license

- * Project: jquery.org
- * Copyright: (c) jQuery Foundation

jQuery Barcode plugin 0.3

- * License: MIT & GPL (<http://www.opensource.org/licenses/mit-license.php> & <http://www.gnu.org/licenses/gpl.html>)
- * Project: <http://www.pasella.it/projects/jquery/barcode>
- * Copyright: (c) 2009 Antonello Pasella antonello.pasella@gmail.com

JSR-166 Extension - JEP 266

- * License: CC0
- * No copyright
- * Written by Doug Lea with assistance from members of JCP JSR-166 Expert Group and released to the public domain, as explained at <http://creativecommons.org/publicdomain/zero/1.0/>

KineticJS, v4.7.1

- * License: MIT license (<http://www.opensource.org/licenses/mit-license.php>)
- * Project: <http://www.kineticjs.com>, <https://github.com/ericdrowell/KineticJS>
- * Copyright: Eric Rowell

org.objectweb.asm Version 8.0

- * License: Modified BSD (<http://asm.objectweb.org/license.html>)
- * Copyright (c) 2000-2011 INRIA, France Telecom. All rights reserved.

org.osgi.core version 6.0.0

- * License: Apache License, 2.0
- * Copyright (c) OSGi Alliance (2005, 2008). All Rights Reserved.

org.glassfish.jersey.server.internal.monitoring.core

- * License: Apache License, 2.0
- * Copyright (c) 2015-2018 Oracle and/or its affiliates. All rights reserved.
- * Copyright 2010-2013 Coda Hale and Yammer, Inc.

W3.org documents

- * License: W3C License
- * Copyright: Copyright (c) 1994-2001 World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University). All Rights Reserved. <http://www.w3.org/Consortium/Legal/>
- * Eclipse Public License - v 2.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial content
Distributed under this Agreement, and

b) in the case of each subsequent Contributor:

- i) changes to the Program, and
- ii) additions to the Program;

where such changes and/or additions to the Program originate from and are Distributed by that particular Contributor. A Contribution "originates" from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include changes or additions to the Program that are not Modified Works.

"Contributor" means any person or entity that Distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions Distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

"Derivative Works" shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

"Modified Works" shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

"Distribute" means the acts of a) distributing or b) making available in any manner that enables the transfer of a copy.

"Source Code" means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Secondary License" means either the GNU General Public License, Version 2.0, or any later versions of that license, including any

exceptions or additional permissions as identified by the initial Contributor.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

e) Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

3. REQUIREMENTS

3.1 If a Contributor Distributes the Program in any form, then:

- a) the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and
- b) the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:
 - i) effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
 - ii) effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
 - iii) does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and
 - iv) requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.

3.2 When the Program is Distributed as Source Code:

- a) it must be made available under this Agreement, or if the Program (i) is combined with other material in a separate file or files made available under a Secondary License, and (ii) the initial Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available under the terms of such Secondary Licenses, and
- b) a copy of this Agreement must be included with each copy of the Program.

3.3 Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability ("notices") contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published,

Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.

Exhibit A - Form of Secondary Licenses Notice

"This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: {name license(s), version(s), and exceptions or additional permissions here}."

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor
Boston, MA 02110-1335
USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free

Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either

verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in

themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR

OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1335 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type
`show w'. This is free software, and you are welcome to redistribute
it under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the
program `Gnomovision' (which makes passes at compilers) written by
James Hacker.
```

```
signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

CLASSPATH EXCEPTION

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this

exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

/*

* Copyright (c) YYYY Oracle and/or its affiliates. All rights reserved.

*

* This program and the accompanying materials are made available under the

* terms of the Eclipse Distribution License v. 1.0, which is available at

* <http://www.eclipse.org/org/documents/edl-v10.php>.

*

* SPDX-License-Identifier: BSD-3-Clause

*/

Notice for Jersey Bean Validation module

This content is produced and maintained by the Eclipse Jersey project.

* <https://projects.eclipse.org/projects/ee4j.jersey>

Trademarks

Eclipse Jersey is a trademark of the Eclipse Foundation.

Copyright

All content is the property of the respective authors or their employers. For more information regarding authorship of content, please consult the listed source code repository logs.

Declared Project Licenses

This program and the accompanying materials are made available under the terms of the Eclipse Public License v. 2.0 which is available at <http://www.eclipse.org/legal/epl-2.0>. This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License v. 2.0 are satisfied: GNU General Public License, version 2 with the GNU Classpath Exception which is available at <https://www.gnu.org/software/classpath/license.html>.

SPDX-License-Identifier: EPL-2.0 OR GPL-2.0 WITH Classpath-exception-2.0

Source Code

The project maintains the following source code repositories:

* <https://github.com/eclipse-ee4j/jersey>

Third-party Content

Hibernate Validator CDI, 6.1.2.Final

* License: Apache License, 2.0

* Project: <https://beanvalidation.org/>

* Repackaged in org.glassfish.jersey.server.validation.internal.hibernate

1.100 assertj-fluent-assertions 3.21.0

1.100.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance
 * with
 * the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on
 * an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the
 * specific language governing permissions and limitations under the License.
 *
 * Copyright 2012-2021 the original author or authors.
 */
/**
 * Base class for implementations of { @link ObjectEnumerableAssert } whose actual value type is
 * { @link Collection }.
 *
 * @param <SELF> the "self" type of this assertion class. Please read Emulating 'self types' using Java Generics to simplify fluent API implementation
 * for more details.
 * @param <ACTUAL> the type of the "actual" value.
 * @param <ELEMENT> the type of elements of the "actual" value.
 * @param <ELEMENT_ASSERT> used for navigational assertions to return the right assert type.
 *
 * @author Yvonne Wang
 * @author Alex Ruiz
 * @author Mathieu Baechler
 * @author Joel Costigliola
 * @author Maciej Jaskowski
 * @author Nicolas François
 * @author Mikhail Mazursky
 * @author Mateusz Haligowski
 * @author Lovro Pandzic
 * @author Marko Bekhta
 */
```

Found in path(s):

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-

jar/org/assertj/core/api/AbstractIterableAssert.java

No license file was found, but licenses were detected in source scan.

```
/*
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance
 * with
 * the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on
 * an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the
 * specific language governing permissions and limitations under the License.
 *
 * Copyright 2012-2021 the original author or authors.
 */
/**
 * Base class for all implementations of assertions for { @link Byte}s.
 *
 * @param <SELF> the "self" type of this assertion class. Please read &quot;<a href="http://bit.ly/1IZIRcY"
 *     target="_blank">Emulating 'self types' using Java Generics to simplify fluent API
 * implementation</a>&quot;
 *     for more details.
 *
 * @author Drummond Dawson
 * @author Yvonne Wang
 * @author David DIDIER
 * @author Ansgar Konermann
 * @author Alex Ruiz
 * @author Mikhail Mazursky
 * @author Nicolas François
 * @author Cal027
 */
```

Found in path(s):

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-

jar/org/assertj/core/api/AbstractByteAssert.java

No license file was found, but licenses were detected in source scan.

```
/*
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance
 * with
 * the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on
```

* an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the

* specific language governing permissions and limitations under the License.

*

* Copyright 2012-2021 the original author or authors.

*/

/**

* Assertions methods applicable to groups of objects (e.g. arrays or collections.)

*

* @param <SELF> the "self" type of this assertion class. Please read "<a href="http://bit.ly/1IZIRcY"

* target="_blank">Emulating 'self types' using Java Generics to simplify fluent API

implementation"

* for more details.

* @param <ELEMENT> the type of elements of the "actual" value.

*

* @author Yvonne Wang

* @author Alex Ruiz

* @author Nicolas François

* @author Mikhail Mazursky

* @author Joel Costigliola

* @author Nicolas François

* @author Florent Biville

*/

Found in path(s):

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-

jar/org/assertj/core/api/ObjectEnumerableAssert.java

No license file was found, but licenses were detected in source scan.

/*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with

* the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on

* an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the

* specific language governing permissions and limitations under the License.

*

* Copyright 2012-2021 the original author or authors.

*/

/**

* Base class for all implementations of assertions for { @code CharSequence}s.

*

* @param <SELF> the "self" type of this assertion class. Please read "<a href="http://bit.ly/1IZIRcY"

* target="_blank">Emulating 'self types' using Java Generics to simplify fluent API

implementation"
* for more details.
* @param <ACTUAL> the type of the "actual" value.
*
* @author Yvonne Wang
* @author David DIDIER
* @author Alex Ruiz
* @author Joel Costigliola
* @author Mikhail Mazursky
* @author Nicolas Francois
* @author Daniel Weber
*/

Found in path(s):

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractCharSequenceAssert.java

No license file was found, but licenses were detected in source scan.

/*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with
* the License. You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on
* an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the
* specific language governing permissions and limitations under the License.
*
* Copyright 2012-2021 the original author or authors.
*/
/**
* Base class for all implementations of assertions for { @link List}s.
* @param <SELF> the "self" type of this assertion class. Please read "Emulating 'self types' using Java Generics to simplify fluent API
* implementation"
* for more details.
* @param <ACTUAL> the type of the "actual" value.
* @param <ELEMENT> the type of elements of the "actual" value.
* @param <ELEMENT_ASSERT> used for navigational assertions to return the right assert type.
*
* @author Yvonne Wang
* @author Alex Ruiz
* @author Joel Costigliola
* @author Mikhail Mazursky
* @author Jacek Jackowiak
*/

Found in path(s):

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractListAssert.java

No license file was found, but licenses were detected in source scan.

/*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with

* the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the

* specific language governing permissions and limitations under the License.

*

* Copyright 2012-2021 the original author or authors.

*/

/**

* Base class for all implementations of assertions for { @link BigDecimal}s.

*

* @param <SELF> the "self" type of this assertion class. Please read ["http://bit.ly/1IZIRcY"](http://bit.ly/1IZIRcY)

* `target="_blank">Emulating 'self types' using Java Generics to simplify fluent API implementation`

* `for more details.`

*

* @author Drummond Dawson

* @author David DIDIER

* @author Ted M. Young

* @author Yvonne Wang

* @author Alex Ruiz

* @author Joel Costigliola

* @author Mikhail Mazursky

*/

Found in path(s):

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractBigDecimalAssert.java

No license file was found, but licenses were detected in source scan.

/*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with

* the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

```

*
* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on
* an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the
* specific language governing permissions and limitations under the License.
*
* Copyright 2012-2021 the original author or authors.
*/
/**
* Base class for all assertions.
*
* @param <SELF> the "self" type of this assertion class. Please read &quot;<a href="http://bit.ly/1IZIRcY"
*     target="_blank">Emulating 'self types' using Java Generics to simplify fluent API
implementation</a>&quot;
*     for more details.
* @param <ACTUAL> the type of the "actual" value.
*
* @author Alex Ruiz
* @author Joel Costigliola
* @author Mikhail Mazursky
* @author Nicolas François
*/

```

Found in path(s):

```

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-
jar/org/assertj/core/api/AbstractAssert.java

```

No license file was found, but licenses were detected in source scan.

```

/*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance
with
* the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on
* an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the
* specific language governing permissions and limitations under the License.
*
* Copyright 2012-2021 the original author or authors.
*/
/**
* Base class for all implementations of assertions for { @link Throwable}s.
*
* @param <SELF> the "self" type of this assertion class. Please read &quot;<a href="http://bit.ly/1IZIRcY"
*     target="_blank">Emulating 'self types' using Java Generics to simplify fluent API
implementation</a>&quot;

```

* for more details.
* @param <ACTUAL> the type of the "actual" value.
*
* @author David DIDIER
* @author Alex Ruiz
* @author Joel Costigliola
* @author Mikhail Mazursky
* @author Jack Gough
* @author Mike Gilchrist
*/

Found in path(s):

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractThrowableAssert.java
No license file was found, but licenses were detected in source scan.

/*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with
with
* the License. You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on
* an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the
* specific language governing permissions and limitations under the License.
*
* Copyright 2012-2021 the original author or authors.

*/
/**
* Base class for all implementations of assertions for { @link Integer}s.
*
* @param <SELF> the "self" type of this assertion class. Please read "Emulating 'self types' using Java Generics to simplify fluent API implementation"
* for more details.
*
* @author Drummond Dawson
* @author Yvonne Wang
* @author David DIDIER
* @author Ansgar Konermann
* @author Alex Ruiz
* @author Joel Costigliola
* @author Mikhail Mazursky
* @author Nicolas François
* @author Cal027
*/

Found in path(s):

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractIntegerAssert.java

No license file was found, but licenses were detected in source scan.

/*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with

* the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the

* specific language governing permissions and limitations under the License.

*

* Copyright 2012-2021 the original author or authors.

*/

/**

* Base class for all implementations of assertions for { @link File}s.

*

* @param <SELF> the "self" type of this assertion class. Please read ["http://bit.ly/1IZIRcY"](http://bit.ly/1IZIRcY)

* `target="_blank">Emulating 'self types' using Java Generics to simplify fluent API implementation`

* `for more details.`

*

* @author David DIDIER

* @author Yvonne Wang

* @author Alex Ruiz

* @author Olivier Michallat

* @author Olivier Demeijer

* @author Mikhail Mazursky

* @author Jean-Christophe Gay

* @author Valeriy Vyrva

* @author Nikolaos Georgiou

*/

Found in path(s):

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractFileAssert.java

No license file was found, but licenses were detected in source scan.

/*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with

* the License. You may obtain a copy of the License at

*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on
* an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the
* specific language governing permissions and limitations under the License.
*
* Copyright 2012-2021 the original author or authors.
*/

Found in path(s):

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/StandardComparisonStrategy.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/condition/MappedCondition.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeAtIndex.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeInSameSecond.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/AbstractShouldHaveTextContent.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveParent.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/IntArrayAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/filter/FilterOperator.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/LongAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/MapAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/OptionalShouldBeEmpty.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/Spliterators.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/ArraySortedAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/future/Warning.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/Objects.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/ClassBasedNavigableListAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/uri/ShouldHaveHost.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveContent.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/BDDSoftAssertions.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/uri/ShouldHaveFragment.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/Float2DArrayAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/Assertions.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ElementsShouldNotBe.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeSorted.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/Iterables.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/DoublePredicateAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/FieldByFieldComparator.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/SpliteratorAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/iterable/ThrowingExtractor.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/OptionalIntAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractArrayAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ElementsShouldBeExactly.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractPredicateLikeAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeAbsolutePath.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ConstructorInvoker.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractIterableSizeAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractOptionalAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldNotStartWith.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeEqualWithinPercentage.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldNotBeEqualNormalizingWhitespace.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/extractor/ToStringExtractor.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/AbstractComparisonStrategy.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeEqualIgnoringCase.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AtomicReferenceArrayAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/Objects.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeWritable.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractDurationAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractUrlAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/ByteArrayAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeInSameMonth.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveMessageFindingMatchRegex.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/future/ShouldBeDone.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldNotBeFinite.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/Predicates.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/WritableAssertionInfo.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/DescribableComparator.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/AssertionErrorMessagesAggregator.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldContainOneOrMoreWhitespaces.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/JUnitBDDSoftAssertions.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/ObjectAssertFactory.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/CommonErrors.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveTime.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/presentation/Representation.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeInstance.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/RealNumbers.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/AnyElementShouldMatch.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldContainEntry.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/JUnitJupiterSoftAssertions.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/matcher/AssertionMatcher.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeMixedCase.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/Java6SoftAssertions.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldNotMatch.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/Float2DArrays.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractAtomicFieldUpdaterAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldEndWithPath.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/ComparatorBasedComparisonStrategy.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/IterableUtil.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/ZonedDateTimeAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/annotations/Beta.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/URLs.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/Uri.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/LongArrayAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeGreaterOrEqual.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/BinaryDiff.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/FloatComparator.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldContainRecursively.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/diff/myers/Equalizer.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/Abstract2DArrayAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ClassModifierShouldBe.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/Arrays.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractMapSizeAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/introspection/FieldUtils.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/diff/myers/DiffNode.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/MessageFormatter.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/Maps.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/ObjectAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeBefore.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveAtLeastOneElementOfType.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/ShortArrayAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/groups/Tuple.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveNoFields.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/FactoryBasedNavigableListAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/Java6StandardSoftAssertionsProvider.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/GenericComparableAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/future/ShouldNotBeCompletedExceptionally.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/IterableSizeAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/condition/Negative.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/Byte2DArrays.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractFloatArrayAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeSame.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveSameContent.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldNotBeInstanceOfAny.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveDuration.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/StringAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/uri/ShouldHaveQuery.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/ComparableAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/description/TextDescription.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/data/TemporalUnitOffset.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/uri/ShouldHaveUserInfo.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/CommonValidations.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldNotContainKey.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractPredicateAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldContainKey.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldNotBeEqual.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/uri/ShouldHavePath.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/future/ShouldBeCompleted.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveOnlyElementsOfType.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldNotContainSubsequence.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/Classes.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveSameDimensionsAs.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeEmpty.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeBase64.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractDoubleArrayAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/Assumptions.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/ThrowingConsumer.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/WithAssertions.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/Floats.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ElementsShouldHaveExactly.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/SoftAssertionsStatement.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ElementsShouldBeAtLeast.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/Booleans.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveCause.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractObjectArrayAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/future/ShouldNotBeDone.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/Boolean2DArrays.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldNotBeNull.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/extractor/ByNameMultipleExtractor.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/Longs.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/ElementsSatisfyingConsumer.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/diff/ChangeDelta.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeOdd.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/SoftAssertionsProvider.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/FileAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractIteratorAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/ErrorCollector.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/Bytes.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldNotBeIn.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/OptionalDoubleShouldHaveValueCloseToPercentage.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveCauseInstance.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AtomicIntegerAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/LocalDateTimeAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeSymbolicLink.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/ObjectArrayElementComparisonStrategy.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/ThrowableAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/SoftAssertions.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeUpperCase.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeBeforeOrEqualTo.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/introspection/Introspection.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/extractor/ByNameSingleExtractor.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/future/ShouldHaveFailed.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/Boolean2DArrayAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/BasicErrorMessageFactory.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldContainCharSequence.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/Java6JUnitBDDSoftAssertions.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/JUnitSoftAssertions.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBe.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/InstanceOfAssertFactory.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHavePropertyOrField.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/OffsetDateTimeAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/array2d/Array2dElementShouldBeDeepEqual.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/ListAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/presentation/UnicodeRepresentation.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/WholeNumbers.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractBooleanArrayAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/ThrowableAssertAlternative.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/introspection/PropertySupport.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/introspection/FieldSupport.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/NumberAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ElementsShouldHaveAtMost.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldNotAccept.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldNotBeInstance.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/filter/NotInFilter.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/junit/jupiter/SoftlyExtension.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeInSameDay.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/Paths.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeInSameHourWindow.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/CharacterAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/Iterators.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeInstanceOfAny.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveNext.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractLocalDateAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveSizeBetween.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveNoCause.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldNotContainNull.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/BDDSoftAssertionsProvider.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveSize.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/OnFieldsComparator.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/WithAssumptions.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeNormalized.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeRegularFile.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeEqualNormalizingPunctuationAndWhitespace.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveReference.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldMatchPattern.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractComparableAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/data/Percentage.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/TextFileWriter.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeRelativePath.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/StandardSoftAssertionsProvider.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldNotHaveDuplicates.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/uri/ShouldHaveAuthority.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/diff/DeleteDelta.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/BDDAssertions.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveNoExtension.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveScale.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/ChronoLocalDateTimeComparator.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/NotThrownAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ElementsShouldBeAtMost.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeAfter.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/Diff.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/Object2DArrayAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveDimensions.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveSizeGreaterThanOrEqualTo.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/description/EmptyTextDescription.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/Sets.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/Java6BDDAssertions.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/Files.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/extractor/Extractors.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/groups/Properties.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/IterableAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/NoElementsShouldMatch.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/AssertionErrorMessagesAggregator.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeEqualByComparingOnlyGivenFields.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/IgnoringFieldsComparator.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/recursive/comparison/FieldComparators.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/AssertJMultipleFailuresError.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/OptionalAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractByteArrayAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/UrlAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeAnnotation.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/GroupTypeDescription.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldContainOnlyNulls.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/presentation/HexadecimalRepresentation.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeEqualWithinOffset.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldContainExactlyInAnyOrder.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/OptionalLongAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeInSameMinuteWindow.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/ComparisonStrategy.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/ExtensionPoints.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/CharSequenceAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractOffsetTimeAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/Short2DArrayAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveSameHashCode.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/BigIntegers.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeLessOrEqual.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/CharArrays.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/Doubles.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/introspection/MethodSupport.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/IndexedObjectEnumerableAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/configuration/PreferredAssumptionException.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ErrorMessageFactory.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeUnmodifiable.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldNotBeEqualIgnoringWhitespace.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/PeriodAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/SoftAssertionsRule.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AtomicIntegerArrayAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/OptionalDoubleShouldHaveValueCloseToOffset.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/extractor/ResultOfExtractor.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AssumptionExceptionFactory.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/Descriptable.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/Int2DArrayAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/iterable/Extractor.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeAfterOrEqualTo.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveSameSizeAs.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHavePropertyOrFieldWithValue.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/future/ShouldHaveFailedWithin.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeInTheFuture.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/BDDAssumptions.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/configuration/Configuration.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/BooleanArrayAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/CompletableFutureAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldContainOnlyWhitespaces.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/ShortAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/LocalTimeAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeDirectory.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/SubarraysShouldHaveSameSize.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/BooleanAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ElementsShouldSatisfy.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/uri/ShouldHaveParameter.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractCharArrayAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/Double2DArrays.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/Object2DArrays.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldContainNull.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/InstantAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveStamp.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeFinite.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/HamcrestCondition.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveName.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/introspection/MemberUtils.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldNotContain.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveSameTime.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/FloatingPointNumberAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/OptionalShouldContain.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/future/ShouldNotBeCompleted.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/diff/InsertDelta.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/OptionalDoubleAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractCompletableFutureAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractTemporalAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/Throwables.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/ConfigurableRecursiveFieldByFieldComparator.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/RecursiveComparisonAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/Java6BDDSoftAssertionsProvider.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/data/Offset.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveFields.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/BigDecimals.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldContainAtIndex.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/condition/Join.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/FloatAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldContainCharSequenceOnlyOnce.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/uri/ShouldHavePort.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/introspection/ClassUtils.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeEqualByComparingFieldByFieldRecursively.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/Java6BDDSoftAssertions.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AtomicReferenceFieldUpdaterAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/CheckReturnValue.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/FloatArrays.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AtomicLongArrayAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldContainExactly.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveSameHourAs.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/future/ShouldBeCancelled.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveSizeLessThanOrEqualTo.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBePeriod.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeInSameMinute.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/ProxifyMethodChangingTheObjectUnderTest.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeMarked.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveNoParent.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/future/ShouldBeCompletedExceptionally.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/Fail.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/PathNaturalOrderComparator.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractLocalTimeAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldNotContainAnyWhitespaces.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractInstantAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/MultipleAssertionsError.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/DeepDifference.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/diff/myers/MyersDiff.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeBetween.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/AssertionErrorFactory.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveSameClass.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/Int2DArrays.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/PredicateAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/Condition.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AssertProvider.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldNotBeEqualWithinOffset.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveMessageMatchingRegex.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/DoubleComparator.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/DefaultAssertionErrorCollector.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeCloseTo.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeEmptyDirectory.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/CanIgnoreReturnValue.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeIn.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/Maps.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldContain.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AssertionsForClassTypes.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AtomicLongFieldUpdaterAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeOfClassIn.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveCauseExactlyInstance.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeBlank.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/ClassBasedNavigableIterableAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractLongArrayAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/configuration/ConfigurationProvider.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/DoubleArrays.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldNotHaveSameHashCode.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/diff/myers/PathNode.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/LongAdderAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/description/JoinDescription.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldNotBeInfinite.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractEnumerableAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/UnambiguousRepresentation.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/CollectionAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/introspection/IntrospectionError.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractLocalDateTimeAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldMatch.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/AbstractComparableNumberComparator.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/ShortArrays.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/Numbers.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractZonedDateTimeAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/InputStreamsException.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeEqualIgnoringNewLineDifferences.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeAtSameInstant.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/Futures.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/OptionalShouldContainInstanceOf.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeLess.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/CharArrayAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldExist.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ZippedElementsShouldSatisfy.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeAbstract.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/ArrayWrapperList.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/DescriptionFormatter.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeCanonicalPath.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ElementsShouldNotHave.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldNotExist.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/description/Description.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/Byte2DArrayAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/configuration/Services.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AssertionsForInterfaceTypes.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/presentation/StandardRepresentation.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/Throwables.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ConditionAndGroupGenericParameterTypeShouldBeTheSame.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/future/ShouldNotBeCancelled.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ElementsShouldHave.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeExhausted.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeReadable.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveMessage.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/diff/DeltaComparator.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/diff/DiffAlgorithm.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/EnumerableAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/IntPredicateAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/exception/PathsException.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/DoubleArrayAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/SoftProxies.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/LongPredicateAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AssertionInfo.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/FailureMessages.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/IterableElementComparisonStrategy.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeSubstring.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/LongArrays.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/data/TemporalUnitWithinOffset.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/Streams.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldNotContainCharSequence.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveToString.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeEqualIgnoringNewLines.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldNotHaveToString.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveExtension.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeInfinite.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/NaturalOrderComparator.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/junit/jupiter/InjectSoftAssertions.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/condition/Not.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/OffsetTimeAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/IteratorAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveAllNullFields.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeEven.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/Long2DArrays.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/NoElementsShouldSatisfy.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ElementsShouldHaveAtLeast.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/uri/ShouldHaveProtocol.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeEqualIgnoringHours.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldNotBeBlank.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeSubsetOf.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHavePeriod.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/Assert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldNotBeExactlyInstanceOf.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeEqualWithTimePrecision.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldNotEndWith.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldNotHaveThrown.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldNotBe.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/future/ShouldNotHaveFailed.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveLineCount.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/Strings.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AtomicIntegerFieldUpdaterAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/NioFilesWrapper.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldContainValue.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/Closeables.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractPeriodAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeLowerCase.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/recursive/comparison/RecursiveComparisonDifferenceCalculator.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeFalse.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/future/ShouldBeCompletedWithin.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldContainPattern.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldEndWith.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/data/Index.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/TypeComparators.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldContainSequence.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldNotContainSequence.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/FloatArrayAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractPathAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveSuppressedException.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/Characters.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeEqualNormalizingUnicode.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldSatisfy.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/Char2DArrays.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/Arrays.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/InstanceOfAssertFactories.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeEqualIgnoringNanos.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ElementsShouldMatch.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/data/TemporalUnitLessThanOffset.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldContainValues.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AtomicLongAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/AtomicReferenceArrayElementComparisonStrategy.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AtomicBooleanAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeEqualNormalizingWhitespace.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldContainAnyOf.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldNotBeEqualWithinPercentage.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/JUnitJupiterBDDSoftAssertions.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeExactlyInstanceOf.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveSizeLessThan.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHave.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractStringAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeInSameHour.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/OffsetDateTimeByInstantComparator.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeAnArray.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractBigIntegerAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldContainSequenceOfCharSequence.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/Hexadecimals.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveNoSuppressedExceptions.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/Java6Assertions.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldContainOnlyDigits.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeInSameSecondWindow.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/Java6JUnitSoftAssertions.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AssertDelegateTarget.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/MapSizeAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeEqualToIgnoringFields.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/FactoryBasedNavigableIterableAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/diff/myers/Snake.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/NullSafeComparator.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/Short2DArrays.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/ObjectArrayAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeExecutable.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldNotContainAtIndex.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/filter/InFilter.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/OptionalShouldBePresent.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/BinaryDiffResult.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/recursive/comparison/ComparisonKeyDifference.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveBinaryContent.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/groups/FieldsOrPropertiesExtractor.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/Lists.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveExactlyTypes.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/DateUtil.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/Urls.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AtomicReferenceAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeAfterYear.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/ChronoZonedDateTimeByInstantComparator.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldAccept.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/InputStreamAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/Digests.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/ComparatorFactory.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/BigIntegerComparator.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/DoubleAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/ExtendedByTypesComparator.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/UriAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/ObjectArrays.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/xml/XmlStringPrettyFormatter.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/junit/jupiter/SoftAssertionsExtension.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AutoCloseableBDDSoftAssertions.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeFile.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/data/MapEntry.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/uri/ShouldHaveAnchor.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractUriAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/Double2DArrayAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldNotMatchPattern.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/ClassAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldStartWithPath.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/recursive/comparison/DualValue.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/IntArrays.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/Char2DArrayAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/description/LazyTextDescription.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeTrue.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveNonNullFields.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/condition/AllOf.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/ClassLoadingStrategyFactory.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/Lists.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/diff/DiffUtils.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeEqualIgnoringSeconds.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractOptionalIntAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AfterAssertionErrorCollected.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/DurationAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/LocalDateAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/uri/ShouldHaveScheme.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldStartWith.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/InputStreams.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/AssertionErrorCreator.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractSpliteratorAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/BigDecimalAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AssertFactory.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeEqualIgnoringTimezone.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/Integers.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/recursive/comparison/RecursiveComparisonConfiguration.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/IterableDiff.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeEqual.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/Comparables.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/filter/Filters.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AtomicMarkableReferenceAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/presentation/PredicateDescription.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveSuperclass.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/Array2DAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/BigIntegerAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/condition/VerboseCondition.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/IntegerAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeGreater.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/Files.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/WithThrowable.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeNullOrEmpty.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldContainEntries.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/recursive/comparison/DualValueDeque.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldNotBeSame.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldNotContainPattern.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/Preconditions.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/Arrays2D.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeBeforeYear.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldNotHave.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/presentation/NumberGrouping.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeInThePast.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldContainOnly.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ElementsShouldBe.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldNotBeEqualIgnoringCase.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldNotContainValue.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/ThrowableTypeAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/filter/NotFilter.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveSizeGreaterThan.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/Strings.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractShortArrayAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveNoSuperclass.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/BigDecimalComparator.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/diff/Patch.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/TriFunction.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveMethods.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldOnlyHaveFields.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveCauseReference.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AssertionErrorCollector.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldContainSubsequenceOfCharSequence.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldNotHaveAnyElementsOfTypes.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldNotContainKeys.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/Dates.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractFutureAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractOffsetDateTimeAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/SoftAssertionError.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/UnsatisfiedRequirement.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/Failures.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/condition/DoesNotHave.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldNotHaveSameClass.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeEqualIgnoringWhitespace.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldContainOnlyKeys.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/diff/Delta.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/presentation/BinaryRepresentation.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AutoCloseableSoftAssertionsProvider.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/introspection/PropertyOrFieldSupport.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/recursive/comparison/FieldLocation.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractSoftAssertions.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/recursive/comparison/ComparisonDifference.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractAtomicReferenceAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AutoCloseableSoftAssertions.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeInterface.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/condition/AnyOf.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldOnlyHaveElementsOfTypes.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldContainsOnlyOnce.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveRootCauseInstance.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeEqualIgnoringMinutes.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/Shorts.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldContainKeys.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveValue.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeAssignableFrom.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldContainSubsequence.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/BooleanArrays.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/VisibleForTesting.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/Conditions.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractOptionalLongAssert.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/DigestDiff.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/uri/ShouldBeEqualToWithSortedQueryParameters.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeInSameYear.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveRootCause.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldNotContainOnlyWhitespaces.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractIntArrayAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/ByteArrays.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/data/TemporalOffset.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldNotBeBetween.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AtomicStampedReferenceAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/FutureAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldNotBeEmpty.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/Long2DArrayAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveDigest.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveRootCauseExactlyInstance.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractOptionalDoubleAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldNotBeEqualComparingFieldByFieldRecursively.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHavePackage.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/util/diff/Chunk.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/DateAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveAnnotations.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/ByteAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveAtIndex.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AssertJProxySetup.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldBeToday.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/internal/ErrorMessage.java

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldNotBeOfClassIn.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/PathAssert.java
* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/error/ShouldHaveDateField.java
No license file was found, but licenses were detected in source scan.

/*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with

* the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the

* specific language governing permissions and limitations under the License.

*

* Copyright 2012-2021 the original author or authors.

*/

/**

* Base class for all implementations of assertions for { @link Float}s.

*

* @param <SELF> the "self" type of this assertion class. Please read ["http://bit.ly/1IZIRcY"](http://bit.ly/1IZIRcY)

* [target="_blank">Emulating 'self types' using Java Generics to simplify fluent API implementation](#)

* [for more details.](#)

*

* @author Drummond Dawson

* @author Yvonne Wang

* @author Alex Ruiz

* @author Ansgar Konermann

* @author Mikhail Mazursky

* @author Nicolas François

* @author Jin Kwon

*/

Found in path(s):

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractFloatAssert.java

No license file was found, but licenses were detected in source scan.

/*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with

* the License. You may obtain a copy of the License at

```

*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on
* an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the
* specific language governing permissions and limitations under the License.
*
* Copyright 2012-2021 the original author or authors.
*/
/**
* Base class for all implementations of assertions for { @link Character}s.
*
* @param <SELF> the "self" type of this assertion class. Please read &quot;<a href="http://bit.ly/1IZIRcY"
*     target="_blank">Emulating 'self types' using Java Generics to simplify fluent API
implementation</a>&quot;
*     for more details.
*
* @author Yvonne Wang
* @author David DIDIER
* @author Ansgar Konermann
* @author Alex Ruiz
* @author Joel Costigliola
* @author Mikhail Mazursky
*/

```

Found in path(s):

```

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-
jar/org/assertj/core/api/AbstractCharacterAssert.java

```

No license file was found, but licenses were detected in source scan.

```

/*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance
with
* the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on
* an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the
* specific language governing permissions and limitations under the License.
*
* Copyright 2012-2021 the original author or authors.
*/
/**
* Base class for all implementations of assertions for { @link Short}s.
*

```

* @param <SELF> the "self" type of this assertion class. Please read [Emulating 'self types' using Java Generics to simplify fluent API implementation](http://bit.ly/1IZIRcY) for more details.

*
* @author Drummond Dawson
* @author Yvonne Wang
* @author David DIDIER
* @author Ansgar Konermann
* @author Alex Ruiz
* @author Mikhail Mazursky
* @author Nicolas François
* @author Cal027
*/

Found in path(s):

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractShortAssert.java

No license file was found, but licenses were detected in source scan.

/*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with

* the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on

* an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the

* specific language governing permissions and limitations under the License.

*

* Copyright 2012-2021 the original author or authors.

*/

/**

* Base class for all implementations of assertions for { @link Object}s.

*

* @param <SELF> the "self" type of this assertion class. Please read [Emulating 'self types' using Java Generics to simplify fluent API implementation](http://bit.ly/1IZIRcY) for more details.

* @param <ACTUAL> the type of the "actual" value.

*
* @author Yvonne Wang

* @author Alex Ruiz

* @author Nicolas François

*

* @author Mikhail Mazursky

* @author Joel Costigliola

* @author Libor Ondrusek

*/

Found in path(s):

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractObjectAssert.java

No license file was found, but licenses were detected in source scan.

/*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with

* the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the

* specific language governing permissions and limitations under the License.

*

* Copyright 2012-2021 the original author or authors.

*/

/**

* Base class for all implementations of assertions for { @link InputStream}s.

* @param <SELF> the "self" type of this assertion class. Please read ["http://bit.ly/1IZIRcY"](http://bit.ly/1IZIRcY)

* target="_blank">Emulating 'self types' using Java Generics to simplify fluent API

implementation"

* for more details.

* @param <ACTUAL> the type of the "actual" value.

*

* @author Matthieu Baechler

* @author Mikhail Mazursky

* @author Stefan Birkner

*/

Found in path(s):

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractInputStreamAssert.java

No license file was found, but licenses were detected in source scan.

/*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with

* the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on

* an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the

* specific language governing permissions and limitations under the License.

*

* Copyright 2012-2021 the original author or authors.

*/

/**

* Base class for all implementations of assertions for { @link LongAdder}s.

*

* @param <SELF> the "self" type of this assertion class. Please read "<a href="http://bit.ly/1IZIRcY"

* target="_blank">Emulating 'self types' using Java Generics to simplify fluent API

implementation"

* for more details.

*

* @author Grzegorz Piwowarek

* @since 3.16.0

*/

Found in path(s):

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-

jar/org/assertj/core/api/AbstractLongAdderAssert.java

No license file was found, but licenses were detected in source scan.

/*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with

* the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on

* an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the

* specific language governing permissions and limitations under the License.

*

* Copyright 2012-2021 the original author or authors.

*/

/**

* Base class for all implementations of assertions for { @link Long}s.

*

* @param <SELF> the "self" type of this assertion class. Please read "<a href="http://bit.ly/1IZIRcY"

* target="_blank">Emulating 'self types' using Java Generics to simplify fluent API

implementation"

* for more details.

*

* @author Drummond Dawson

* @author Yvonne Wang

* @author David DIDIER

* @author Ansgar Konermann
* @author Alex Ruiz
* @author Joel Costigliola
* @author Mikhail Mazursky
* @author Nicolas François
* @author Cal027
*/

Found in path(s):

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractLongAssert.java

No license file was found, but licenses were detected in source scan.

/*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with

* the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the

* specific language governing permissions and limitations under the License.

*

* Copyright 2012-2021 the original author or authors.

*/

/**

* Base class for all implementations of assertions for { @link Collection}s.

* @param <SELF> the "self" type of this assertion class. Please read <http://bit.ly/1IZIRcY>

* [target="_blank">Emulating 'self types' using Java Generics to simplify fluent API](#)

implementation

* for more details.

* @param <ACTUAL> the type of the "actual" value.

* @param <ELEMENT> the type of elements of the "actual" value.

* @param <ELEMENT_ASSERT> used for navigational assertions to return the right assert type.

*

* @since 3.21.0

*/

Found in path(s):

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractCollectionAssert.java

No license file was found, but licenses were detected in source scan.

/*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with

```

* the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on
* an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the
* specific language governing permissions and limitations under the License.
*
* Copyright 2012-2021 the original author or authors.
*/
/**
* Base class for all implementations of assertions for { @link Double}s.
*
* @param <SELF> the "self" type of this assertion class. Please read &quot;<a href="http://bit.ly/1IZIRcY"
*     target="_blank">Emulating 'self types' using Java Generics to simplify fluent API
implementation</a>&quot;
*     for more details.
*
* @author Drummond Dawson
* @author Yvonne Wang
* @author David DIDIER
* @author Alex Ruiz
* @author Ansgar Konermann
* @author Joel Costigliola
* @author Mikhail Mazursky
* @author Nicolas François
* @author Jack Gough
*/

```

Found in path(s):

```

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-
jar/org/assertj/core/api/AbstractDoubleAssert.java

```

No license file was found, but licenses were detected in source scan.

```

/*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance
with
* the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on
* an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the
* specific language governing permissions and limitations under the License.
*
* Copyright 2012-2021 the original author or authors.

```

```
*/
/**
 * Base class for all implementations of assertions for { @link Class }es.
 *
 * @param <SELF> the "self" type of this assertion class. Please read &quot;<a href="http://bit.ly/1IZIRcY"
 *     target="_blank">Emulating 'self types' using Java Generics to simplify fluent API
 * implementation</a>&quot;
 *     for more details.
 *
 * @author William Delanoue
 * @author Mikhail Mazursky
 */
```

Found in path(s):

```
*/opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-
jar/org/assertj/core/api/AbstractClassAssert.java
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance
 * with
 * the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on
 * an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the
 * specific language governing permissions and limitations under the License.
 *
 * Copyright 2012-2021 the original author or authors.
 */
```

```
/**
 * Base class for all implementations of assertions for { @link Date }s.
 *
 * <p>
 * Note that assertions with date parameter comes with two flavor, one is obviously a { @link Date } and the other is a
 * { @link String } representing a Date.<br>
 * For the latter, the default format follows ISO 8901 : "yyyy-MM-dd", user can override it with a custom format by
 * calling { @link #withDateFormat(DateFormat)}.<br>
 * The user custom format will then be used for all next Date assertions (i.e not limited to the current assertion) in
 * the test suite.<br>
 * To turn back to default format, simply call { @link #withDefaultDateFormatsOnly()}.
 *
 * @param <SELF> the "self" type of this assertion class. Please read "<a href="http://bit.ly/1IZIRcY"
 *     target="_blank">Emulating 'self types' using Java Generics to simplify fluent API implementation</a>" for
 *     more details.
 * @author Tomasz Nurkiewicz (thanks for giving assertions idea)
 * @author Joel Costigliola
 */
```

* @author Mikhail Mazursky
* @author William Delanoue
* @author Michal Kordas
* @author Eddú Meléndez
*/

Found in path(s):

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractDateAssert.java

No license file was found, but licenses were detected in source scan.

/*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with

* the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on

* an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the

* specific language governing permissions and limitations under the License.

*

* Copyright 2012-2021 the original author or authors.

*/

/**

* Base class for all implementations of assertions for { @link Map}s.

*

* @param <SELF> the "self" type of this assertion class. Please read ["http://bit.ly/1IZIRcY"](http://bit.ly/1IZIRcY)

* `target="_blank">Emulating 'self types' using Java Generics to simplify fluent API`

implementation"

* `for more details.`

* @param <ACTUAL> the type of the "actual" value.

* @param <K> the type of keys in the map.

* @param <V> the type of values in the map.

*

* @author David DIDIER

* @author Yvonne Wang

* @author Alex Ruiz

* @author Mikhail Mazursky

* @author Nicolas François

* @author dorzey

* @author Filip Hrisafov

*/

Found in path(s):

* /opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-jar/org/assertj/core/api/AbstractMapAssert.java

No license file was found, but licenses were detected in source scan.

```
/*
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance
 * with
 * the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on
 * an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the
 * specific language governing permissions and limitations under the License.
 *
 * Copyright 2012-2021 the original author or authors.
 */
/**
 * Base class for all implementations of assertions for { @link Boolean}s.
 *
 * @param <SELF> the "self" type of this assertion class. Please read &quot;<a href="http://bit.ly/1IZIRcY"
 *     target="_blank">Emulating 'self types' using Java Generics to simplify fluent API
 * implementation</a>&quot;
 *     for more details.
 *
 * @author Alex Ruiz
 * @author Yvonne Wang
 * @author David DIDIER
 * @author Ansgar Konermann
 * @author Mikhail Mazursky
 */
```

Found in path(s):

```
*/opt/cola/permits/1340815564_1654861496.226667/0/assertj-core-3-21-0-sources-1-
jar/org/assertj/core/api/AbstractBooleanAssert.java
```

1.101 kafka-protobuf-serializer 5.5.5

1.101.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

    <modelVersion>4.0.0</modelVersion>

    <parent>
        <groupId>io.confluent</groupId>
```

```
<artifactId>kafka-schema-registry-parent</artifactId>
<version>5.5.5</version>
</parent>

<licenses>
  <license>
    <name>Confluent Community License</name>
    <url>http://www.confluent.io/confluent-community-license</url>
    <distribution>repo</distribution>
  </license>
  <license>
    <name>Apache License 2.0</name>
    <url>http://www.apache.org/licenses/LICENSE-2.0.html</url>
    <distribution>repo</distribution>
  </license>
</licenses>
```

```
<artifactId>kafka-protobuf-serializer</artifactId>
<packaging>jar</packaging>
<name>kafka-protobuf-serializer</name>
```

```
<dependencies>
  <dependency>
    <groupId>org.apache.kafka</groupId>
    <artifactId>kafka_${kafka.scala.version}</artifactId>
    <scope>provided</scope>
  </dependency>
  <dependency>
    <groupId>io.confluent</groupId>
    <artifactId>kafka-protobuf-provider</artifactId>
  </dependency>
  <dependency>
    <groupId>com.google.protobuf</groupId>
    <artifactId>protobuf-java-util</artifactId>
  </dependency>
  <dependency>
    <groupId>io.confluent</groupId>
    <artifactId>kafka-schema-serializer</artifactId>
  </dependency>
  <dependency>
    <groupId>io.confluent</groupId>
    <artifactId>kafka-schema-registry-client</artifactId>
  </dependency>
  <dependency>
    <groupId>io.confluent</groupId>
    <artifactId>kafka-schema-registry</artifactId>
    <scope>test</scope>
  </dependency>
```

```

<dependency>
  <groupId>io.confluent</groupId>
  <artifactId>kafka-schema-registry</artifactId>
  <type>test-jar</type>
  <scope>test</scope>
</dependency>
<dependency>
  <groupId>org.apache.kafka</groupId>
  <artifactId>connect-api</artifactId>
  <scope>provided</scope>
</dependency>
<dependency>
  <groupId>org.apache.kafka</groupId>
  <artifactId>kafka-clients</artifactId>
  <classifier>test</classifier>
  <scope>test</scope>
</dependency>
<dependency>
  <groupId>org.apache.kafka</groupId>
  <artifactId>kafka_${kafka.scala.version}</artifactId>
  <classifier>test</classifier>
  <scope>test</scope>
</dependency>
<dependency>
  <groupId>org.mockito</groupId>
  <artifactId>mockito-core</artifactId>
  <scope>test</scope>
</dependency>
<dependency>
  <groupId>junit</groupId>
  <artifactId>junit</artifactId>
  <scope>test</scope>
</dependency>
</dependencies>

```

```

<build>
  <plugins>
    <plugin>
      <groupId>com.github.os72</groupId>
      <artifactId>protoc-jar-maven-plugin</artifactId>
    </plugin>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-jar-plugin</artifactId>
      <version>2.6</version>
      <executions>
        <execution>
          <goals>

```

```
<goal>test-jar</goal>
</goals>
<phase>test-compile</phase>
</execution>
</executions>
</plugin>
</plugins>
</build>
</project>
```

Found in path(s):

* /opt/cola/permits/1340031763_1654757661.098579/0/kafka-protobuf-serializer-5-5-5-jar/META-INF/maven/io.confluent/kafka-protobuf-serializer/pom.xml

1.102 apache-commons-cli 1.4

1.102.1 Available under license :

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text

file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and

description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

1.103 utils 5.5.5

1.103.1 Available under license :

MIT License

Copyright (c) 2016 MobX

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is

furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.104 appdynamics-java-agent-api

4.5.13.27526

1.104.1 Available under license :

No license file was found, but licenses were detected in source scan.

<div class="aboutLanguage">Copyright 2019 AppDynamics Inc. All rights reserved.</div>

Found in path(s):

- * /opt/cola/permits/1337070300_1654792344.40951/0/java-agent-api-4-5-13-27526-zip/agent-api-javadoc-jar/overview-tree.html
- * /opt/cola/permits/1337070300_1654792344.40951/0/java-agent-api-4-5-13-27526-zip/agent-api-javadoc-jar/com/appdynamics/agent/api/package-tree.html
- * /opt/cola/permits/1337070300_1654792344.40951/0/java-agent-api-4-5-13-27526-zip/agent-api-javadoc-jar/com/appdynamics/agent/api/ExitCall.html
- * /opt/cola/permits/1337070300_1654792344.40951/0/java-agent-api-4-5-13-27526-zip/agent-api-javadoc-jar/deprecated-list.html
- * /opt/cola/permits/1337070300_1654792344.40951/0/java-agent-api-4-5-13-27526-zip/agent-api-javadoc-jar/com/appdynamics/agent/api/impl/NoOpTransaction.html
- * /opt/cola/permits/1337070300_1654792344.40951/0/java-agent-api-4-5-13-27526-zip/agent-api-javadoc-jar/com/appdynamics/agent/api/Transaction.html
- * /opt/cola/permits/1337070300_1654792344.40951/0/java-agent-api-4-5-13-27526-zip/agent-api-javadoc-jar/com/appdynamics/instrumentation/sdk/logging/package-tree.html
- * /opt/cola/permits/1337070300_1654792344.40951/0/java-agent-api-4-5-13-27526-zip/agent-api-javadoc-jar/com/appdynamics/instrumentation/sdk/logging/package-summary.html
- * /opt/cola/permits/1337070300_1654792344.40951/0/java-agent-api-4-5-13-27526-zip/agent-api-javadoc-jar/com/appdynamics/agent/api/AppdynamicsAgent.html
- * /opt/cola/permits/1337070300_1654792344.40951/0/java-agent-api-4-5-13-27526-zip/agent-api-javadoc-jar/com/appdynamics/instrumentation/sdk/logging/ISDKLogger.html
- * /opt/cola/permits/1337070300_1654792344.40951/0/java-agent-api-4-5-13-27526-zip/agent-api-javadoc-jar/com/appdynamics/agent/api/EventPublisher.html
- * /opt/cola/permits/1337070300_1654792344.40951/0/java-agent-api-4-5-13-27526-zip/agent-api-javadoc-jar/com/appdynamics/agent/api/impl/package-summary.html

* /opt/cola/permits/1337070300_1654792344.40951/0/java-agent-api-4-5-13-27526-zip/agent-api-javadoc-jar/com/appdynamics/agent/api/EntryTypes.html
* /opt/cola/permits/1337070300_1654792344.40951/0/java-agent-api-4-5-13-27526-zip/agent-api-javadoc-jar/overview-summary.html
* /opt/cola/permits/1337070300_1654792344.40951/0/java-agent-api-4-5-13-27526-zip/agent-api-javadoc-jar/constant-values.html
* /opt/cola/permits/1337070300_1654792344.40951/0/java-agent-api-4-5-13-27526-zip/agent-api-javadoc-jar/com/appdynamics/agent/api/package-summary.html
* /opt/cola/permits/1337070300_1654792344.40951/0/java-agent-api-4-5-13-27526-zip/agent-api-javadoc-jar/index-all.html
* /opt/cola/permits/1337070300_1654792344.40951/0/java-agent-api-4-5-13-27526-zip/agent-api-javadoc-jar/com/appdynamics/agent/api/ExitTypes.html
* /opt/cola/permits/1337070300_1654792344.40951/0/java-agent-api-4-5-13-27526-zip/agent-api-javadoc-jar/help-doc.html
* /opt/cola/permits/1337070300_1654792344.40951/0/java-agent-api-4-5-13-27526-zip/agent-api-javadoc-jar/com/appdynamics/agent/api/impl/NoOpExitCall.html
* /opt/cola/permits/1337070300_1654792344.40951/0/java-agent-api-4-5-13-27526-zip/agent-api-javadoc-jar/com/appdynamics/agent/api/impl/package-tree.html
* /opt/cola/permits/1337070300_1654792344.40951/0/java-agent-api-4-5-13-27526-zip/agent-api-javadoc-jar/com/appdynamics/agent/api/MetricPublisher.html
No license file was found, but licenses were detected in source scan.

/*

* Copyright (c) AppDynamics, Inc., and its affiliates
* 2018
* All Rights Reserved
* THIS IS UNPUBLISHED PROPRIETARY CODE OF APPDYNAMICS, INC.
* The copyright notice above does not evidence any actual or intended publication of such source code
*/

Found in path(s):

* /opt/cola/permits/1337070300_1654792344.40951/0/java-agent-api-4-5-13-27526-zip/agent-api-sources-jar/com/appdynamics/agent/api/impl/NoOpTransaction.java
* /opt/cola/permits/1337070300_1654792344.40951/0/java-agent-api-4-5-13-27526-zip/agent-api-sources-jar/com/appdynamics/agent/api/EumDelegate.java
* /opt/cola/permits/1337070300_1654792344.40951/0/java-agent-api-4-5-13-27526-zip/agent-api-sources-jar/com/appdynamics/agent/api/impl/NoOpExitCall.java
* /opt/cola/permits/1337070300_1654792344.40951/0/java-agent-api-4-5-13-27526-zip/agent-api-sources-jar/com/appdynamics/agent/api/AppdynamicsAgent.java
* /opt/cola/permits/1337070300_1654792344.40951/0/java-agent-api-4-5-13-27526-zip/agent-api-sources-jar/com/appdynamics/agent/api/ExitCall.java
* /opt/cola/permits/1337070300_1654792344.40951/0/java-agent-api-4-5-13-27526-zip/agent-api-sources-jar/com/appdynamics/agent/api/EventPublisher.java
* /opt/cola/permits/1337070300_1654792344.40951/0/java-agent-api-4-5-13-27526-zip/agent-api-sources-jar/com/appdynamics/agent/api/MetricPublisher.java
* /opt/cola/permits/1337070300_1654792344.40951/0/java-agent-api-4-5-13-27526-zip/agent-api-sources-jar/com/appdynamics/agent/api/Transaction.java

No license file was found, but licenses were detected in source scan.

```
/*  
* Copyright (c) AppDynamics, Inc., and its affiliates  
* 2019  
* All Rights Reserved  
* THIS IS UNPUBLISHED PROPRIETARY CODE OF APPDYNAMICS, INC.  
* The copyright notice above does not evidence any actual or intended publication of such source code  
*/
```

Found in path(s):

```
* /opt/cola/permits/1337070300_1654792344.40951/0/java-agent-api-4-5-13-27526-zip/agent-api-sources-jar/com/appdynamics/agent/api/bootstrap/NoOpTransactionDelegate.java  
* /opt/cola/permits/1337070300_1654792344.40951/0/java-agent-api-4-5-13-27526-zip/agent-api-sources-jar/com/appdynamics/agent/api/ExitTypes.java  
* /opt/cola/permits/1337070300_1654792344.40951/0/java-agent-api-4-5-13-27526-zip/agent-api-sources-jar/com/appdynamics/agent/api/EntryTypes.java  
* /opt/cola/permits/1337070300_1654792344.40951/0/java-agent-api-4-5-13-27526-zip/agent-api-sources-jar/com/appdynamics/agent/api/bootstrap/IApiBootstrapFacade.java  
* /opt/cola/permits/1337070300_1654792344.40951/0/java-agent-api-4-5-13-27526-zip/agent-api-sources-jar/com/appdynamics/agent/api/bootstrap/IApiTransactionDelegate.java
```

1.105 jackson-xc 2.14.0

1.105.1 Available under license :

Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library.

It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has been in development since 2007.

It is currently developed by a community of developers, as well as supported commercially by FasterXML.com.

Licensing

Jackson core and extension components may be licensed under different licenses.

To find the details that apply to this artifact see the accompanying LICENSE file.

For more information, including possible other licensing options, contact FasterXML.com (<http://fasterxml.com>).

Credits

A list of contributors may be found from CREDITS file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.

This copy of Jackson JSON processor `jackson-module-jaxb-annotations` module is licensed under the Apache (Software) License, version 2.0 ("the License").

See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

<http://www.apache.org/licenses/LICENSE-2.0>

1.106 jetbrains-annotations 13.0

1.106.1 Available under license :

Copyright 2000-2012 JetBrains s.r.o.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.107 io-grpc-grpc-context 1.47.0

1.107.1 Available under license :

Envoy

Copyright The Envoy Project Authors

Licensed under Apache License 2.0. See LICENSE for terms.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity

on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one

of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a

result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner].

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

protoc-gen-validate

Copyright 2019 Envoy Project Authors

Licensed under Apache License 2.0. See LICENSE for terms.

Licensed under Apache License 2.0. See LICENSE for terms.

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain

separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the

origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

/*

* Copyright 2015 The gRPC Authors

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Copyright 2014 The gRPC Authors

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

This product contains a modified portion of 'OkHttp', an open source
HTTP & SPDY client for Android and Java applications, which can be obtained
at:

- * LICENSE:
 - * [okhttp/third_party/okhttp/LICENSE](#) (Apache License 2.0)
- * HOMEPAGE:
 - * <https://github.com/square/okhttp>
- * LOCATION_IN_GRPC:
 - * [okhttp/third_party/okhttp](#)

This product contains a modified portion of 'Envoy', an open source cloud-native high-performance edge/middle/service proxy, which can be obtained at:

- * LICENSE:
 - * [xds/third_party/envoy/LICENSE](#) (Apache License 2.0)
- * NOTICE:
 - * [xds/third_party/envoy/NOTICE](#)
- * HOMEPAGE:
 - * <https://www.envoyproxy.io>
- * LOCATION_IN_GRPC:
 - * [xds/third_party/envoy](#)

This product contains a modified portion of 'protoc-gen-validate (PGV)', an open source protoc plugin to generate polyglot message validators, which can be obtained at:

- * LICENSE:
 - * [xds/third_party/protoc-gen-validate/LICENSE](#) (Apache License 2.0)
- * NOTICE:
 - * [xds/third_party/protoc-gen-validate/NOTICE](#)
- * HOMEPAGE:
 - * <https://github.com/envoyproxy/protoc-gen-validate>
- * LOCATION_IN_GRPC:
 - * [xds/third_party/protoc-gen-validate](#)

This product contains a modified portion of 'udpa', an open source universal data plane API, which can be obtained at:

- * LICENSE:
 - * [xds/third_party/udpa/LICENSE](#) (Apache License 2.0)
- * HOMEPAGE:
 - * <https://github.com/cncf/udpa>
- * LOCATION_IN_GRPC:
 - * [xds/third_party/udpa](#)

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and

attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the

appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.108 io-swagger-swagger-annotations 2.2.2

1.108.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate

as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify

the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include

the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright (c) 2015. SmartBear Software Inc.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Swagger Core - swagger-annotations

Copyright (c) 2015. SmartBear Software Inc.

Swagger Core - swagger-annotations is licensed under Apache 2.0 license.

Copy of the Apache 2.0 license can be found in `LICENSE` file.

1.109 cloudevents---api 2.2.0

1.109.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2018-Present The CloudEvents Authors
 * <p>
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * <p>
 * http://www.apache.org/licenses/LICENSE-2.0
 * <p>
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 *
 */
```

Found in path(s):

* /opt/cola/permits/1340815985_1654861270.8175628/0/cloudevents-api-2-2-0-sources-jar/io/cloudevents/lang/Nullable.java
* /opt/cola/permits/1340815985_1654861270.8175628/0/cloudevents-api-2-2-0-sources-jar/io/cloudevents/rw/CloudEventContextReader.java
* /opt/cola/permits/1340815985_1654861270.8175628/0/cloudevents-api-2-2-0-sources-jar/io/cloudevents/rw/CloudEventContextWriter.java
* /opt/cola/permits/1340815985_1654861270.8175628/0/cloudevents-api-2-2-0-sources-jar/io/cloudevents/rw/CloudEventRWException.java
* /opt/cola/permits/1340815985_1654861270.8175628/0/cloudevents-api-2-2-0-sources-jar/io/cloudevents/CloudEvent.java
* /opt/cola/permits/1340815985_1654861270.8175628/0/cloudevents-api-2-2-0-sources-jar/io/cloudevents/rw/CloudEventReader.java
* /opt/cola/permits/1340815985_1654861270.8175628/0/cloudevents-api-2-2-0-sources-jar/io/cloudevents/rw/CloudEventDataMapper.java
* /opt/cola/permits/1340815985_1654861270.8175628/0/cloudevents-api-2-2-0-sources-jar/io/cloudevents/CloudEventExtension.java
* /opt/cola/permits/1340815985_1654861270.8175628/0/cloudevents-api-2-2-0-sources-jar/io/cloudevents/CloudEventContext.java
* /opt/cola/permits/1340815985_1654861270.8175628/0/cloudevents-api-2-2-0-sources-jar/io/cloudevents/rw/CloudEventWriterFactory.java
* /opt/cola/permits/1340815985_1654861270.8175628/0/cloudevents-api-2-2-0-sources-jar/io/cloudevents/CloudEventAttributes.java
* /opt/cola/permits/1340815985_1654861270.8175628/0/cloudevents-api-2-2-0-sources-jar/io/cloudevents/types/Time.java
* /opt/cola/permits/1340815985_1654861270.8175628/0/cloudevents-api-2-2-0-sources-jar/io/cloudevents/CloudEventData.java
* /opt/cola/permits/1340815985_1654861270.8175628/0/cloudevents-api-2-2-0-sources-jar/io/cloudevents/SpecVersion.java
* /opt/cola/permits/1340815985_1654861270.8175628/0/cloudevents-api-2-2-0-sources-jar/io/cloudevents/rw/CloudEventWriter.java
* /opt/cola/permits/1340815985_1654861270.8175628/0/cloudevents-api-2-2-0-sources-jar/io/cloudevents/CloudEventExtensions.java

No license file was found, but licenses were detected in source scan.

<!--

~ Copyright 2018-Present The CloudEvents Authors

~ <p>

~ Licensed under the Apache License, Version 2.0 (the "License");

~ you may not use this file except in compliance with the License.

~ You may obtain a copy of the License at

~ <p>

~ <http://www.apache.org/licenses/LICENSE-2.0>

~ <p>

~ Unless required by applicable law or agreed to in writing, software

~ distributed under the License is distributed on an "AS IS" BASIS,

~ WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

~ See the License for the specific language governing permissions and

~ limitations under the License.

~
-->

Found in path(s):

* /opt/cola/permits/1340815985_1654861270.8175628/0/cloudevents-api-2-2-0-sources-jar/META-INF/maven/io.cloudevents/cloudevents-api/pom.xml

1.110 slf4j-api-module 1.7.26

1.110.1 Available under license :

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.111 asm 9.1

1.111.1 Available under license :

No license file was found, but licenses were detected in source scan.

2011 INRIA, France Telecom

* All rights reserved.

*

* Redistribution and use in source and binary forms, with or without

* modification, are permitted provided that the following conditions

* are met:

* 1. Redistributions of source code must retain the above copyright

* notice, this list of conditions and the following disclaimer.

* 2. Redistributions in binary form must reproduce the above copyright

* notice, this list of conditions and the following disclaimer in the

* documentation and/or other materials provided with the distribution.

* 3. Neither the name of the copyright holders nor the names of its

* contributors may be used to endorse or promote products derived from

* this software without specific prior written permission.

*

* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"

* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.

Found in path(s):

* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/signature/package.html

* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/package.html

No license file was found, but licenses were detected in source scan.

// All rights reserved.

// Redistribution and use in source and binary forms, with or without

// modification, are permitted provided that the following conditions

// are met:

// 1. Redistributions of source code must retain the above copyright

// notice, this list of conditions and the following disclaimer.

// 2. Redistributions in binary form must reproduce the above copyright

// notice, this list of conditions and the following disclaimer in the

// documentation and/or other materials provided with the distribution.

// 3. Neither the name of the copyright holders nor the names of its

// this software without specific prior written permission.

Found in path(s):

* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/TypePath.java

* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/Frame.java

* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-

jar/org/objectweb/asm/RecordComponentWriter.java

* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-

jar/org/objectweb/asm/AnnotationVisitor.java

* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/ModuleVisitor.java

* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/ModuleWriter.java

* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/CurrentFrame.java

* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/AnnotationWriter.java

* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/ConstantDynamic.java

* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/ClassWriter.java

* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-

jar/org/objectweb/asm/signature/SignatureVisitor.java

* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/Attribute.java

* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/Opcodes.java

* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/Handle.java

* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-

```

jar/org/objectweb/asm/MethodTooLargeException.java
* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-
jar/org/objectweb/asm/signature/SignatureReader.java
* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/TypeReference.java
* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/Context.java
* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/Label.java
* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/Type.java
* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-
jar/org/objectweb/asm/signature/SignatureWriter.java
* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/FieldVisitor.java
* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/Handler.java
* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/SymbolTable.java
* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/ClassReader.java
* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/Edge.java
* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-
jar/org/objectweb/asm/RecordComponentVisitor.java
* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/Constants.java
* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/Symbol.java
* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/FieldWriter.java
* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/MethodWriter.java
* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/MethodVisitor.java
* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-
jar/org/objectweb/asm/ClassTooLargeException.java
* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/ByteVector.java
* /opt/cola/permits/1150153141_1627681655.27/0/asm-9-1-sources-1-jar/org/objectweb/asm/ClassVisitor.java

```

1.112 metrics-utility-servlets 4.0.5

1.112.1 Available under license :

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0

Bnd-LastModified: 1545938238563

Build-Jdk: 1.8.0_191

Built-By: artem

Bundle-Description: A set of utility servlets for Metrics, allowing you to expose valuable information about your production environment.

Bundle-License: <http://www.apache.org/licenses/LICENSE-2.0.html>

Bundle-ManifestVersion: 2

Bundle-Name: Metrics Utility Servlets

Bundle-SymbolicName: io.dropwizard.metrics.servlets

Bundle-Version: 4.0.5

Created-By: Apache Maven Bundle Plugin

Export-Package: com.codahale.metrics.servlets;uses:="com.codahale.metrics,com.codahale.metrics.health,javax.servlet,javax.servlet.http";version="4.0.5"

Implementation-Title: Metrics Utility Servlets
Implementation-URL: <http://metrics.dropwizard.io/metrics-servlets>
Implementation-Vendor-Id: io.dropwizard.metrics
Implementation-Version: 4.0.5
Import-Package: javax.servlet;version="[2.5.0,4.0.0)",javax.servlet.http;version="[2.5.0,4.0.0)",com.codahale.metrics;version="[4.0,5)",com.codahale.metrics.health;version="[4.0,5)",com.codahale.metrics.json;version="[4.0,5)",com.codahale.metrics.jvm;version="[4.0,5)",com.fasterxml.jackson.databind;version="[2.9,3)",com.fasterxml.jackson.databind.util;version="[2.9,3)",com.papertrail.profiler,org.joda.time;version="[2.9,3)"
Require-Capability: osgi.ee;filter="(&(osgi.ee=JavaSE)(version=1.8))"
Tool: Bnd-3.3.0.201609221906

Found in path(s):

* /opt/cola/permits/1274701310_1648835822.95/0/metrics-servlets-4-0-5-jar/META-INF/MANIFEST.MF

1.113 jackson-datatype-joda 2.14.0

1.113.1 Available under license :

This copy of Jackson JSON processor streaming parser/generator is licensed under the Apache (Software) License, version 2.0 ("the License").

See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

<http://www.apache.org/licenses/LICENSE-2.0>

1.114 tdigestsketch 0.17.0

1.114.1 Available under license :

Apache Druid

Copyright 2011-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems,

and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and

limitations under the License.

1.115 feign-jackson 8.18.0

1.115.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2013 Netflix, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1274699876_1645234767.83/0/feign-jackson-8-18-0-sources-
jar/feign/jackson/JacksonEncoder.java
* /opt/cola/permits/1274699876_1645234767.83/0/feign-jackson-8-18-0-sources-
jar/feign/jackson/JacksonDecoder.java
```

1.116 apache-groovy 4.0.1

1.116.1 Available under license :

This directory contains generated LICENSE files and snippets used to generate those files. See the assemble.gradle file (updateLicenses task) for details on how this is done. Snippets have predefined suffix values in their name to determine which files they go into. LICENSE (the one for source), LICENSE-DOC and LICENSE-JARJAR get snippets containing SRC, DOC and JARJAR respectively. LICENSE-BINZIP gets JARJAR and BINZIP snippets. In addition, LICENSE files are generated for these subprojects: groovy-docgenerator, groovy-groovydoc, groovy-groovysh, groovy-jsr223
Copyright (c) Nicolas Gallagher and Jonathan Neal

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies

of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a

cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,

any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

JSR223 License

The following classes within this product:

org.codehaus.groovy.jsr223.GroovyCompiledScript
org.codehaus.groovy.jsr223.GroovyScriptEngineFactory
org.codehaus.groovy.jsr223.GroovyScriptEngineImpl

were derived from reference implementation files developed by Sun in collaboration with the Groovy community. The reference implementation has a BSD-style license. Details can be found in: licenses/jsr223-license.txt

Apache Commons Lang

Copyright 2001-2015 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

This product includes software from the Spring Framework,
under the Apache License 2.0 (see: `StringUtils.containsWhitespace()`)
Copyright jQuery Foundation and other contributors, <https://jquery.org/>

Permission is hereby granted, free of charge, to any person obtaining

a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
XStream License (optional dependency when serializing AST as XML)

This product bundles the XStream jar, which is available under a "3-clause BSD" license. For details, see licenses/xstream-license.txt.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,

including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf

of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

ANTLR 4 License

Antlr4 is released under a BSD 3-clause license.
See <licenses/antlr4-license.txt> for details.

ASM License

ASM uses a 3-clause BSD license. For details, see <licenses/asm-license.txt>.
Apache Groovy
Copyright 2003-2022 The Apache Software Foundation

This product includes software developed at

The Apache Software Foundation (<http://www.apache.org/>).

This product includes/uses ANTLR4 (<https://github.com/antlr/antlr4>)
Copyright (c) 2012-2017 The ANTLR Project. All rights reserved.
ANTLR 4 License

[The "BSD 3-clause license"]

Copyright (c) 2012-2017 The ANTLR Project. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
NORMALIZE.CSS LICENSE

The following file is used with documentation:

`org/codehaus/groovy/tools/stylesheet.css`

Copyright (c) Nicolas Gallagher and Jonathan Neal

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all

copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate

as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify

the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include

the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

ANTLR 4 License

Antlr4 is released under a BSD 3-clause license.
See licenses/antlr4-license.txt for details.

ASM License

ASM uses a 3-clause BSD license. For details, see licenses/asm-license.txt.

Hamcrest License (needed when using optional JUnit dependency)

This product bundles the Hamcrest jar, which is available under a BSD license. For details, see licenses/hamcrest-license.txt.

JLine2 License (optional dependency used with groovysh)

This product bundles the JLine2 jar, which is available under a BSD License. For details, see licenses/jline2-license.txt.

JSR166y License (optionally used by the optional GPar dependency)

This product bundles the jsr166y jar (containing works from the JSR-166 EG, Doug Lea, and Jason T. Greene) made available in the public domain. For details, see [licenses/jsr166y-license.txt](#).

JSR223 License

The following classes within this product:

```
org.codehaus.groovy.jsr223.GroovyCompiledScript
org.codehaus.groovy.jsr223.GroovyScriptEngineFactory
org.codehaus.groovy.jsr223.GroovyScriptEngineImpl
```

were derived from reference implementation files developed by Sun in collaboration with the Groovy community. The reference implementation has a BSD-style license. Details can be found in: [licenses/jsr223-license.txt](#)

JUnit Licenses (optional dependencies when using Groovy for testing)

This product bundles the JUnit 4 jar, which is available under the Eclipse Public License v1.0. For details, see [licenses/junit4-license.txt](#).

This product bundles several JUnit 5 jars, which are available under the Eclipse Public License v2.0. For details, see [licenses/junit5-license.txt](#).

normalize.css License

The stylesheet.css file (originally normalize.css) is used by the groovydoc and docgenerator components for groovy-jdk/gapi documentation. It is made available under a MIT License:
[licenses/normalize-stylesheet-license.txt](#)

XStream License (optional dependency when serializing AST as XML)

This product bundles the XStream jar, which is available under a "3-clause BSD" license. For details, see [licenses/xstream-license.txt](#).

This convenience zip embeds Groovy's src and doc zips.
See also src/LICENSE and doc/LICENSE files for additional license information.

////////////////////////////////////

Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to you under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

////////////////////////////////////

= License

This work is licensed under the <http://www.apache.org/licenses/LICENSE-2.0> [Apache License, Version 2.0].

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or

otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents

of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

AsciiDoc License

This product uses the style.css from asciidoctor.org within documentation. The file is available under the MIT License. For details, see licenses/asciidoc-style-license.txt.

JQuery License

The following file is used within documentation:

`src/spec/assets/css/jquery-2.1.1.min.js`

This file is made available under the following MIT license:

`licenses/jquery-js-license.txt`

JSR223 License

The following classes within this product:

`org.codehaus.groovy.jsr223.GroovyCompiledScript`
`org.codehaus.groovy.jsr223.GroovyScriptEngineFactory`
`org.codehaus.groovy.jsr223.GroovyScriptEngineImpl`

were derived from reference implementation files developed by Sun in collaboration with the Groovy community. The reference implementation has a BSD-style license. Details can be found in: `licenses/jsr223-license.txt`

normalize.css License

The `stylesheet.css` file (originally `normalize.css`) is used by the `groovydoc` and `docgenerator` components for `groovy-jdk/gapi` documentation.

It is made available under a MIT License:

`licenses/normalize-stylesheet-license.txt`

`normalize.css` License

The `stylesheet.css` file (originally `normalize.css`) is used by the `groovydoc` and `docgenerator` components for `groovy-jdk/gapi` documentation.

It is made available under a MIT License:

`licenses/normalize-stylesheet-license.txt`

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and

b) in the case of each subsequent Contributor:

i) changes to the Program, and

ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations

which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:

i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and

iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and

b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

JSR223 License

The following classes within this product:

```
org.codehaus.groovy.jsr223.GroovyCompiledScript
org.codehaus.groovy.jsr223.GroovyScriptEngineFactory
org.codehaus.groovy.jsr223.GroovyScriptEngineImpl
```

were derived from reference implementation files developed by Sun in collaboration with the Groovy community. The reference implementation has a BSD-style license. Details can be found in: licenses/jsr223-license.txt

Revised BSD license

This is a specific instance of the Open Source Initiative (OSI) BSD license template
<http://www.opensource.org/licenses/bsd-license.php>

Copyright 2004-2009 Brent Fulgham
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of "The Computer Language Benchmarks Game" nor the name of "The Computer Language Shootout Benchmarks" nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

JSR166y License (optionally used by the optional GPar dependency)

This product bundles the jsr166y jar (containing works from the JSR-166 EG, Doug Lea, and Jason T. Greene) made available in the public domain. For details, see [licenses/jsr166y-license.txt](#).

JLine2 License (optional dependency used with groovysh)

This product bundles the JLine2 jar, which is available under a BSD License. For details, see [licenses/jline2-license.txt](#).

ANTLR 4 License

Antlr4 is released under a BSD 3-clause license.

See [licenses/antlr4-license.txt](#) for details.

Apache Groovy

Copyright 2003-2022 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

The Java source files in `src/main/java/org/apache/groovy/util/concurrent/concurrentlinkedhashmap/` are from <https://github.com/ben-manes/concurrentlinkedhashmap> and the following notice applies:
Copyright 2010-2012 Google Inc. All Rights Reserved.

The Java source file `src/main/java/org/apache/groovy/util/concurrent/ConcurrentReferenceHashMap` is from <https://github.com/hazelcast/hazelcast> and the following notice applies:
Copyright (c) 2008-2020, Hazelcast, Inc. All Rights Reserved.

This product bundles icons from the famfamfam.com silk icons set

<http://www.famfamfam.com/lab/icons/silk/>

Licensed under the Creative Commons Attribution Licence v2.5

<http://creativecommons.org/licenses/by/2.5/>

Eclipse Public License - v 2.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE (AGREEMENT). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. Definitions

Contribution means:

- a) in the case of the initial Contributor, the initial content Distributed under this Agreement, and
- b) in the case of each subsequent Contributor:
 - i) changes to the Program, and
 - ii) additions to the Program; where such changes and/or additions to the Program originate from and are Distributed by that particular Contributor. A Contribution originates from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include changes or additions to the Program that are not Modified Works.

Contributor means any person or entity that Distributes the Program.

Licensed Patents mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

Program means the Contributions Distributed in accordance with this Agreement.

Recipient means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

Derivative Works shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

Modified Works shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

Distribute means the acts of a) distributing or b) making available in any manner that enables the transfer of a copy.

Source Code means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

Secondary License means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.

2. Grant of Rights

- a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.
- b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.
- c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.
- d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.
- e) Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

3. Requirements

3.1 If a Contributor Distributes the Program in any form, then:

- a) the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and
- b) the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:
- i) effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
 - ii) effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
 - iii) does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and
 - iv) requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.

3.2 When the Program is Distributed as Source Code:

- a) it must be made available under this Agreement, or if the Program (i) is combined with other material in a

separate file or files made available under a Secondary License, and (ii) the initial Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available under the terms of such Secondary Licenses, and

b) a copy of this Agreement must be included with each copy of the Program.

3.3 Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability (notices) contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

4. Commercial Distribution

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor (Commercial Contributor) hereby agrees to defend and indemnify every other Contributor (Indemnified Contributor) against any losses, damages and costs (collectively Losses) arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. No Warranty

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN AS IS BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. Disclaimer of Liability

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF

SUCH DAMAGES.

7. General

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.

Exhibit A - Form of Secondary Licenses Notice

This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: {name license(s), version(s), and exceptions or additional permissions here}.

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Apache License

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions

to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices

stating that You changed the files; and

- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

- 5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
- 6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
- 7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

ANTLR 4 License

Antlr4 is released under a BSD 3-clause license.
See [licenses/antlr4-license.txt](#) for details.

ASM License

ASM uses a 3-clause BSD license. For details, see [licenses/asm-license.txt](#).

Hamcrest License (needed when using optional JUnit dependency)

This product bundles the Hamcrest jar, which is available under a BSD license. For details, see [licenses/hamcrest-license.txt](#).

JLine2 License (optional dependency used with groovysh)

This product bundles the JLine2 jar, which is available under a BSD License. For details, see [licenses/jline2-license.txt](#).

JSR166y License (optionally used by the optional GPar dependency)

This product bundles the jsr166y jar (containing works from the JSR-166 EG, Doug Lea, and Jason T. Greene) made available in the public domain. For details, see [licenses/jsr166y-license.txt](#).

JSR223 License

The following classes within this product:

```
org.codehaus.groovy.jsr223.GroovyCompiledScript
org.codehaus.groovy.jsr223.GroovyScriptEngineFactory
org.codehaus.groovy.jsr223.GroovyScriptEngineImpl
```

were derived from reference implementation files developed by Sun in collaboration with the Groovy community. The reference implementation has a BSD-style license. Details can be found in: [licenses/jsr223-license.txt](#)

JUnit Licenses (optional dependencies when using Groovy for testing)

This product bundles the JUnit 4 jar, which is available under the Eclipse Public License v1.0. For details, see [licenses/junit4-license.txt](#).

This product bundles several JUnit 5 jars, which are available under the Eclipse Public License v2.0. For details, see [licenses/junit5-license.txt](#).

normalize.css License

The stylesheet.css file (originally normalize.css) is used by the groovydoc and docgenerator components for groovy-jdk/gapi documentation. It is made available under a MIT License: [licenses/normalize-stylesheet-license.txt](#)

XStream License (optional dependency when serializing AST as XML)

This product bundles the XStream jar, which is available under a "3-clause BSD" license. For details, see [licenses/xstream-license.txt](#).
COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

1. Definitions.

1.1. Contributor means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing

Modifications, in each case including portions thereof.

1.4. Executable means the Covered Software in any form other than Source Code.

1.5. Initial Developer means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License means this document.

1.8. Licensable means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications means the Source Code and Executable form of any of the following: A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications; B. Any new file that contains any part of the Original Software or previous Modification; or C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You (or Your) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant. Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof);

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License;

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant. Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code. Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications. The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices. You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms. You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions. You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipients rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works. You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions. Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions. You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions. When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY. COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY. UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTYS NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS. The Covered Software is a commercial item, as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of commercial computer software (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer software documentation as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS. This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be

unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdictions conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS. As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

For Covered Software in this distribution, this License shall be governed by the laws of the State of California (excluding conflict-of-law provisions).

Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.
BSD License

Copyright (c) 2000-2015 www.hamcrest.org
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of Hamcrest nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written

permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The person or persons who have associated work with this document (the "Dedicator" or "Certifier") hereby either (a) certifies that, to the best of his knowledge, the work of authorship identified is in the public domain of the country from which the work is published, or (b) hereby dedicates whatever copyright the dedicator holds in the work of authorship identified below (the "Work") to the public domain. A certifier, moreover, dedicates any copyright interest he may have in the associated work, and for these purposes, is described as a "dedicator" below.

A certifier has taken reasonable steps to verify the copyright status of this work. Certifier recognizes that his good faith efforts may not shield him from liability if in fact the work certified is not in the public domain.

Dedicator makes this dedication for the benefit of the public at large and to the detriment of the Dedicator's heirs and successors. Dedicator intends this dedication to be an overt act of relinquishment in perpetuity of all present and future rights under copyright law, whether vested or contingent, in the Work. Dedicator understands that such relinquishment of all rights includes the relinquishment of all rights to enforce (by lawsuit or otherwise) those copyrights in the Work.

Dedicator recognizes that, once placed in the public domain, the Work may be freely reproduced, distributed, transmitted, used, modified, built upon, or otherwise exploited by anyone for any purpose, commercial or non-commercial, and in any way, including by methods that have not yet been invented or conceived.

Apache Groovy

Copyright 2003-2022 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

This product embeds the OpenBeans jar within its grooid jar artifacts
OpenBeans includes/uses files from Apache Harmony and the following notice applies
Copyright 2006, 2010 The Apache Software Foundation.
Portions of Apache Harmony were originally developed by Intel Corporation and are
licensed to the Apache Software Foundation under the "Software Grant and Corporate
Contribution License Agreement" and for which the following copyright notices apply
(C) Copyright 2005 Intel Corporation
(C) Copyright 2005-2006 Intel Corporation
(C) Copyright 2006 Intel Corporation
Apache Groovy
Copyright 2003-2022 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

This product includes/uses ANTLR4 (<https://github.com/antlr/antlr4>)
Copyright (c) 2012-2017 The ANTLR Project. All rights reserved.

This product bundles the JUnit4 jar (junit.org)
which is available under the terms of the Eclipse Public License v1.0

This product bundles several of the JUnit5 jars (junit.org)
which are available under the terms of the Eclipse Public License v2.0

This product embeds the OpenBeans jar within its grooid jar artifacts
OpenBeans includes/uses files from Apache Harmony and the following notice applies
Copyright 2006, 2010 The Apache Software Foundation.
Portions of Apache Harmony were originally developed by Intel Corporation and are
licensed to the Apache Software Foundation under the "Software Grant and Corporate
Contribution License Agreement" and for which the following copyright notices apply
(C) Copyright 2005 Intel Corporation
(C) Copyright 2005-2006 Intel Corporation
(C) Copyright 2006 Intel Corporation

This product bundles icons from the famfamfam.com silk icons set
<http://www.famfamfam.com/lab/icons/silk/>
Licensed under the Creative Commons Attribution Licence v2.5
<http://creativecommons.org/licenses/by/2.5/>

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems,

and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and

limitations under the License.

NORMALIZE.CSS LICENSE

The following file is used with documentation:

`org/codehaus/groovy/tools/stylesheet.css`

Copyright (c) Nicolas Gallagher and Jonathan Neal

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache Groovy

Copyright 2003-2022 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

This product includes/uses ANTLR4 (<https://github.com/antlr/antlr4>)
Copyright (c) 2012-2017 The ANTLR Project. All rights reserved.

This product bundles the JUnit4 jar (junit.org)
which is available under the terms of the Eclipse Public License v1.0

This product bundles several of the JUnit5 jars (junit.org)
which are available under the terms of the Eclipse Public License v2.0

This product embeds the OpenBeans jar within its grooid jar artifacts
OpenBeans includes/uses files from Apache Harmony and the following notice applies
Copyright 2006, 2010 The Apache Software Foundation.

Portions of Apache Harmony were originally developed by Intel Corporation and are

licensed to the Apache Software Foundation under the "Software Grant and Corporate Contribution License Agreement" and for which the following copyright notices apply

(C) Copyright 2005 Intel Corporation

(C) Copyright 2005-2006 Intel Corporation

(C) Copyright 2006 Intel Corporation

This product bundles icons from the famfamfam.com silk icons set

<http://www.famfamfam.com/lab/icons/silk/>

Licensed under the Creative Commons Attribution Licence v2.5

<http://creativecommons.org/licenses/by/2.5/>

This convenience zip embeds Groovy's src and doc zips.

See also src/NOTICE and doc/NOTICE files for additional notice information.

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a

cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,

any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

AsciiDoc License

This product uses the style.css from asciidoctor.org within documentation. The file is available under the MIT License. For details, see licenses/asciidoc-style-license.txt.

JQuery License

The following file is used within documentation:

src/spec/assets/css/jquery-2.1.1.min.js

This file is made available under the following MIT license:
licenses/jquery-js-license.txt

normalize.css License

The stylesheet.css file (originally normalize.css) is used by the

groovydoc and docgenerator components for groovy-jdk/gapi documentation.

It is made available under a MIT License:

licenses/normalize-stylesheet-license.txt

ASM License

ASM: a very small and fast Java bytecode manipulation framework

Copyright (c) 2000-2011 INRIA, France Telecom

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The MIT License

Copyright (C) 2012-2015 Dan Allen, Ryan Waldron and the AsciiDoctor Project

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2003-2006, Joe Walnes

Copyright (c) 2006-2009, 2011 XStream Committers

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of XStream nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache Groovy

Copyright 2003-2022 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

This product bundles icons from the famfamfam.com silk icons set

<http://www.famfamfam.com/lab/icons/silk/>
Licensed under the Creative Commons Attribution Licence v2.5
<http://creativecommons.org/licenses/by/2.5/>
Apache Groovy
Copyright 2003-2022 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

This product includes/uses ANTLR4 (<https://github.com/antlr/antlr4>)
Copyright (c) 2012-2017 The ANTLR Project. All rights reserved.

This product embeds the OpenBeans jar within its grooid jar artifacts
OpenBeans includes/uses files from Apache Harmony and the following notice applies
Copyright 2006, 2010 The Apache Software Foundation.

Portions of Apache Harmony were originally developed by Intel Corporation and are
licensed to the Apache Software Foundation under the "Software Grant and Corporate
Contribution License Agreement" and for which the following copyright notices apply

(C) Copyright 2005 Intel Corporation

(C) Copyright 2005-2006 Intel Corporation

(C) Copyright 2006 Intel Corporation

Hamcrest License (needed when using optional JUnit dependency)

This product bundles the Hamcrest jar, which is available under a
BSD license. For details, see [licenses/hamcrest-license.txt](#).

NORMALIZE.CSS LICENSE

The following file is used with documentation:

[org/codehaus/groovy/tools/groovydoc/gstringTemplates/topLevel/styleSheet.css](#)

Copyright (c) Nicolas Gallagher and Jonathan Neal

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
of the Software, and to permit persons to whom the Software is furnished to do
so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,

OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

JUnit Licenses (optional dependencies when using Groovy for testing)

This product bundles the JUnit 4 jar, which is available under the Eclipse Public License v1.0. For details, see licenses/junit4-license.txt.

This product bundles several JUnit 5 jars, which are available under the Eclipse Public License v2.0. For details, see licenses/junit5-license.txt.

JQuery License

The following file is used within documentation:

src/spec/assets/css/jquery-2.1.1.min.js

This file is made available under the following MIT license:

licenses/jquery-js-license.txt

////////////////////////////////////

Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to you under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

////////////////////////////////////

= Contributors

The Groovy team would like to thank the contributors of this documentation (in alphabetical order of last/surname):

- * <https://github.com/mojavelinux>[Dan Allen]
- * <https://github.com/and-dmitry>[Dmitry Andreychuk]
- * <http://hamletdarcy.blogspot.fr/>[Hamlet D'Arcy]
- * <https://github.com/anshbansal>[Aseem Bansal]
- * <https://github.com/bura>[Andrey Bloshchetsov]
- * <https://github.com/JBrownVisualSpection>[J Brown]

- * <https://github.com/jeffbrown>[Jeff Scott Brown]
- * <http://twitter.com/CedricChampeau>[Cdric Champeau]
- * <https://github.com/tobia>[Tobia Conforto]
- * <https://github.com/ddimitrov>[Dimitar Dimitrov]
- * <http://twitter.com/werdnagreb>[Andrew Eisenberg]
- * <https://github.com/erdi>[Marcin Erdmann]
- * <https://github.com/christoph-frick>[Christoph Frick]
- * <http://twitter.com/marioggar>[Mario Garca]
- * <https://github.com/davidmichaelkarr>[David Michael Karr]
- * http://twitter.com/paulk_asert[Paul King]
- * <http://twitter.com/glaforge>[Guillaume Laforge]
- * <http://twitter.com/pledbrook>[Peter Ledbrook]
- * <http://grantmconnaughey.github.io/>[Grant McConnaughey]
- * <https://github.com/eric-milles>[Eric Milles]
- * <https://github.com/dnahodil>[David Nahodil]
- * <https://github.com/jnorthr>[James Northrop]
- * <https://github.com/marcpa00>[Marc Paquette]
- * <https://github.com/michaelss>[Michael Schuenck]
- * <https://github.com/PascalSchumacher>[Pascal Schumacher]
- * <https://github.com/shils>[Shil Sinha]
- * <https://github.com/stavytskyi>[Maksym Stavytskyi]
- * <https://twitter.com/asteingr>[Andr Steingre]
- * https://twitter.com/daniel_sun[Daniel Sun]
- * <https://github.com/EPadronU>[Edinson Padrn Urdaneta]
- * <https://github.com/keeganwitt>[Keegan Witt]

ASM License

ASM uses a 3-clause BSD license. For details, see [licenses/asm-license.txt](#).

Asciidoc License

This product uses the [style.css](#) from [asciidoc.org](#) within documentation. The file is available under the MIT License.

For details, see [licenses/asciidoc-style-license.txt](#).

Copyright (c) 2002-2012, the original author or authors.

All rights reserved.

<http://www.opensource.org/licenses/bsd-license.php>

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with

the distribution.

Neither the name of JLine nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made,

use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability

incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

NORMALIZE.CSS LICENSE

The following file is used with documentation:

org/codehaus/groovy/tools/groovydoc/gstringTemplates/topLevel/stylesheet.css

Copyright (c) Nicolas Gallagher and Jonathan Neal

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all

copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache Groovy

Copyright 2003-2022 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).
Copyright (c) 2006, Sun Microsystems, Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the Sun Microsystems, Inc. nor the names of contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions

to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices

stating that You changed the files; and

- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

- 5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
- 6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
- 7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.117 micronaut-kotlin-integrations 3.2.2

1.117.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2017-2020 original authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1331473908_1653514800.6790595/0/micronaut-ktor-3-2-2-sources-
jar/io/micronaut/ktor/KtorApplicationBuilder.kt
* /opt/cola/permits/1331473908_1653514800.6790595/0/micronaut-ktor-3-2-2-sources-
jar/io/micronaut/ktor/KtorApplication.kt
* /opt/cola/permits/1331473908_1653514800.6790595/0/micronaut-ktor-3-2-2-sources-
jar/io/micronaut/ktor/server/AbstractKtorEmbeddedServer.kt
* /opt/cola/permits/1331473908_1653514800.6790595/0/micronaut-ktor-3-2-2-sources-
jar/io/micronaut/ktor/KtorRoutingBuilder.kt
* /opt/cola/permits/1331473908_1653514800.6790595/0/micronaut-ktor-3-2-2-sources-
jar/io/micronaut/ktor/server/KtorTomcatEmbeddedServer.kt
* /opt/cola/permits/1331473908_1653514800.6790595/0/micronaut-ktor-3-2-2-sources-
jar/io/micronaut/ktor/env/MicronautKtorEnvironmentConfig.kt
* /opt/cola/permits/1331473908_1653514800.6790595/0/micronaut-ktor-3-2-2-sources-
jar/io/micronaut/ktor/server/KtorNettyEmbeddedServer.kt
* /opt/cola/permits/1331473908_1653514800.6790595/0/micronaut-ktor-3-2-2-sources-
jar/io/micronaut/ktor/factory/MicronautKtorApplicationFactory.kt
```

* /opt/cola/permits/1331473908_1653514800.6790595/0/micronaut-ktor-3-2-2-sources-jar/io/micronaut/ktor/server/KtorJettyEmbeddedServer.kt

1.118 scala 2.13.10

1.118.1 Available under license :

Scala includes the JLine library, which includes the JNA library, which is made available under multiple licenses, including the Apache 2 license:

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work

(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses

granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]"

replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

not-a-legal-formal-parameter-tuple.scala:2: error: not a legal formal parameter.

Note: Tuples cannot be directly destructured in method or function parameters.

```
Either create a single parameter accepting the Tuple2,  
or consider a pattern matching anonymous function: `{ case (a, b) => ... }`  
val x: ((Int, Int) => Int) = (((a, b)) => a)  
      ^
```

not-a-legal-formal-parameter-tuple.scala:3: error: not a legal formal parameter.

Note: Tuples cannot be directly destructured in method or function parameters.

```
Either create a single parameter accepting the Tuple2,  
or consider a pattern matching anonymous function: `{ case (param1, param2) => ... }`  
val y: ((Int, Int, Int) => Int) = (((a, !!)) => a)  
      ^
```

not-a-legal-formal-parameter-tuple.scala:4: error: not a legal formal parameter.

Note: Tuples cannot be directly destructured in method or function parameters.

```
Either create a single parameter accepting the Tuple3,  
or consider a pattern matching anonymous function: `{ case (param1, ..., param3) => ... }`  
val z: ((Int, Int, Int) => Int) = (((a, NotAPatternVariableName, c)) => a)  
      ^
```

3 errors

Scala is licensed under the [Apache License Version 2.0](<https://www.apache.org/licenses/LICENSE-2.0>).

Scala License

Copyright (c) 2002-2022 EPFL

Copyright (c) 2011-2022 Lightbend, Inc.

All rights reserved.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Other Licenses

This software includes projects with the following licenses,
which are also included in the `licenses/`` directory:

[Apache License](<http://www.apache.org/licenses/LICENSE-2.0.html>)

This license is used by the following third-party libraries:

- * JNA

[BSD 3-Clause License](<http://opensource.org/licenses/BSD-3-Clause>)

This license is used by the following third-party libraries:

- * ASM
- * JLine 3

[MIT License](<http://www.opensource.org/licenses/MIT>)

This license is used by the following third-party libraries:

- * jQuery

Scala includes the jQuery library:

Copyright JS Foundation and other contributors, <https://js.foundation/>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Scala

Copyright (c) 2002-2022 EPFL

Copyright (c) 2011-2022 Lightbend, Inc.

Scala includes software developed at
LAMP/EPFL (<https://lamp.epfl.ch/>) and
Lightbend, Inc. (<https://www.lightbend.com/>).

Licensed under the Apache License, Version 2.0 (the "License").
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

This software includes projects with other licenses -- see `doc/LICENSE.md`.
(c) 2012-2014 GitHub

When using the GitHub logos, be sure to follow the GitHub logo guidelines (<https://github.com/logos>)

Font License: SIL OFL 1.1 (<http://scripts.sil.org/OFL>)

Applies to all font files

Code License: MIT (<http://choosealicense.com/licenses/mit/>)

Applies to all other files

Copyright (c) 2006, Ivan Sagalaev

All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of highlight.js nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED

WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS AND CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```
{% if site.thisScalaVersion != site.latestScalaVersion %}
```

```
<div class="version-notice">This is the specification of {{ site.versionCompareMessage }} version of Scala. See the <a href="{{ site.baseurl }}/../{{ site.latestScalaVersion }}">Scala {{ site.latestScalaVersion }} spec</a>.</div>
```

```
{% endif %}
```

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a

copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct

or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of

this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following

boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Scala includes the ASM library.

Copyright (c) 2000-2011 INRIA, France Telecom
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holders nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS

INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Scala includes the JLine 3 library:

Copyright (c) 2002-2018, the original author or authors.

All rights reserved.

<https://opensource.org/licenses/BSD-3-Clause>

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of JLine nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

(The MIT License)

Copyright (c) 2013 Greg Allen

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to

the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.119 micronaut-grpc 3.3.1

1.119.1 Available under license :

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You

institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.120 micronaut-data 3.8.1

1.120.1 Available under license :

mcr.microsoft.com/mssql/server:2017-CU12

Apache License

Version 2.0, January 2004

<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the

direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of

this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and

wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor

has been advised of the possibility of such damages.

9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.121 jackson-jaxrs-base 2.14.0

1.121.1 Available under license :

This copy of Jackson JSON processor databind module is licensed under the Apache (Software) License, version 2.0 ("the License").

See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

<http://www.apache.org/licenses/LICENSE-2.0>

1.122 open-feign-forms-core 3.8.0

1.122.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2019 the original author or authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1131826771_1613019813.73/0/feign-form-3-8-0-sources-
jar/feign/form/multipart/PojoWriter.java
* /opt/cola/permits/1131826771_1613019813.73/0/feign-form-3-8-0-sources-jar/feign/form/multipart/Writer.java
* /opt/cola/permits/1131826771_1613019813.73/0/feign-form-3-8-0-sources-
jar/feign/form/multipart/ByteArrayWriter.java
* /opt/cola/permits/1131826771_1613019813.73/0/feign-form-3-8-0-sources-jar/feign/form/ContentProcessor.java
* /opt/cola/permits/1131826771_1613019813.73/0/feign-form-3-8-0-sources-jar/feign/form/util/PojoUtil.java
* /opt/cola/permits/1131826771_1613019813.73/0/feign-form-3-8-0-sources-jar/feign/form/multipart/Output.java
* /opt/cola/permits/1131826771_1613019813.73/0/feign-form-3-8-0-sources-jar/feign/form/FormProperty.java
* /opt/cola/permits/1131826771_1613019813.73/0/feign-form-3-8-0-sources-
jar/feign/form/multipart/FormDataReader.java
* /opt/cola/permits/1131826771_1613019813.73/0/feign-form-3-8-0-sources-
jar/feign/form/multipart/AbstractWriter.java
* /opt/cola/permits/1131826771_1613019813.73/0/feign-form-3-8-0-sources-jar/feign/form/ContentType.java
* /opt/cola/permits/1131826771_1613019813.73/0/feign-form-3-8-0-sources-
jar/feign/form/multipart/ManyParametersWriter.java
* /opt/cola/permits/1131826771_1613019813.73/0/feign-form-3-8-0-sources-jar/feign/form/util/CharsetUtil.java
* /opt/cola/permits/1131826771_1613019813.73/0/feign-form-3-8-0-sources-jar/feign/form/FormEncoder.java
* /opt/cola/permits/1131826771_1613019813.73/0/feign-form-3-8-0-sources-jar/feign/form/FormData.java
* /opt/cola/permits/1131826771_1613019813.73/0/feign-form-3-8-0-sources-
```

jar/feign/form/MultipartFormContentProcessor.java
* /opt/cola/permits/1131826771_1613019813.73/0/feign-form-3-8-0-sources-jar/feign/form/multipart/DelegateWriter.java
* /opt/cola/permits/1131826771_1613019813.73/0/feign-form-3-8-0-sources-jar/feign/form/multipart/SingleParameterWriter.java
* /opt/cola/permits/1131826771_1613019813.73/0/feign-form-3-8-0-sources-jar/feign/form/multipart/SingleFileWriter.java
* /opt/cola/permits/1131826771_1613019813.73/0/feign-form-3-8-0-sources-jar/feign/form/UrlencodedFormContentProcessor.java
* /opt/cola/permits/1131826771_1613019813.73/0/feign-form-3-8-0-sources-jar/feign/form/multipart/ManyFilesWriter.java
No license file was found, but licenses were detected in source scan.

Copyright 2019 the original author or authors.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Found in path(s):
* /opt/cola/permits/1131826771_1613019813.73/0/feign-form-3-8-0-sources-jar/META-INF/maven/io.github.openfeign.form/feign-form/pom.xml

1.123 wire-schema 3.6.0

1.123.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*  
* Copyright (C) 2014 Square, Inc.  
*  
* Licensed under the Apache License, Version 2.0 (the "License");  
* you may not use this file except in compliance with the License.  
* You may obtain a copy of the License at  
*  
* http://www.apache.org/licenses/LICENSE-2.0  
*  
* Unless required by applicable law or agreed to in writing, software
```

- * distributed under the License is distributed on an "AS IS" BASIS,
- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.
- */

Found in path(s):

- * /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-jar/commonMain/com/squareup/wire/schema/internal/parser/FieldElement.kt
- * /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-jar/commonMain/com/squareup/wire/schema/internal/parser/OneOfElement.kt
- * /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-jar/commonMain/com/squareup/wire/schema/internal/parser/EnumConstantElement.kt
- * /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-jar/commonMain/com/squareup/wire/schema/internal/parser/RpcElement.kt

No license file was found, but licenses were detected in source scan.

/*

- * Copyright (C) 2017 Square, Inc.
- *
- * Licensed under the Apache License, Version 2.0 (the "License");
- * you may not use this file except in compliance with the License.
- * You may obtain a copy of the License at
- *
- * <http://www.apache.org/licenses/LICENSE-2.0>
- *
- * Unless required by applicable law or agreed to in writing, software
- * distributed under the License is distributed on an "AS IS" BASIS,
- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.
- */

Found in path(s):

- * /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-jar/commonMain/com/squareup/wire/schema/internal/parser/OptionReader.kt

No license file was found, but licenses were detected in source scan.

/*

- * Copyright (C) 2019 Square, Inc.
- *
- * Licensed under the Apache License, Version 2.0 (the "License");
- * you may not use this file except in compliance with the License.
- * You may obtain a copy of the License at
- *
- * <http://www.apache.org/licenses/LICENSE-2.0>
- *
- * Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-jar/commonMain/com/squareup/wire/schema/Loader.kt
* /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-jar/jvmMain/com/squareup/wire/schema/internal/UtilJVM.kt
* /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-jar/commonMain/com/squareup/wire/schema/FileLinker.kt
* /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-jar/jvmMain/com/squareup/wire/schema/CoreLoader.kt
* /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-jar/commonMain/com/squareup/wire/schema/SemVer.kt

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2016 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-jar/commonMain/com/squareup/wire/schema/internal/parser/SyntaxReader.kt
* /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-jar/commonMain/com/squareup/wire/schema/EnclosingType.kt
* /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-jar/commonMain/com/squareup/wire/schema/Reserved.kt
* /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-jar/commonMain/com/squareup/wire/schema/internal/parser/ReservedElement.kt
* /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-jar/commonMain/com/squareup/wire/schema/internal/parser/GroupElement.kt

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2013 Square, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
 * /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-
jar/commonMain/com/squareup/wire/schema/internal/parser/EnumElement.kt
 * /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-
jar/commonMain/com/squareup/wire/schema/internal/parser/ServiceElement.kt
 * /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-
jar/commonMain/com/squareup/wire/schema/internal/parser/ProtoParser.kt
 * /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-
jar/commonMain/com/squareup/wire/schema/internal/parser/ProtoFileElement.kt
 * /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-
jar/commonMain/com/squareup/wire/schema/internal/parser/TypeElement.kt
 * /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-
jar/commonMain/com/squareup/wire/schema/internal/parser/ExtendElement.kt
 * /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-
jar/commonMain/com/squareup/wire/schema/internal/parser/ExtensionsElement.kt
 * /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-
jar/commonMain/com/squareup/wire/schema/internal/Util.kt
 * /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-
jar/commonMain/com/squareup/wire/schema/internal/parser/OptionElement.kt
 * /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-
jar/commonMain/com/squareup/wire/schema/internal/parser/MessageElement.kt
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2015 Square, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
```

- * Unless required by applicable law or agreed to in writing, software
- * distributed under the License is distributed on an "AS IS" BASIS,
- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.
- */

Found in path(s):

- * /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-jar/commonMain/com/squareup/wire/schema/MessageType.kt
- * /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-jar/commonMain/com/squareup/wire/schema/MarkSet.kt
- * /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-jar/commonMain/com/squareup/wire/schema/PruningRules.kt
- * /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-jar/commonMain/com/squareup/wire/schema/ProtoMember.kt
- * /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-jar/commonMain/com/squareup/wire/schema/Rpc.kt
- * /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-jar/commonMain/com/squareup/wire/schema/OneOf.kt
- * /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-jar/commonMain/com/squareup/wire/schema/Pruner.kt
- * /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-jar/commonMain/com/squareup/wire/schema/Extend.kt
- * /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-jar/commonMain/com/squareup/wire/schema/Options.kt
- * /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-jar/commonMain/com/squareup/wire/schema/ProtoType.kt
- * /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-jar/commonMain/com/squareup/wire/schema/Field.kt
- * /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-jar/commonMain/com/squareup/wire/schema/EnumType.kt
- * /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-jar/commonMain/com/squareup/wire/schema/EnumConstant.kt
- * /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-jar/commonMain/com/squareup/wire/schema/Location.kt
- * /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-jar/commonMain/com/squareup/wire/schema/ProtoFile.kt
- * /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-jar/commonMain/com/squareup/wire/schema/Extensions.kt
- * /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-jar/commonMain/com/squareup/wire/schema/Schema.kt
- * /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-jar/commonMain/com/squareup/wire/schema/Service.kt
- * /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-jar/commonMain/com/squareup/wire/schema/Linker.kt
- * /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-jar/commonMain/com/squareup/wire/schema/SchemaException.kt

* /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-jar/commonMain/com/squareup/wire/schema/SchemaProtoAdapterFactory.kt
* /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-jar/commonMain/com/squareup/wire/schema/Type.kt
No license file was found, but licenses were detected in source scan.

```
/*  
* Copyright (C) 2020 Square, Inc.  
*  
* Licensed under the Apache License, Version 2.0 (the "License");  
* you may not use this file except in compliance with the License.  
* You may obtain a copy of the License at  
*  
* http://www.apache.org/licenses/LICENSE-2.0  
*  
* Unless required by applicable law or agreed to in writing, software  
* distributed under the License is distributed on an "AS IS" BASIS,  
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
* See the License for the specific language governing permissions and  
* limitations under the License.  
*/
```

Found in path(s):

* /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-jar/jvmMain/com/squareup/wire/schema/EmittingRules.kt
* /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-jar/jvmMain/com/squareup/wire/schema/ClaimedDefinitions.kt
* /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-jar/commonMain/com/squareup/wire/schema/CoreLoader.kt
* /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-jar/jvmMain/com/squareup/wire/schema/Multimap.kt
* /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-jar/commonMain/com/squareup/wire/schema/SyntaxRules.kt
* /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-jar/commonMain/com/squareup/wire/schema/ErrorCollector.kt
* /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-jar/commonMain/com/squareup/wire/schema/Multimap.kt
* /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-jar/jvmMain/com/squareup/wire/schema/internal/JvmLanguages.kt
* /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-jar/commonMain/com/squareup/wire/schema/internal/DagChecker.kt
* /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-jar/commonMain/com/squareup/wire/schema/internal/TypeMover.kt
* /opt/cola/permits/1340031715_1654618469.2355554/0/wire-schema-3-6-0-sources-1-jar/commonMain/com/squareup/wire/schema/CycleChecker.kt

1.124 swagger-core v2.2.3

1.124.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes

of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You

meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor,

except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright (c) 2015. SmartBear Software Inc.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Swagger Core - \${pom.name}

Copyright (c) 2015. SmartBear Software Inc.

Swagger Core - \${pom.name} is licensed under Apache 2.0 license.

Copy of the Apache 2.0 license can be found in `LICENSE` file.

1.125 reactive-streams v1.0.3

1.125.1 Available under license :

Copyright Statement for Contributions to the Reactive Streams Project

=====

I hereby represent that all present, past and future contributions I make to
the Reactive Streams project (which includes all repositories owned by the
reactive-streams github organization) are governed by the Creative Commons
Zero 1.0 Universal copyright statement, placing my contributions in the public
domain. This entails that to the extent possible under law I waive all
copyright and related or neighboring rights to the code or documents I
contribute. I also represent that I have the authority to perform the above
waiver with respect to the entirety of my contributions.

The text of the copyright statement is included in the COPYING file at the root
of the reactive-streams repository at
<https://github.com/reactive-streams/reactive-streams-jvm/blob/master/COPYING>.

Underwriting parties:

github name | Real Name, Email Address used for git commits, Company

-----+-----
rkuhn | Roland Kuhn, rk@rkuhn.info, Typesafe Inc.

benjchristensen| Ben Christensen, benjchristensen@gmail.com, Netflix Inc.

viktorklang | Viktor Klang, viktorklang@gmail.com, Typesafe Inc.

smaldini | Stephane Maldini, stephane.maldini@gmail.com, Pivotal Software Inc.

savulchik | Stanislav Savulchik, s.savulchik@gmail.com

ktoso | Konrad Malawski, konrad.malawski@project13.pl, Typesafe Inc.
ouertani | Slim Ouertani, ouertani@gmail.com
2m | Martynas Mickevicius, mmartynas@gmail.com, Typesafe Inc.
ldaley | Luke Daley, luke.daley@gradleware.com, Gradleware Inc.
colingodsey | Colin Godsey, crgodsey@gmail.com, MediaMath Inc.
davidmoten | Dave Moten, davidmoten@gmail.com
briantopping | Brian Topping, brian.topping@gmail.com, Mauswerks LLC
rstoyanchev | Rossen Stoyanchev, rstoyanchev@pivotal.io, Pivotal
BjornHamels | Bjrn Hamels, bjorn@hamels.nl
JakeWharton | Jake Wharton, jakewharton@gmail.com
anthonyvdotbe | Anthony Vanelverdinghe, anthonyv.be@outlook.com
seratch | Kazuhiro Sera, seratch@gmail.com, SmartNews, Inc.
akarnokd | David Karnok, akarnokd@gmail.com
egetman | Evgeniy Getman, getman.eugene@gmail.com
patriknw | Patrik Nordwall, patrik.nordwall@gmail.com, Lightbend Inc
angelsanz | ngel Sanz, angelsanz@users.noreply.github.com
shenghaiyang | , shenghaiyang@aliyun.com
kiiadi | Kyle Thomson, kylthoms@amazon.com, Amazon.com
jroper | James Roper, james@jazzy.id.au, Lightbend Inc.
olegdokuka | Oleh Dokuka, shadowgun@i.ua, Netifi Inc.
Scottmitch | Scott Mitchell, scott_mitchell@apple.com, Apple Inc.
retronym | Jason Zaugg, jzaugg@gmail.com, Lightbend Inc.
Creative Commons Legal Code

CC0 1.0 Universal

CREATIVE COMMONS CORPORATION IS NOT A LAW FIRM AND DOES NOT PROVIDE LEGAL SERVICES. DISTRIBUTION OF THIS DOCUMENT DOES NOT CREATE AN ATTORNEY-CLIENT RELATIONSHIP. CREATIVE COMMONS PROVIDES THIS INFORMATION ON AN "AS-IS" BASIS. CREATIVE COMMONS MAKES NO WARRANTIES REGARDING THE USE OF THIS DOCUMENT OR THE INFORMATION OR WORKS PROVIDED HEREUNDER, AND DISCLAIMS LIABILITY FOR DAMAGES RESULTING FROM THE USE OF THIS DOCUMENT OR THE INFORMATION OR WORKS PROVIDED HEREUNDER.

Statement of Purpose

The laws of most jurisdictions throughout the world automatically confer exclusive Copyright and Related Rights (defined below) upon the creator and subsequent owner(s) (each and all, an "owner") of an original work of authorship and/or a database (each, a "Work").

Certain owners wish to permanently relinquish those rights to a Work for the purpose of contributing to a commons of creative, cultural and scientific works ("Commons") that the public can reliably and without fear of later claims of infringement build upon, modify, incorporate in other works, reuse and redistribute as freely as possible in any form whatsoever and for any purposes, including without limitation commercial purposes.

These owners may contribute to the Commons to promote the ideal of a free culture and the further production of creative, cultural and scientific works, or to gain reputation or greater distribution for their Work in part through the use and efforts of others.

For these and/or other purposes and motivations, and without any expectation of additional consideration or compensation, the person associating CC0 with a Work (the "Affirmer"), to the extent that he or she is an owner of Copyright and Related Rights in the Work, voluntarily elects to apply CC0 to the Work and publicly distribute the Work under its terms, with knowledge of his or her Copyright and Related Rights in the Work and the meaning and intended legal effect of CC0 on those rights.

1. Copyright and Related Rights. A Work made available under CC0 may be protected by copyright and related or neighboring rights ("Copyright and Related Rights"). Copyright and Related Rights include, but are not limited to, the following:

- i. the right to reproduce, adapt, distribute, perform, display, communicate, and translate a Work;
- ii. moral rights retained by the original author(s) and/or performer(s);
- iii. publicity and privacy rights pertaining to a person's image or likeness depicted in a Work;
- iv. rights protecting against unfair competition in regards to a Work, subject to the limitations in paragraph 4(a), below;
- v. rights protecting the extraction, dissemination, use and reuse of data in a Work;
- vi. database rights (such as those arising under Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, and under any national implementation thereof, including any amended or successor version of such directive); and
- vii. other similar, equivalent or corresponding rights throughout the world based on applicable law or treaty, and any national implementations thereof.

2. Waiver. To the greatest extent permitted by, but not in contravention of, applicable law, Affirmer hereby overtly, fully, permanently, irrevocably and unconditionally waives, abandons, and surrenders all of Affirmer's Copyright and Related Rights and associated claims and causes of action, whether now known or unknown (including existing as well as future claims and causes of action), in the Work (i) in all territories worldwide, (ii) for the maximum duration provided by applicable law or treaty (including future time extensions), (iii) in any current or future medium and for any number of copies, and (iv) for any purpose whatsoever, including without limitation commercial, advertising or promotional purposes (the "Waiver"). Affirmer makes the Waiver for the benefit of each member of the public at large and to the detriment of Affirmer's heirs and

successors, fully intending that such Waiver shall not be subject to revocation, rescission, cancellation, termination, or any other legal or equitable action to disrupt the quiet enjoyment of the Work by the public as contemplated by Affirmer's express Statement of Purpose.

3. **Public License Fallback.** Should any part of the Waiver for any reason be judged legally invalid or ineffective under applicable law, then the Waiver shall be preserved to the maximum extent permitted taking into account Affirmer's express Statement of Purpose. In addition, to the extent the Waiver is so judged Affirmer hereby grants to each affected person a royalty-free, non transferable, non sublicensable, non exclusive, irrevocable and unconditional license to exercise Affirmer's Copyright and Related Rights in the Work (i) in all territories worldwide, (ii) for the maximum duration provided by applicable law or treaty (including future time extensions), (iii) in any current or future medium and for any number of copies, and (iv) for any purpose whatsoever, including without limitation commercial, advertising or promotional purposes (the "License"). The License shall be deemed effective as of the date CC0 was applied by Affirmer to the Work. Should any part of the License for any reason be judged legally invalid or ineffective under applicable law, such partial invalidity or ineffectiveness shall not invalidate the remainder of the License, and in such case Affirmer hereby affirms that he or she will not (i) exercise any of his or her remaining Copyright and Related Rights in the Work or (ii) assert any associated claims and causes of action with respect to the Work, in either case contrary to Affirmer's express Statement of Purpose.

4. Limitations and Disclaimers.

- a. No trademark or patent rights held by Affirmer are waived, abandoned, surrendered, licensed or otherwise affected by this document.
- b. Affirmer offers the Work as-is and makes no representations or warranties of any kind concerning the Work, express, implied, statutory or otherwise, including without limitation warranties of title, merchantability, fitness for a particular purpose, non infringement, or the absence of latent or other defects, accuracy, or the present or absence of errors, whether or not discoverable, all to the greatest extent permissible under applicable law.
- c. Affirmer disclaims responsibility for clearing rights of other persons that may apply to the Work or any use thereof, including without limitation any person's Copyright and Related Rights in the Work. Further, Affirmer disclaims responsibility for obtaining any necessary consents, permissions or other rights required for any use of the Work.
- d. Affirmer understands and acknowledges that Creative Commons is not a party to this document and has no duty or obligation with respect to this CC0 or use of the Work.

Licensed under Public Domain (CC0)

To the extent possible under law, the person who associated CC0 with this code has waived all copyright and related or neighboring rights to this code.

You should have received a copy of the CC0 legalcode along with this work. If not, see <<http://creativecommons.org/publicdomain/zero/1.0/>>.

1.126 kotlin-libraries-bill-of-materials 1.6.21

1.126.1 Available under license :

No license file was found, but licenses were detected in source scan.

<url><http://www.apache.org/licenses/LICENSE-2.0.txt></url>

Found in path(s):

* /opt/cola/permits/1343420717_1655246882.684955/0/kotlin-bom-1-6-21-pom-zip/kotlin-bom-1.6.21.pom

1.127 rest-assured 5.1.0

1.127.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2019 the original author or authors.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-jar/io/restassured/specification/PreemptiveAuthSpec.java

* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/util/MatcherErrorMessageBuilder.java

* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-

jar/io/restassured/config/RedirectConfig.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/ValidatableResponseOptionsImpl.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/response/ResponseOptions.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/support/Prettifier.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/TrustAndKeystoreSpec.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/config/SSLConfig.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/config/DecoderConfig.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/matcher/ResponseAwareMatcher.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/ContentParser.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/authentication/BasicAuthScheme.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/filter/session/SessionFilter.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/filter/FilterContextImpl.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/http/HttpResponseException.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/response/ResponseBodyData.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/filter/FormAuthFilter.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/authentication/OAuthSignature.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/RedirectSpecificationImpl.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/support/ParameterUpdater.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/mapper/ObjectMapperSerializationContext.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/http/CharsetExtractor.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/http/ContentTypeExtractor.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/multipart/MultiPartSpecificationImpl.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/http/ResponseParseException.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/PreemptiveAuthSpecImpl.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-

jar/io/restassured/internal/proxy/RestAssuredProxySelector.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/config/MultiPartConfig.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/RestAssuredResponseOptionsImpl.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/http/StringHashMap.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/matcher/RestAssuredMatchers.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/mapper/ObjectMapperDeserializationContext.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/config/MatcherConfig.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/http/DeflateEncoding.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/matcher/xml/LoadFromClasspathSupport.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/NameAndValue.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/http/Cookies.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/LogSpecificationImpl.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/assertion/HeaderMatcher.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/authentication/AuthenticationScheme.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/RestAssuredResponseImpl.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/http/EncoderRegistry.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/util/SafeExceptionRethrower.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/support/PathSupport.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/http/CustomHttpMethod.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/authentication/PreemptiveOAuth2HeaderScheme.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/http/BoundaryExtractor.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/mapping/Jackson2Mapper.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/specification/LogSpecification.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/ResponseSpecificationImpl.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-

jar/io/restassured/internal/mapping/JohnzonMapper.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/assertion/ResponseTimeMatcher.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/serialization/SerializationSupport.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/http/ContentTypeSubTypeExtractor.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/matcher/ResponseAwareMatcherComposer.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/specification/RequestSpecification.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/config/ConnectionConfig.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/config/OAuthConfig.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/mapping/GsonMapper.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/http/HttpRequestFactory.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/specification/AuthenticationSpecification.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/config/RestAssuredConfig.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/log/LogRepository.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/filter/time/TimingFilter.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/mapping/ObjectMapping.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/authentication/NoAuthScheme.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/config/JsonConfig.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/specification/QueryableRequestSpecification.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/config/ObjectMapperConfig.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/authentication/PreemptiveBasicAuthScheme.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/http/ContentEncoding.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/authentication/CertAuthScheme.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/http/HttpResponseDecorator.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/authentication/FormAuthScheme.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-

jar/io/restassured/internal/multipart/RestAssuredMultiPartEntity.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/response/ValidatableResponseOptions.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/specification/MultiPartSpecification.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/filter/log/ResponseLoggingFilter.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/filter/log/StatusCodeBasedLoggingFilter.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/mapping/ObjectMapperSerializationContextImpl.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/authentication/OAuth2Scheme.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/specification/RequestSender.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/MapCreator.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/assertion/DetailedCookieAssertion.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/mapper/ObjectMapper.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/http/Header.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/filter/log/LogDetail.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/print/RequestPrinter.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/http/HttpContextDecorator.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/response/ValidatableResponse.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/filter/log/ErrorLoggingFilter.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/specification/SpecificationQuerier.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/config/Config.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/TestSpecificationImpl.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/MultiValueEntity.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/http/Status.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/response/ValidatableResponseLogSpec.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/support/CloseHttpClientConnectionInputStreamWrapper.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-

jar/io/restassured/mapper/ObjectMapperType.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/listener/ResponseValidationFailureListener.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/LogRequestAndResponseOnFailListener.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/config/EncoderConfig.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/ResponseLogSpecificationImpl.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/filter/SendRequestFilter.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/specification/Argument.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/filter/OrderedFilter.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/mapping/ObjectMapperDeserializationContextImpl.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/http/ContentType.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/specification/RequestSenderOptions.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/TrustAndKeystoreSpecImpl.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/http/URIBuilder.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/authentication/PreemptiveAuthProvider.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/http/Cookie.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/authentication/NTLMAuthScheme.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/config/HeaderConfig.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/matcher/DetailedCookieMatcher.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/assertion/StreamVerifier.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/response/Response.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/SpecificationMerger.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/NoParameterValue.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/authentication/CertificateAuthSettings.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/config/HttpClientConfig.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-

jar/io/restassured/specification/ProxySpecification.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/builder/ResponseBuilder.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/RestAssuredResponseOptionsGroovyImpl.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/filter/FilterContext.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/mapping/Jackson1Mapper.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/filter/Filter.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/filter/log/RequestLoggingFilter.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/response/ResponseBodyExtractionOptions.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/http/ContentEncodingRegistry.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/print/ResponsePrinter.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/builder/MultiPartSpecBuilder.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/config/FailureConfig.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/specification/FilterableRequestSpecification.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/specification/RedirectSpecification.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/authentication/OAuthScheme.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/response/Validatable.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/http/AuthConfig.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/http/HTTPBuilder.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/mapping/JakartaEEMapper.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/proxy/RestAssuredProxySelectorRoutePlanner.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/specification/RequestLogSpecification.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/mapping/JaxbMapper.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/response/ResponseBody.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/authentication/FormAuthConfig.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-

jar/io/restassured/internal/RequestLogSpecificationImpl.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/http/BasicNameValuePairWithNoValueSupport.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/builder/ResponseSpecBuilder.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/http/Headers.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/parsing/Parser.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/authentication/ExplicitNoAuthScheme.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/filter/log/UrlDecoder.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/http/GZIPEncoding.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/multipart/MultiPartInternal.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/http/Method.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/RestAssured.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/builder/RequestSpecBuilder.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/ValidatableResponseImpl.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/http/HttpResponseContentTypeFinder.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/matcher/xml/XmlDtdMatcher.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/filter/cookie/CookieFilter.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/config/SessionConfig.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/specification/FilterableResponseSpecification.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/matcher/xml/XmlXsdMatcher.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/util/IOUtils.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/config/ParamConfig.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/AuthenticationSpecificationImpl.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/specification/ResponseSpecification.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/spi/AuthFilter.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-

jar/io/restassured/internal/support/FileReader.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/response/ExtractableResponse.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/RequestSpecificationImpl.groovy
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/config/LogConfig.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/UriValidator.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/specification/ResponseLogSpecification.java
* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/config/XmlConfig.java
No license file was found, but licenses were detected in source scan.

```
/*  
* Copyright 2022 the original author or authors.  
*  
* Licensed under the Apache License, Version 2.0 (the "License");  
* you may not use this file except in compliance with the License.  
* You may obtain a copy of the License at  
*  
* http://www.apache.org/licenses/LICENSE-2.0  
*  
* Unless required by applicable law or agreed to in writing, software  
* distributed under the License is distributed on an "AS IS" BASIS,  
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
* See the License for the specific language governing permissions and  
* limitations under the License.  
*/
```

Found in path(s):

* /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/RestAssuredHttpBuilder.java
No license file was found, but licenses were detected in source scan.

```
/*  
* Copyright 2020 the original author or authors.  
*  
* Licensed under the Apache License, Version 2.0 (the "License");  
* you may not use this file except in compliance with the License.  
* You may obtain a copy of the License at  
*  
* http://www.apache.org/licenses/LICENSE-2.0  
*  
* Unless required by applicable law or agreed to in writing, software  
* distributed under the License is distributed on an "AS IS" BASIS,  
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
```

- * See the License for the specific language governing permissions and
- * limitations under the License.
- */

Found in path(s):

- * /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/ResponseParserRegistrar.java
- * /opt/cola/permits/1473459965_1668478161.7397234/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/mapping/JsonbMapper.groovy

1.128 jul-to-slf4j-bridge 1.7.26

1.128.1 Available under license :

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.129 apache-httpcomponents-core 4.4.11

1.129.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity

on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one

of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a

result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

Apache HttpComponents Core

Copyright 2005-2019 The Apache Software Foundation

This product includes software developed at

The Apache Software Foundation (<http://www.apache.org/>).

1.130 jackson-annotations 2.14.0

1.130.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or

otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents

of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.131 wire-multiplatform-runtime- (experimental) 3.6.0

1.131.1 Available under license :

Apache License
Version 2.0, January 2004

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally

submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or

implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.132 json-p 1.0.4

1.132.1 Available under license :

Found license 'General Public License 2.0' in '* Copyright (c) 2011-2013 Oracle and/or its affiliates. All rights reserved. * The contents of this file are subject to the terms of either the GNU * General Public License Version 2 only ("GPL") or the Common Development * and Distribution License("CDDL") (collectively, the "License"). You * may not use this file except in compliance with the License. You can * Oracle designates this particular file as subject to the "Classpath" * exception as provided by Oracle in the GPL Version 2 section of the License'

Found license 'General Public License 2.0' in '* Copyright (c) 2013 Oracle and/or its affiliates. All rights reserved. * The contents of this file are subject to the terms of either the GNU * General Public License Version 2 only ("GPL") or the Common Development * and Distribution License("CDDL") (collectively, the "License"). You * may not use this file except in compliance with the License. You can * Oracle designates this particular file as subject to the "Classpath" * exception as provided by Oracle in the GPL Version 2 section of the License'

Found license 'General Public License 2.0' in '* Copyright (c) 2012-2013 Oracle and/or its affiliates. All rights reserved. * The contents of this file are subject to the terms of either the GNU * General Public License Version 2 only ("GPL") or the Common Development * and Distribution License("CDDL") (collectively, the "License"). You * may not use this file except in compliance with the License. You can * Oracle designates this particular file as subject to the "Classpath" * exception as provided by Oracle in the GPL Version 2 section of the License'

Found license 'General Public License 2.0' in 'Copyright (c) 2011-2013 Oracle and/or its affiliates. All rights reserved. The contents of this file are subject to the terms of either the GNU General Public License Version 2 only ("GPL") or the Common Development and Distribution License("CDDL") (collectively, the "License"). You may not use this file except in compliance with the License. You can Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the GPL Version 2 section of the License'

Found license 'Common Development and Distribution License (CDDL) 1.1' in 'Copyright (c) 2011-2013 Oracle and/or its affiliates. All rights reserved. The contents of this file are subject to the terms of either the GNU General Public License Version 2 only ("GPL") or the Common Development and Distribution License("CDDL") (collectively, the "License"). You may not use this file except in compliance with the License. You can Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the GPL Version 2 section of the License'

1.133 apache-httpcomponents-asyncclient

4.1.4

1.133.1 Available under license :

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner

or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions

of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

This project contains annotations derived from JCIP-ANNOTATIONS Copyright (c) 2005 Brian Goetz and Tim Peierls.
See <http://www.jcip.net> and the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/2.5>)
Apache HttpComponents AsyncClient
Copyright 2010-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

1.134 kafka-schema-serializer 5.5.5

1.134.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
<project xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://maven.apache.org/POM/4.0.0"
```

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">

```
<modelVersion>4.0.0</modelVersion>

<parent>
  <groupId>io.confluent</groupId>
  <artifactId>kafka-schema-registry-parent</artifactId>
  <version>5.5.5</version>
</parent>

<licenses>
  <license>
    <name>Apache License 2.0</name>
    <url>http://www.apache.org/licenses/LICENSE-2.0.html</url>
    <distribution>repo</distribution>
  </license>
</licenses>

<artifactId>kafka-schema-serializer</artifactId>
<packaging>jar</packaging>
<name>kafka-schema-serializer</name>

<dependencies>
  <dependency>
    <groupId>org.apache.kafka</groupId>
    <artifactId>kafka_${kafka.scala.version}</artifactId>
    <scope>provided</scope>
  </dependency>
  <dependency>
    <groupId>io.confluent</groupId>
    <artifactId>kafka-schema-registry-client</artifactId>
  </dependency>
  <dependency>
    <groupId>io.confluent</groupId>
    <artifactId>common-config</artifactId>
  </dependency>

  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <scope>test</scope>
  </dependency>
</dependencies>
</project>
```

Found in path(s):

* /opt/cola/permits/1341639782_1654785878.0524044/0/kafka-schema-serializer-5-5-5-jar/META-INF/maven/io.confluent/kafka-schema-serializer/pom.xml

1.135 antlr-4-tool 4.8-1

1.135.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * [The "BSD license"]
 * Copyright (c) 2012-2016 Terence Parr
 * Copyright (c) 2012-2016 Sam Harwell
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * 3. The name of the author may not be used to endorse or promote products
 * derived from this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
 * IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
 * IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
 * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
 * DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
 * THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
 * THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */

/** templates used to generate make-compatible dependencies */

/** Generate "f : x, y, z" dependencies for input
 * dependencies and generated files. in and out
 * are File objects. For example, you can say
 * <f.canonicalPath>
 */
dependencies(grammarFileName,in,out) ::= <<
<if(in)><grammarFileName>: <in; separator=", "><endif>
<out:{ f | <f> : <grammarFileName>}; separator="\n">
>>
```

Found in path(s):

* /opt/cola/permits/1473460068_1668481512.5494933/0/antlr4-4-8-1-jar/org/antlr/v4/tool/templates/depend.stg

No license file was found, but licenses were detected in source scan.

/*

* [The "BSD license"]

* Copyright (c) 2013 Terence Parr

* Copyright (c) 2013 Sam Harwell

* All rights reserved.

*

* Redistribution and use in source and binary forms, with or without

* modification, are permitted provided that the following conditions

* are met:

*

* 1. Redistributions of source code must retain the above copyright

* notice, this list of conditions and the following disclaimer.

* 2. Redistributions in binary form must reproduce the above copyright

* notice, this list of conditions and the following disclaimer in the

* documentation and/or other materials provided with the distribution.

* 3. The name of the author may not be used to endorse or promote products

* derived from this software without specific prior written permission.

*

* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR

* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES

* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.

* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,

* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT

* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,

* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY

* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT

* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF

* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

// args must be <object-model-object>, <fields-resulting-in-STs>

ParserFile(file, parser, namedActions, contextSuperClass) ::= <<

<fileHeader(file.grammarFileName, file.ANTLRVersion)>

<if(file.genPackage)>

namespace <file.genPackage> {

<endif>

<namedActions.header>

using System;

using System.IO;

using System.Text;

using System.Diagnostics;

using System.Collections.Generic;

```

using Antlr4.Runtime;
using Antlr4.Runtime.Atn;
using Antlr4.Runtime.Misc;
using Antlr4.Runtime.Tree;
using DFA = Antlr4.Runtime.Dfa.DFA;

```

```

<parser>
<if(file.genPackage)>
} // namespace <file.genPackage>
<endif>
>>

```

```

ListenerFile(file, header, namedActions) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
<if(file.genPackage)>
namespace <file.genPackage> {
<endif>
<header>
using Antlr4.Runtime.Misc;
using IParseTreeListener = Antlr4.Runtime.Tree.IParseTreeListener;
using IToken = Antlr4.Runtime.IToken;

/// \<summary>
/// This interface defines a complete listener for a parse tree produced by
/// \<see cref="<csIdentifier.(file.parserName)>"/>.
/// \</summary>
[System.CodeDom.Compiler.GeneratedCode("ANTLR", "<file.ANTLRVersion>")]
[System.CLSCompliant(false)]
public interface I<file.grammarName>Listener : IParseTreeListener {
<file.listenerNames: {lname |
/// \<summary>
<if(file.listenerLabelRuleNames.(lname))>
/// Enter a parse tree produced by the \<c><lname>\</c>
/// labeled alternative in \<see cref="<file.parserName>.<file.listenerLabelRuleNames.(lname)>"/>.
<else>
/// Enter a parse tree produced by \<see cref="<file.parserName>.<lname>"/>.
<endif>
/// \</summary>
/// \<param name="context">The parse tree.\</param>
void Enter<lname; format="cap">([NotNull] <csIdentifier.(file.parserName)>.<lname; format="cap">Context
context);
/// \<summary>
<if(file.listenerLabelRuleNames.(lname))>
/// Exit a parse tree produced by the \<c><lname>\</c>
/// labeled alternative in \<see cref="<file.parserName>.<file.listenerLabelRuleNames.(lname)>"/>.
<else>
/// Exit a parse tree produced by \<see cref="<file.parserName>.<lname>"/>.
<endif>

```

```

/// \</summary>
/// \<param name="context">The parse tree.\</param>
void Exit<Iname; format="cap">([NotNull] <csIdentifier.(file.parserName)>.<Iname; format="cap">Context
context);); separator="\n">
}
<if(file.genPackage)>
} // namespace <file.genPackage>
<endif>
>>

```

```

BaseListenerFile(file, header, namedActions) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
<if(file.genPackage)>
namespace <file.genPackage> {
<endif>
<header>

```

```

using Antlr4.Runtime.Misc;
using IErrorNode = Antlr4.Runtime.Tree.IErrorNode;
using ITerminalNode = Antlr4.Runtime.Tree.ITerminalNode;
using IToken = Antlr4.Runtime.IToken;
using ParserRuleContext = Antlr4.Runtime.ParserRuleContext;

```

```

/// \<summary>
/// This class provides an empty implementation of \<see cref="I<file.grammarName>Listener"/>,
/// which can be extended to create a listener which only needs to handle a subset
/// of the available methods.
/// \</summary>
[System.CodeDom.Compiler.GeneratedCode("ANTLR", "<file.ANTLRVersion>")]
[System.CLSCompliant(false)]
public partial class <file.grammarName>BaseListener : I<file.grammarName>Listener {
    <file.listenerNames; {Iname |
    /// \<summary>
    <if(file.listenerLabelRuleNames.(Iname))>
    /// Enter a parse tree produced by the \<c><Iname>\</c>
    /// labeled alternative in \<see cref="<file.parserName>.<file.listenerLabelRuleNames.(Iname)>">.
    <else>
    /// Enter a parse tree produced by \<see cref="<file.parserName>.<Iname>">.
    <endif>
    /// \<para>The default implementation does nothing.\</para>
    /// \</summary>
    /// \<param name="context">The parse tree.\</param>
    public virtual void Enter<Iname; format="cap">([NotNull] <csIdentifier.(file.parserName)>.<Iname;
    format="cap">Context context) { \}
    /// \<summary>
    <if(file.listenerLabelRuleNames.(Iname))>
    /// Exit a parse tree produced by the \<c><Iname>\</c>
    /// labeled alternative in \<see cref="<file.parserName>.<file.listenerLabelRuleNames.(Iname)>">.

```

```

<else>
/// Exit a parse tree produced by \<see cref="<file.parserName>.<lname>"/>.
<endif>
/// \<para>The default implementation does nothing.\</para>
/// \</summary>
/// \<param name="context">The parse tree.\</param>
public virtual void Exit<lname; format="cap">([NotNull] <csIdentifier.(file.parserName)>.<lname;
format="cap">Context context) { \} }; separator="\n">

/// \<inheritdoc/>
/// \<remarks>The default implementation does nothing.\</remarks>
public virtual void EnterEveryRule([NotNull] ParserRuleContext context) { }
/// \<inheritdoc/>
/// \<remarks>The default implementation does nothing.\</remarks>
public virtual void ExitEveryRule([NotNull] ParserRuleContext context) { }
/// \<inheritdoc/>
/// \<remarks>The default implementation does nothing.\</remarks>
public virtual void VisitTerminal([NotNull] ITerminalNode node) { }
/// \<inheritdoc/>
/// \<remarks>The default implementation does nothing.\</remarks>
public virtual void VisitErrorNode([NotNull] IErrorNode node) { }
}
<if(file.genPackage)>
} // namespace <file.genPackage>
<endif>
>>

```

```

VisitorFile(file, header, namedActions) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
<if(file.genPackage)>
namespace <file.genPackage> {
<endif>
<header>
using Antlr4.Runtime.Misc;
using Antlr4.Runtime.Tree;
using IToken = Antlr4.Runtime.IToken;

/// \<summary>
/// This interface defines a complete generic visitor for a parse tree produced
/// by \<see cref="<csIdentifier.(file.parserName)>"/>.
/// \</summary>
/// \<typeparam name="Result">The return type of the visit operation.\</typeparam>
[System.CodeDom.Compiler.GeneratedCode("ANTLR", "<file.ANTLRVersion>")]
[System.CLSCompliant(false)]
public interface I<file.grammarName>Visitor<Result> : IParseTreeVisitor<Result> {
<file.visitorNames>: { lname |
/// \<summary>
<if(file.visitorLabelRuleNames.(lname))>

```

```

/// Visit a parse tree produced by the \<c><lname>\</c>
/// labeled alternative in \<see cref="<file.parserName>.<file.visitorLabelRuleNames.(lname)>">.
<else>
/// Visit a parse tree produced by \<see cref="<file.parserName>.<lname>">.
<endif>
</summary>
/// \<param name="context">The parse tree.\</param>
/// \<return>The visitor result.\</return>
Result Visit<lname; format="cap">([NotNull] <csIdentifier.(file.parserName)>.<lname; format="cap">Context
context);); separator="\n">
}
<if(file.genPackage)>
} // namespace <file.genPackage>
<endif>
>>

```

```

BaseVisitorFile(file, header, namedActions) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
<if(file.genPackage)>
namespace <file.genPackage> {
<endif>
<header>
using Antlr4.Runtime.Misc;
using Antlr4.Runtime.Tree;
using IToken = Antlr4.Runtime.IToken;
using ParserRuleContext = Antlr4.Runtime.ParserRuleContext;

/// \<summary>
/// This class provides an empty implementation of \<see cref="I<file.grammarName>Visitor{Result}">,
/// which can be extended to create a visitor which only needs to handle a subset
/// of the available methods.
/// \</summary>
/// \<typeparam name="Result">The return type of the visit operation.\</typeparam>
[System.CodeDom.Compiler.GeneratedCode("ANTLR", "<file.ANTLRVersion>")]
[System.CLSCompliant(false)]
public partial class <file.grammarName>BaseVisitor<Result> : AbstractParseTreeVisitor<Result>,
I<file.grammarName>Visitor<Result> {
    <file.visitorNames>{ lname |
/// \<summary>
<if(file.visitorLabelRuleNames.(lname))>
/// Visit a parse tree produced by the \<c><lname>\</c>
/// labeled alternative in \<see cref="<file.parserName>.<file.visitorLabelRuleNames.(lname)>">.
<else>
/// Visit a parse tree produced by \<see cref="<file.parserName>.<lname>">.
<endif>
/// \<para>
/// The default implementation returns the result of calling \<see
cref="AbstractParseTreeVisitor{Result}.VisitChildren(IRuleNode)">

```

```

/// on \<paramref name="context"/>.
/// \</para>
/// \</summary>
/// \<param name="context">The parse tree.\</param>
/// \<return>The visitor result.\</return>
public virtual Result Visit<Iname; format="cap">([NotNull] <csIdentifier.(file.parserName)>.<Iname;
format="cap">Context context) { return VisitChildren(context); \}}; separator="\n">
}
<if(file.genPackage)>
} // namespace <file.genPackage>
<endif>
>>

```

```

fileHeader(grammarFileName, ANTLRVersion) ::= <<
//-----
// \<auto-generated>
// This code was generated by a tool.
// ANTLR Version: <ANTLRVersion>
//
// Changes to this file may cause incorrect behavior and will be lost if
// the code is regenerated.
// \</auto-generated>
//-----

```

```

// Generated from <grammarFileName> by ANTLR <ANTLRVersion>

```

```

// Unreachable code detected
#pragma warning disable 0162
// The variable '...' is assigned but its value is never used
#pragma warning disable 0219
// Missing XML comment for publicly visible type or member '...'
#pragma warning disable 1591
// Ambiguous reference in cref attribute
#pragma warning disable 419

```

```

>>

```

```

Parser(parser, funcs, atn, sempredFuncs, superClass) ::= <<
<Parser_(ctor="parser_ctor", ...)>
>>

```

```

Parser_(parser, funcs, atn, sempredFuncs, ctor, superClass) ::= <<
[System.CodeDom.Compiler.GeneratedCode("ANTLR", "<file.ANTLRVersion>")]
[System.CLSCompliant(false)]
public partial class <csIdentifier.(parser.name)> : <superClass; null="Parser"> {
protected static DFA[] decisionToDFA;
protected static PredictionContextCache sharedContextCache = new PredictionContextCache();
<if(parser.tokens)>

```

```

public const int
  <parser.tokens:{k | <k>=<parser.tokens.(k)>}; separator=", ", wrap, anchor>;
<endif>
public const int
  <parser.rules:{r | RULE_<r.name> = <r.index>}; separator=", ", wrap, anchor>;
public static readonly string[] ruleNames = {
  <parser.ruleNames:{r | "<r>"}; separator=", ", wrap, anchor>
};

<vocabulary(parser.literalNames, parser.symbolicNames)>

public override string GrammarFileName { get { return "<parser.grammarFileName>"; } }

public override string[] RuleNames { get { return ruleNames; } }

public override string SerializedAtn { get { return new string(_serializedATN); } }

static <csIdentifier.(parser.name)>() {
  decisionToDFA = new DFA[_ATN.NumberOfDecisions];
  for (int i = 0; i < _ATN.NumberOfDecisions; i++) {
    decisionToDFA[i] = new DFA(_ATN.GetDecisionState(i), i);
  }
}

<namedActions.members>
<parser:(ctor)()>
<funcs; separator="\n">

<if(sempredFuncs)>
public override bool Sempred(RuleContext _localctx, int ruleIndex, int predIndex) {
  switch (ruleIndex) {
    <parser.sempredFuncs.values:{f}
case <f.ruleIndex>: return <f.name>_sempred((<f.ctxType>)_localctx, predIndex);}; separator="\n">
  }
  return true;
}
<sempredFuncs.values; separator="\n">
<endif>

<atn>
}
>>

vocabulary(literalNames, symbolicNames) ::= <<
private static readonly string[] _LiteralNames = {
  <literalNames:{t | <t>}; null="null", separator=", ", wrap, anchor>
};
private static readonly string[] _SymbolicNames = {

```

```

    <symbolicNames:{t | <t>}; null="null", separator=", ", wrap, anchor>
};
public static readonly IVocabulary DefaultVocabulary = new Vocabulary(_LiteralNames, _SymbolicNames);

[NotNull]
public override IVocabulary Vocabulary
{
    get
    {
        return DefaultVocabulary;
    }
}
>>

dumpActions(recog, argFuncs, actionFuncs, sempredFuncs) ::= <<
<if(actionFuncs)>
public override void Action(RuleContext _localctx, int ruleIndex, int actionIndex) {
    switch (ruleIndex) {
        <recog.actionFuncs.values:{f}
        case <f.ruleIndex> : <f.name>_action(<if(!recog.modes)>(<f.ctxType>)<endif>_localctx, actionIndex); break;};
        separator="\n">
    }
}
<actionFuncs.values; separator="\n">
<endif>
<if(sempredFuncs)>
public override bool Sempred(RuleContext _localctx, int ruleIndex, int predIndex) {
    switch (ruleIndex) {
        <recog.sempredFuncs.values:{f}
        case <f.ruleIndex> : return <f.name>_sempred(<if(!recog.modes)>(<f.ctxType>)<endif>_localctx, predIndex);};
        separator="\n">
    }
    return true;
}
<sempredFuncs.values; separator="\n">
<endif>
>>

parser_ctor(parser) ::= <<
    public <csIdentifier.(parser.name)>(ITokenStream input) : this(input, Console.Out, Console.Error) { }

    public <csIdentifier.(parser.name)>(ITokenStream input, TextWriter output, TextWriter errorOutput)
    : base(input, output, errorOutput)
    {
        Interpreter = new ParserATNSimulator(this, _ATN, decisionToDFA, sharedContextCache);
    }
}
>>

```

```
/* This generates a private method since the actionIndex is generated, making an
* overriding implementation impossible to maintain.
*/
```

```
RuleActionFunction(r, actions) ::= <<
private void <r.name>_action(<r.ctxType> _localctx, int actionIndex) {
switch (actionIndex) {
<actions: {index|
case <index>: <actions.(index)> break; }; separator="\n">
}
}
>>
```

```
/* This generates a private method since the predIndex is generated, making an
* overriding implementation impossible to maintain.
*/
```

```
RuleSempredFunction(r, actions) ::= <<
private bool <r.name>_sempred(<r.ctxType> _localctx, int predIndex) {
switch (predIndex) {
<actions: {index|
case <index>: return <actions.(index)>; }; separator="\n">
}
return true;
}
>>
```

```
RuleFunction(currentRule,args,code,locals,ruleCtx,altLabelCtxs,namedActions,finallyAction,postamble,exceptions)
::= <<
```

```
<if(ruleCtx)>
<ruleCtx>
<endif>
<altLabelCtxs: {l | <altLabelCtxs.(l)> }; separator="\n">
```

```
[RuleVersion(<namedActions.version; null="0">)]
<if(currentRule.modifiers)><currentRule.modifiers: {f | <f> }><else>public <endif><currentRule.ctxType>
<csIdentifier.(currentRule.name)>(<args; separator=", ">) {
<currentRule.ctxType> _localctx = new <currentRule.ctxType>(Context, State<currentRule.args: {a | ,
<csIdentifier.(a.name)>>>);
EnterRule(_localctx, <currentRule.startState>, RULE_<currentRule.name>);
<namedActions.init>
<locals; separator="\n">
try {
<if(currentRule.hasLookaheadBlock)>
int _alt;
<endif>
<code>
<postamble; separator="\n">
<namedActions.after>
```

```

}
<if(exceptions)>
<exceptions; separator="\n">
<else>
catch (RecognitionException re) {
    _localctx.exception = re;
    ErrorHandler.ReportError(this, re);
    ErrorHandler.Recover(this, re);
}
<endif>
finally {
    <finallyAction>
    ExitRule();
}
return _localctx;
}
>>

LeftFactoredRuleFunction(currentRule,args,code,locals,namedActions,finallyAction,postamble) ::=
<<

<if(currentRule.modifiers)><currentRule.modifiers:{f | <f> }><else>private <endif><currentRule.ctxType>
<csIdentifier(currentRule.name)><(args; separator=", "> {
    <currentRule.ctxType> _localctx = new <currentRule.ctxType>(Context, State<currentRule.args:{a | ,
<csIdentifier(a.name)>>});
    EnterLeftFactoredRule(_localctx, <currentRule.startState>, RULE_<currentRule.variantOf>);
    <namedActions.init>
    <locals; separator="\n">
    try {
    <if(currentRule.hasLookaheadBlock)>
        int _alt;
    <endif>
    <code>
    <postamble; separator="\n">
    <namedActions.after>
    }
    catch (RecognitionException re) {
        _localctx.exception = re;
        ErrorHandler.ReportError(this, re);
        ErrorHandler.Recover(this, re);
    }
    finally {
        <finallyAction>
        ExitRule();
    }
    return _localctx;
}
>>

```

```

// This behaves similar to RuleFunction (enterRule is called, and no adjustments
// are made to the parse tree), but since it's still a variant no context class
// needs to be generated.
LeftUnfactoredRuleFunction(currentRule,args,code,locals,namedActions,finallyAction,postamble) ::=
<<

<if(currentRule.modifiers)><currentRule.modifiers:{f | <f> }><else>private <endif><currentRule.ctxType>
<csIdentifier(currentRule.name)><(args; separator=", "> {
  <currentRule.ctxType> _localctx = new <currentRule.ctxType>(Context, State<currentRule.args:{a | ,
<csIdentifier(a.name)>>>);
  EnterRule(_localctx, <currentRule.startState>, RULE_<currentRule.variantOf>);
  <namedActions.init>
  <locals; separator="\n">
  try {
<if(currentRule.hasLookaheadBlock)>
  int _alt;
<endif>
  <code>
  <postamble; separator="\n">
  <namedActions.after>
  }
  catch (RecognitionException re) {
    _localctx.exception = re;
    ErrorHandler.ReportError(this, re);
    ErrorHandler.Recover(this, re);
  }
  finally {
    <finallyAction>
    ExitRule();
  }
  return _localctx;
}
>>

LeftRecursiveRuleFunction(currentRule,args,code,locals,ruleCtx,altLabelCtxs,
namedActions,finallyAction,postamble) ::=
<<

<ruleCtx>
<altLabelCtxs:{l | <altLabelCtxs.(l)>}; separator="\n">

[RuleVersion(<namedActions.version; null="0">)]
<if(currentRule.modifiers)><currentRule.modifiers:{f | <f> }><else>public <endif><currentRule.ctxType>
<csIdentifier(currentRule.name)><(args; separator=", "> {
  return <csIdentifier(currentRule.name)><(0<currentRule.args:{a | , <csIdentifier(a.name)>>>);
}

```

```

private <currentRule.ctxType> <csIdentifier.(currentRule.name)>(int _p<args:{ a | , <a> }>) {
  ParserRuleContext _parentctx = Context;
  int _parentState = State;
  <currentRule.ctxType> _localctx = new <currentRule.ctxType>(Context, _parentState<currentRule.args:{ a | ,
<csIdentifier.(a.name)> }>);
  <currentRule.ctxType> _prevctx = _localctx;
  int _startState = <currentRule.startState>;
  EnterRecursionRule(_localctx, <currentRule.startState>, RULE_<currentRule.name>, _p);
  <namedActions.init>
  <locals; separator="\n">
  try {
  <if(currentRule.hasLookaheadBlock)>
    int _alt;
  <endif>
  <code>
  <postamble; separator="\n">
  <namedActions.after>
  }
  catch (RecognitionException re) {
    _localctx.exception = re;
    ErrorHandler.ReportError(this, re);
    ErrorHandler.Recover(this, re);
  }
  finally {
  <finallyAction>
  UnrollRecursionContexts(_parentctx);
  }
  return _localctx;
  }
  >>

```

```

CodeBlockForOuterMostAlt(currentOuterMostAltCodeBlock, locals, preamble, ops) ::= <<
  <if(currentOuterMostAltCodeBlock.altLabel)>_localctx = new <currentOuterMostAltCodeBlock.altLabel;
  format="cap">Context(_localctx);<endif>
  EnterOuterAlt(_localctx, <currentOuterMostAltCodeBlock.alt.altNum>);
  <CodeBlockForAlt(currentAltCodeBlock=currentOuterMostAltCodeBlock, ...)>
  >>

```

```

CodeBlockForAlt(currentAltCodeBlock, locals, preamble, ops) ::= <<
  {
  <locals; separator="\n">
  <preamble; separator="\n">
  <ops; separator="\n">
  }
  >>

```

```

LL1AltBlock(choice, preamble, alts, error) ::= <<
  State = <choice.stateNumber>;

```

```

ErrorHandler.Sync(this);
<if(choice.label)><labelref(choice.label)> = TokenStream.LT(1);<endif>
<preamble; separator="\n">
switch (TokenStream.LA(1)) {
<choice.altLook,alts:{look,alt| <cases(ttypes=look)>
<alt>
break;}; separator="\n">
default:
<error>
}
>>

```

```

LL1OptionalBlock(choice, alts, error) ::= <<
State = <choice.stateNumber>;
ErrorHandler.Sync(this);
switch (TokenStream.LA(1)) {
<choice.altLook,alts:{look,alt| <cases(ttypes=look)>
<alt>
break;}; separator="\n">
default:
break;
}
>>

```

```

LL1OptionalBlockSingleAlt(choice, expr, alts, preamble, error, followExpr) ::= <<
State = <choice.stateNumber>;
ErrorHandler.Sync(this);
<preamble; separator="\n">
if (<expr>) {
<alts; separator="\n">
}
<!else if ( !(<followExpr> ) ) <error>!>
>>

```

```

LL1StarBlockSingleAlt(choice, loopExpr, alts, preamble, iteration) ::= <<
State = <choice.stateNumber>;
ErrorHandler.Sync(this);
<preamble; separator="\n">
while (<loopExpr>) {
<alts; separator="\n">
State = <choice.loopBackStateNumber>;
ErrorHandler.Sync(this);
<iteration>
}
>>

```

```

LL1PlusBlockSingleAlt(choice, loopExpr, alts, preamble, iteration) ::= <<
State = <choice.blockStartStateNumber>;<! alt block decision !>

```

```

ErrorHandler.Sync(this);
<preamble; separator="\n">
do {
  <alts; separator="\n">
  State = <choice.stateNumber>;<! loopback/exit decision !>
  ErrorHandler.Sync(this);
  <iteration>
} while ( <loopExpr> );
>>

// LL(*) stuff

AltBlock(choice, preamble, alts, error) ::= <<
State = <choice.stateNumber>;
ErrorHandler.Sync(this);
<if(choice.label)><labelref(choice.label)> = TokenStream.LT(1);<endif>
<preamble; separator="\n">
switch ( Interpreter.AdaptivePredict(TokenStream,<choice.decision>,Context) ) {
<alts:{ alt |
case <i>:
  <alt>
  break;}; separator="\n">
}
>>

OptionalBlock(choice, alts, error) ::= <<
State = <choice.stateNumber>;
ErrorHandler.Sync(this);
switch ( Interpreter.AdaptivePredict(TokenStream,<choice.decision>,Context) ) {
<alts:{ alt |
case <i><if(!choice.ast.greedy)>+1<endif>:
  <alt>
  break;}; separator="\n">
}
>>

StarBlock(choice, alts, sync, iteration) ::= <<
State = <choice.stateNumber>;
ErrorHandler.Sync(this);
_alt = Interpreter.AdaptivePredict(TokenStream,<choice.decision>,Context);
while ( !_alt!=<choice.exitAlt> && !_alt!=global::Antlr4.Runtime.Atn.ATN.INVALID_ALT_NUMBER ) {
if ( _alt==1<if(!choice.ast.greedy)>+1<endif> ) {
  <iteration>
  <alts> <! should only be one !>
}
State = <choice.loopBackStateNumber>;
ErrorHandler.Sync(this);
_alt = Interpreter.AdaptivePredict(TokenStream,<choice.decision>,Context);

```

```
}  
>>
```

```
PlusBlock(choice, alts, error) ::= <<  
State = <choice.blockStartStateNumber>;<! alt block decision !>  
ErrorHandler.Sync(this);  
_alt = 1<if(!choice.ast.greedy)>+1<endif>;  
do {  
  switch (_alt) {  
    <alts:{alt|  
case <i><if(!choice.ast.greedy)>+1<endif>:  
    <alt>  
    break;}; separator="\n">  
  default:  
    <error>  
  }  
  State = <choice.loopBackStateNumber>;<! loopback/exit decision !>  
  ErrorHandler.Sync(this);  
  _alt = Interpreter.AdaptivePredict(TokenStream,<choice.decision>,Context);  
} while ( _alt!=<choice.exitAlt> && _alt!=global::Antlr4.Runtime.Atn.ATN.INVALID_ALT_NUMBER );  
>>
```

```
Sync(s) ::= "Sync(<s.expecting.name>);"
```

```
ThrowNoViableAlt(t) ::= "throw new NoViableAltException(this);"
```

```
TestSetInline(s) ::= <<  
<s.bitsets:{bits | <if(rest(rest(bits.ttypes)))><bitsetBitfieldComparison(s, bits)><else><bitsetInlineComparison(s,  
bits)><endif>}; separator=" || ">  
>>
```

```
// Java language spec 15.19 - shift operators mask operands rather than overflow to 0... need range test  
testShiftInRange(shiftAmount) ::= <<  
((<shiftAmount>) & ~0x3f) == 0  
>>
```

```
// produces smaller bytecode only when bits.ttypes contains more than two items  
bitsetBitfieldComparison(s, bits) ::= <%  
(<testShiftInRange({ <offsetShift(s.varName, bits.shift)>})> && ((1L \<< <offsetShift(s.varName, bits.shift)>) &  
(<bits.ttypes:{ttype | (1L \<< <offsetShift(tokenType.(ttype), bits.shift)>)}; separator=" | ">)) != 0)  
%>
```

```
isZero ::= [  
  "0":true,  
  default:false  
]
```

```
offsetShift(shiftAmount, offset) ::= <%
```

```

<if(!isZero(offset))><shiftAmount> - <offset><else><shiftAmount><endif>
%>

// produces more efficient bytecode when bits.ttypes contains at most two items
bitsetInlineComparison(s, bits) ::= <%
<bits.ttypes:{ ttype | <s.varName>==<tokenType.(ttype)>}; separator=" || ">
%>

cases(ttypes) ::= <<
<types:{ t | case <tokenType.(t)>:}; separator="\n">
>>

InvokeRule(r, argExprsChunks) ::= <<
State = <r.stateNumber>; <if(r.labels)><r.labels:{1 | <labelref(l)> =
}><endif><csIdentifier.(r.name)><(if(r.ast.options.p)><r.ast.options.p><if(argExprsChunks)><endif><endif><arg
ExprsChunks>;
>>

MatchToken(m) ::= <<
State = <m.stateNumber>; <if(m.labels)><m.labels:{1 | <labelref(l)> = }><endif>Match(<tokenType.(m.name)>);
>>

MatchSet(m, expr, capture) ::= "<CommonSetStuff(m, expr, capture, false)>"

MatchNotSet(m, expr, capture) ::= "<CommonSetStuff(m, expr, capture, true)>"

CommonSetStuff(m, expr, capture, invert) ::= <<
State = <m.stateNumber>;
<if(m.labels)><m.labels:{1 | <labelref(l)> = }>TokenStream.LT(1);<endif>
<capture>
if ( <if(invert)><m.varName> \<= 0 || <else>!<endif><(expr)> ) {
  <if(m.labels)><m.labels:{1 | <labelref(l)> = }><endif>ErrorHandler.RecoverInline(this);
}
else {
  ErrorHandler.ReportMatch(this);
  Consume();
}
>>

Wildcard(w) ::= <<
State = <w.stateNumber>;
<if(w.labels)><w.labels:{1 | <labelref(l)> = }><endif>MatchWildcard();
>>

// ACTION STUFF

Action(a, foo, chunks) ::= "<chunks>"

```

```

ArgAction(a, chunks) ::= "<chunks>"

SemPred(p, chunks, failChunks) ::= <<
State = <p.stateNumber>;
if (!<chunks>) throw new FailedPredicateException(this, <p.predicate><if(failChunks)>,
<failChunks><elseif(p.msg)>, <p.msg><endif>);
>>

ExceptionClause(e, catchArg, catchAction) ::= <<
catch (<catchArg>) {
  <catchAction>
}
>>

// lexer actions are not associated with model objects

LexerSkipCommand() ::= "Skip();"
LexerMoreCommand() ::= "More();"
LexerPopModeCommand() ::= "PopMode();"

LexerTypeCommand(arg, grammar) ::= "_type = <tokenType.(arg)>";
LexerChannelCommand(arg, grammar) ::= "_channel = <channelName.(arg)>";
LexerModeCommand(arg, grammar) ::= "_mode = <modeName.(arg)>";
LexerPushModeCommand(arg, grammar) ::= "PushMode(<modeName.(arg)>);"

ActionText(t) ::= "<t.text>"
ActionTemplate(t) ::= "<t.st>"
ArgRef(a) ::= "_localctx.<csIdentifier.(a.name)>"
LocalRef(a) ::= "_localctx.<csIdentifier.(a.name)>"
RetValRef(a) ::= "_localctx.<csIdentifier.(a.name)>"
QRetValRef(a) ::= "<ctx(a)>.<a.dict>.<csIdentifier.(a.name)>"
/** How to translate $tokenLabel */
TokenRef(t) ::= "<ctx(t)>.<csIdentifier.(tokenType.(t.name))>"
LabelRef(t) ::= "<ctx(t)>.<csIdentifier.(t.name)>"
ListLabelRef(t) ::= "<ctx(t)>.<ListLabelName(csIdentifier.(t.name))>"
SetAttr(s,rhsChunks) ::= "<ctx(s)>.<csIdentifier.(s.name)> = <rhsChunks>";

TokenLabelType() ::= "<file.TokenLabelType; null={IToken}>"
InputSymbolType() ::= "<file.InputSymbolType; null={IToken}>"

TokenPropertyRef_text(t) ::= "<ctx(t)>.<tokenType.(t.label)>!=null?<ctx(t)>.<tokenType.(t.label)>.Text:null"
TokenPropertyRef_type(t) ::= "<ctx(t)>.<tokenType.(t.label)>!=null?<ctx(t)>.<tokenType.(t.label)>.Type:0"
TokenPropertyRef_line(t) ::= "<ctx(t)>.<tokenType.(t.label)>!=null?<ctx(t)>.<tokenType.(t.label)>.Line:0"
TokenPropertyRef_pos(t) ::=
"<ctx(t)>.<tokenType.(t.label)>!=null?<ctx(t)>.<tokenType.(t.label)>.CharPositionInLine:0"
TokenPropertyRef_channel(t) ::=
"<ctx(t)>.<tokenType.(t.label)>!=null?<ctx(t)>.<tokenType.(t.label)>.Channel:0"
TokenPropertyRef_index(t) ::=

```

```

("<ctx(t)>.<tokenType.(t.label)>!=null?<ctx(t)>.<tokenType.(t.label)>.TokenIndex:0)"
TokenPropertyRef_int(t) ::=
("<ctx(t)>.<tokenType.(t.label)>!=null?int.Parse(<ctx(t)>.<tokenType.(t.label)>.Text):0)"

RulePropertyRef_start(r) ::= "<ctx(r)>.<r.label>!=null?(<ctx(r)>.<r.label>.Start):null)"
RulePropertyRef_stop(r) ::= "<ctx(r)>.<r.label>!=null?(<ctx(r)>.<r.label>.Stop):null)"
RulePropertyRef_text(r) ::=
("<ctx(r)>.<r.label>!=null?TokenStream.GetText(<ctx(r)>.<r.label>.Start,<ctx(r)>.<r.label>.Stop):null)"
RulePropertyRef_ctx(r) ::= "<ctx(r)>.<r.label>"
RulePropertyRef_parser(r) ::= "this"

ThisRulePropertyRef_start(r) ::= "_localctx.Start"
ThisRulePropertyRef_stop(r) ::= "_localctx.Stop"
ThisRulePropertyRef_text(r) ::= "TokenStream.GetText(_localctx.Start, TokenStream.LT(-1))"
ThisRulePropertyRef_ctx(r) ::= "_localctx"
ThisRulePropertyRef_parser(r) ::= "this"

NonLocalAttrRef(s) ::= <%((<s.ruleName;
format="cap">Context)GetInvokingContext(<s.ruleIndex>)).<csIdentifier.(s.name)>%>
SetNonLocalAttr(s, rhsChunks) ::=
<%((<s.ruleName; format="cap">Context)GetInvokingContext(<s.ruleIndex>)).<csIdentifier.(s.name)> =
<rhsChunks>;%>

AddToLabelList(a) ::= "<ctx(a.label)>.<a.listName>.Add(<labelref(a.label)>);"

TokenDecl(t) ::= "<TokenLabelType() <csIdentifier.(tokenType.(t.name))>"
TokenTypeDecl(t) ::= "int <csIdentifier.(tokenType.(t.name))>);"
TokenListDecl(t) ::= "IList<IToken> <csIdentifier.(tokenType.(t.name))> = new List<IToken>()"
RuleContextDecl(r) ::= "<r.ctxName> <csIdentifier.(r.name)>"
RuleContextListDecl(rdecl) ::= "IList<<rdecl.ctxName>> <csIdentifier.(rdecl.name)> = new
List<<rdecl.ctxName>>()"

contextGetterCollection(elementType) ::= <%
<elementType>[]
%>

ContextTokenGetterDecl(t) ::=
"public ITerminalNode <csIdentifier.(tokenType.(t.name))>() { return
GetToken(<csIdentifier.(parser.name)>.<csIdentifier.(tokenType.(t.name))>, 0); }"
ContextTokenListGetterDecl(t) ::= <<
public <contextGetterCollection("ITerminalNode")> <csIdentifier.(tokenType.(t.name))>() { return
GetTokens(<csIdentifier.(parser.name)>.<csIdentifier.(tokenType.(t.name))>); }
>>

ContextTokenListIndexedGetterDecl(t) ::= <<
public ITerminalNode <csIdentifier.(tokenType.(t.name))>(int i) {
return GetToken(<csIdentifier.(parser.name)>.<csIdentifier.(tokenType.(t.name))>, i);
}
>>

```

```

ContextRuleGetterDecl(r) ::= <<
public <r.ctxName> <csIdentifier.(r.name)>() {
    return GetRuleContext\<<r.ctxName>\>(0);
}
>>

ContextRuleListGetterDecl(r) ::= <<
public <contextGetterCollection({<r.ctxName>})> <csIdentifier.(r.name)>() {
    return GetRuleContexts\<<r.ctxName>\>();
}
>>

ContextRuleListIndexedGetterDecl(r) ::= <<
public <r.ctxName> <csIdentifier.(r.name)>(int i) {
    return GetRuleContext\<<r.ctxName>\>(i);
}
>>

LexerRuleContext() ::= "RuleContext"

/** The rule context name is the rule followed by a suffix; e.g.,
 * r becomes rContext.
 */
RuleContextNameSuffix() ::= "Context"

ImplicitTokenLabel(tokenName) ::= "_<tokenType.(tokenName)>"
ImplicitRuleLabel(ruleName) ::= "_<ruleName>"
ImplicitSetLabel(id) ::= "_tset<id>"
ListLabelName(label) ::= "_<label>"

CaptureNextToken(d) ::= "<d.varName> = TokenStream.LT(1);"
CaptureNextTokenType(d) ::= "<d.varName> = TokenStream.LA(1);"

StructDecl(struct,ctorAttrs,attrs,getters,dispatchMethods,interfaces,extensionMembers,
    superClass={ParserRuleContext}) ::= <<
public partial class <struct.name> :
<if(contextSuperClass)><contextSuperClass><else>ParserRuleContext<endif><if(interfaces)>, <interfaces;
separator=", "><endif> {
    <attrs:{a | public <a>;}; separator="\n">
    <getters:{g | <g>;}; separator="\n">
    <if(ctorAttrs)>public <struct.name>(ParserRuleContext parent, int invokingState) : base(parent, invokingState) {
    }<endif>
    public <struct.name>(ParserRuleContext parent, int invokingState<ctorAttrs:{a | , <a>}>)
    : base(parent, invokingState)
    {
        <struct.ctorAttrs:{a | this.<csIdentifier.(a.name)> = <csIdentifier.(a.name)>;}; separator="\n">
    }
    public override int RuleIndex { get { return RULE_<struct.derivedFromName>; } }
<if(struct.provideCopyFrom)> <! don't need copy unless we have subclasses !>
    public <struct.name>() { }

```

```

public virtual void CopyFrom(<struct.name> context) {
    base.CopyFrom(context);
    <struct.attrs:{a | this.<csIdentifier.(a.name)> = context.<csIdentifier.(a.name)>;}; separator="\n">
}
<endif>
<dispatchMethods; separator="\n">
<extensionMembers; separator="\n">
}
>>

```

```

AltLabelStructDecl(struct,attrs, getters,dispatchMethods) ::= <<
public partial class <struct.name> : <currentRule.name; format="cap">Context {
    <attrs:{a | public <a>;}; separator="\n">
    <getters:{g | <g>;}; separator="\n">
    public <struct.name>(<currentRule.name; format="cap">Context context) { CopyFrom(context); }
    <dispatchMethods; separator="\n">
}
>>

```

```

ListenerDispatchMethod(method) ::= <<
public override void <if(method.isEnter)>Enter<else>Exit<endif>Rule(IParseTreeListener listener) {
    I<parser.grammarName>Listener typedListener = listener as I<parser.grammarName>Listener;
    if (typedListener != null) typedListener.<if(method.isEnter)>Enter<else>Exit<endif><struct.derivedFromName;
format="cap">(this);
}
>>

```

```

VisitorDispatchMethod(method) ::= <<
public override TResult Accept<TResult>(IParseTreeVisitor<TResult> visitor) {
    I<parser.grammarName>Visitor<TResult> typedVisitor = visitor as I<parser.grammarName>Visitor<TResult>;
    if (typedVisitor != null) return typedVisitor.Visit<struct.derivedFromName; format="cap">(this);
    else return visitor.VisitChildren(this);
}
>>

```

```

AttributeDecl(d) ::= "<d.type> <csIdentifier.(d.name)><if(d.initValue)> = <d.initValue><endif>"

```

```

/** If we don't know location of label def x, use this template */

```

```

labelref(x) ::= "<if(!x.isLocal)><typedContext(x.ctx)>.<endif><csIdentifier.(x.name)>"

```

```

/** For any action chunk, what is correctly-typed context struct ptr? */

```

```

ctx(actionChunk) ::= "<typedContext(actionChunk.ctx)>"

```

```

// only casts _localctx to the type when the cast isn't redundant (i.e. to a sub-context for a labeled alt)

```

```

typedContext(ctx) ::= "<if(ctx.provideCopyFrom)>((<ctx.name>)_localctx)<else>_localctx<endif>"

```

```

// used for left-recursive rules

```

```

recRuleAltPredicate(ruleName,opPrec) ::= "Precpred(Context, <opPrec>)"

```

```

recRuleSetReturnAction(src,name) ::= "$<name>=$<src>.<name>";
recRuleSetStopToken()          ::= "Context.Stop = TokenStream.LT(-1);"

recRuleAltStartAction(ruleName, ctxName, label, isListLabel) ::= <<
_localctx = new <ctxName>Context(_parentctx, _parentState);
<if(label)>
<if(isListLabel)>
_localctx.<label>.Add(_prevctx);
<else>
_localctx.<label> = _prevctx;
<endif>
<endif>
PushNewRecursionContext(_localctx, _startState, RULE_<ruleName>);
>>

recRuleLabeledAltStartAction(ruleName, currentAltLabel, label, isListLabel) ::= <<
_localctx = new <currentAltLabel; format="cap">Context(new <ruleName; format="cap">Context(_parentctx,
_parentState));
<if(label)>
<if(isListLabel)>
((<currentAltLabel; format="cap">Context)_localctx).<label>.Add(_prevctx);
<else>
((<currentAltLabel; format="cap">Context)_localctx).<label> = _prevctx;
<endif>
<endif>
PushNewRecursionContext(_localctx, _startState, RULE_<ruleName>);
>>

recRuleReplaceContext(ctxName) ::= <<
_localctx = new <ctxName>Context(_localctx);
Context = _localctx;
_prevctx = _localctx;
>>

recRuleSetPrevCtx() ::= <<
if ( ParseListeners!=null )
    TriggerExitRuleEvent();
_prevctx = _localctx;
>>

LexerFile(file, lexer, namedActions) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
<if(file.genPackage)>
namespace <file.genPackage> {
<endif>
<namedActions.header>
using System;

```

```

using System.IO;
using System.Text;
using Antlr4.Runtime;
using Antlr4.Runtime.Atn;
using Antlr4.Runtime.Misc;
using DFA = Antlr4.Runtime.Dfa.DFA;

<lexer>
<if(file.genPackage)>
} // namespace <file.genPackage>
<endif>
>>

Lexer(lexer, atn, actionFuncs, sempredFuncs, superClass) ::= <<
[System.CodeDom.Compiler.GeneratedCode("ANTLR", "<file.ANTLRVersion>")]
[System.CLSCompliant(false)]
public partial class <csIdentifier.(lexer.name)> : <superClass; null="Lexer"> {
protected static DFA[] decisionToDFA;
protected static PredictionContextCache sharedContextCache = new PredictionContextCache();
public const int
<lexer.tokens:{k | <tokenType.(k)>=<lexer.tokens.(k)>}; separator=", ", wrap, anchor>;
<if(lexer.channels)>
public const int
<lexer.channels:{k | <csIdentifier.(k)>=<lexer.channels.(k)>}; separator=", ", wrap, anchor>;
<endif>
<if(rest(lexer.modes)>
public const int
<rest(lexer.modes):{m | <m>=<i>}; separator=", ", wrap, anchor>;
<endif>
public static string[] channelNames = {
"DEFAULT_TOKEN_CHANNEL", "HIDDEN"<if (lexer.channels)>, <lexer.channels:{c | "<c">"}; separator=", ",
wrap, anchor><endif>
};

public static string[] modeNames = {
<lexer.modes:{m | "<m">"}; separator=", ", wrap, anchor>
};

public static readonly string[] ruleNames = {
<lexer.ruleNames:{r | "<r">"}; separator=", ", wrap, anchor>
};

<namedActions.members>

public <csIdentifier.(lexer.name)>(ICharStream input)
: this(input, Console.Out, Console.Error) { }

public <csIdentifier.(lexer.name)>(ICharStream input, TextWriter output, TextWriter errorOutput)

```

```

: base(input, output, errorOutput)
{
  Interpreter = new LexerATNSimulator(this, _ATN, decisionToDFA, sharedContextCache);
}

<vocabulary(lexer.literalNames, lexer.symbolicNames)>

public override string GrammarFileName { get { return "<lexer.grammarFileName>"; } }

public override string[] RuleNames { get { return ruleNames; } }

public override string[] ChannelNames { get { return channelNames; } }

public override string[] ModeNames { get { return modeNames; } }

public override string SerializedAtn { get { return new string(_serializedATN); } }

static <csIdentifier.(lexer.name)>() {
  decisionToDFA = new DFA[_ATN.NumberOfDecisions];
  for (int i = 0; i < _ATN.NumberOfDecisions; i++) {
    decisionToDFA[i] = new DFA(_ATN.GetDecisionState(i), i);
  }
}
<dumpActions(lexer, "", actionFuncs, sempredFuncs)>
<atn>
}
>>

SerializedATN(model) ::= <<
private static char[] _serializedATN = {
  <model.serialized; separator=", ", wrap>,
};

public static readonly ATN _ATN =
  new ATNDeserializer().Deserialize(_serializedATN);

>>

initValue(typeName) ::= <<
default(<typeName>)
>>

codeFileExtension() ::= ".cs"

modeName ::= [
  "DEFAULT_MODE" : "DefaultMode",

```

```
default : key
```

```
]
```

```
channelName ::= [
```

```
"HIDDEN" : "Hidden",
```

```
"DEFAULT_TOKEN_CHANNEL" : "DefaultTokenChannel",
```

```
default : key
```

```
]
```

```
tokenType ::= [
```

```
"EOF" : "Eof",
```

```
default : key
```

```
]
```

```
csIdentifier ::= [
```

```
"abstract" : "@abstract",
```

```
"as" : "@as",
```

```
"base" : "@base",
```

```
"bool" : "@bool",
```

```
"break" : "@break",
```

```
"byte" : "@byte",
```

```
"case" : "@case",
```

```
"catch" : "@catch",
```

```
"char" : "@char",
```

```
"checked" : "@checked",
```

```
"class" : "@class",
```

```
"const" : "@const",
```

```
"continue" : "@continue",
```

```
"decimal" : "@decimal",
```

```
"default" : "@default",
```

```
"delegate" : "@delegate",
```

```
"do" : "@do",
```

```
"double" : "@double",
```

```
"else" : "@else",
```

```
"enum" : "@enum",
```

```
"event" : "@event",
```

```
"explicit" : "@explicit",
```

```
"extern" : "@extern",
```

```
"false" : "@false",
```

```
"finally" : "@finally",
```

```
"fixed" : "@fixed",
```

```
"float" : "@float",
```

```
"for" : "@for",
```

```
"foreach" : "@foreach",
```

```
"goto" : "@goto",
```

```
"if" : "@if",
```

```
"implicit" : "@implicit",
```

```
"in" : "@in",
```

```
"int" : "@int",
"interface" : "@interface",
"internal" : "@internal",
"is" : "@is",
"lock" : "@lock",
"long" : "@long",
"namespace" : "@namespace",
"new" : "@new",
"null" : "@null",
"object" : "@object",
"operator" : "@operator",
"out" : "@out",
"override" : "@override",
"params" : "@params",
"private" : "@private",
"protected" : "@protected",
"public" : "@public",
"readonly" : "@readonly",
"ref" : "@ref",
"return" : "@return",
"sbyte" : "@sbyte",
"sealed" : "@sealed",
"short" : "@short",
"sizeof" : "@sizeof",
"stackalloc" : "@stackalloc",
"static" : "@static",
"string" : "@string",
"struct" : "@struct",
"switch" : "@switch",
"this" : "@this",
"throw" : "@throw",
>true" : "@true",
"try" : "@try",
"typeof" : "@typeof",
"uint" : "@uint",
"ulong" : "@ulong",
"unchecked" : "@unchecked",
"unsafe" : "@unsafe",
"ushort" : "@ushort",
"using" : "@using",
"virtual" : "@virtual",
"values" : "@values",
"void" : "@void",
"volatile" : "@volatile",
"while" : "@while",
default : key
]
```

Found in path(s):

* /opt/cola/permits/1473460068_1668481512.5494933/0/antlr4-4-8-1-jar/org/antlr/v4/tool/templates/codegen/CSharp/CSharp.stg

No license file was found, but licenses were detected in source scan.

/*

* [The "BSD license"]

* Copyright (c) 2012-2016 Terence Parr

* Copyright (c) 2012-2016 Sam Harwell

* Copyright (c) 2014 Eric Vergnaud

* All rights reserved.

*

* Redistribution and use in source and binary forms, with or without

* modification, are permitted provided that the following conditions

* are met:

*

* 1. Redistributions of source code must retain the above copyright

* notice, this list of conditions and the following disclaimer.

* 2. Redistributions in binary form must reproduce the above copyright

* notice, this list of conditions and the following disclaimer in the

* documentation and/or other materials provided with the distribution.

* 3. The name of the author may not be used to endorse or promote products

* derived from this software without specific prior written permission.

*

* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR

* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES

* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.

* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,

* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT

* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,

* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY

* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT

* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF

* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

/** ANTLR tool checks output templates are compatible with tool code generation.

* For now, a simple string match used on x.y of x.y.z scheme.

* Must match Tool.VERSION during load to templates.

*

* REQUIRED.

*/

pythonTypeInitMap ::= [

"bool": "False",

"int": "0",

"float": "0.0",

"str": "",

```

default:"None" // anything other than a primitive type is an object
]

// args must be <object-model-object>, <fields-resulting-in-STs>

ParserFile(file, parser, namedActions, contextSuperClass) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
# encoding: utf-8
from antlr4 import *
from io import StringIO
import sys
if sys.version_info[1] > 5:
    from typing import TextIO
else:
    from typing.io import TextIO

<namedActions.header>
<parser>

>>

ListenerFile(file, header, namedActions) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
from antlr4 import *
if __name__ is not None and "." in __name__:
    from <file.parserName> import <file.parserName>
else:
    from <file.parserName> import <file.parserName>
<header>

# This class defines a complete listener for a parse tree produced by <file.parserName>.
class <file.grammarName>Listener(ParseTreeListener):

    <file.listenerNames:{lname |
# Enter a parse tree produced by <file.parserName>#<lname>.
def enter<lname; format="cap">(self, ctx:<file.parserName>.<lname; format="cap">Context):
    pass

# Exit a parse tree produced by <file.parserName>#<lname>.
def exit<lname; format="cap">(self, ctx:<file.parserName>.<lname; format="cap">Context):
    pass

}; separator="\n">

del <file.parserName>
>>

```

```

VisitorFile(file, header, namedActions) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
from antlr4 import *
if __name__ is not None and "." in __name__:
    from .<file.parserName> import <file.parserName>
else:
    from <file.parserName> import <file.parserName>
<header>

# This class defines a complete generic visitor for a parse tree produced by <file.parserName>.

class <file.grammarName>Visitor(ParseTreeVisitor):

    <file.visitorNames:{lname |
# Visit a parse tree produced by <file.parserName>#<lname>.
def visit<lname; format="cap">(self, ctx:<file.parserName>.<lname; format="cap">Context):
    return self.visitChildren(ctx)

}; separator="\n">

del <file.parserName>
>>

fileHeader(grammarFileName, ANTLRVersion) ::= <<
# Generated from <grammarFileName> by ANTLR <ANTLRVersion>
>>

Parser(parser, funcs, atn, sempredFuncs, superClass) ::= <<
<Parser_(ctor="parser_ctor", ...)>
>>

Parser_(parser, funcs, atn, sempredFuncs, ctor, superClass) ::= <<
<if(superClass)>
if __name__ is not None and "." in __name__:
    from .<superClass> import <superClass>
else:
    from <superClass> import <superClass>

<endif>
<atn>

class <parser.name> ( <if(superClass)><superClass><else>Parser<endif> ):

    grammarFileName = "<parser.grammarFileName>"

    atn = ATNDeserializer().deserialize(serializedATN())

```

```

decisionsToDFA = [ DFA(ds, i) for i, ds in enumerate(atn.decisionToState) ]

sharedContextCache = PredictionContextCache()

literalNames = [ <parser.literalNames:{t | <t>}; null="\\"<INVALID>\\"", separator=", ", wrap, anchor> ]

symbolicNames = [ <parser.symbolicNames:{t | <t>}; null="\\"<INVALID>\\"", separator=", ", wrap, anchor> ]

<parser.rules:{r | RULE_<r.name> = <r.index>}; separator="\n", wrap, anchor>

ruleNames = [ <parser.ruleNames:{r | "<r>"}; separator=", ", wrap, anchor> ]

EOF = <TokenLabelType().EOF
<if(parser.tokens)>
<parser.tokens:{k | <k>=<parser.tokens.(k)>}; separator="\n", wrap, anchor>
<endif>

<parser:(ctor())>

<namedActions.members>

<funcs; separator="\n">

<if(sembredFuncs)>
def sembred(self, localctx:RuleContext, ruleIndex:int, predIndex:int):
    if self._predicates == None:
        self._predicates = dict()
<parser.sembredFuncs.values:{ f |
    self._predicates[<f.ruleIndex>] = self.<f.name>_sembred}; separator="\n    ">
    pred = self._predicates.get(ruleIndex, None)
    if pred is None:
        raise Exception("No predicate with index:" + str(ruleIndex))
    else:
        return pred(localctx, predIndex)

<sembredFuncs.values; separator="\n">
<endif>

>>

dumpActions(recog, argFuncs, actionFuncs, sembredFuncs) ::= <<
<if(actionFuncs)>
def action(self, localctx:RuleContext, ruleIndex:int, actionIndex:int):
    if self._actions is None:
        actions = dict()

```

```

<recog.actionFuncs.values: {f|
    actions[<f.ruleIndex>] = self.<f.name>_action }; separator="\n">
    self._actions = actions
    action = self._actions.get(ruleIndex, None)
    if action is not None:
        action(localctx, actionIndex)
    else:
        raise Exception("No registered action for:" + str(ruleIndex))

<actionFuncs.values; separator="\n">

<endif>
<if( sempredFuncs)>
def sempred(self, localctx:RuleContext, ruleIndex:int, predIndex:int):
    if self._predicates is None:
        preds = dict()
<recog.sempredFuncs.values: {f|
    preds[<f.ruleIndex>] = self.<f.name>_sempred}; separator="\n">
    self._predicates = preds
    pred = self._predicates.get(ruleIndex, None)
    if pred is not None:
        return pred(localctx, predIndex)
    else:
        raise Exception("No registered predicate for:" + str(ruleIndex))

<sempredFuncs.values; separator="\n">
<endif>
>>

parser_ctor(p) ::= <<
def __init__(self, input:TokenStream, output:TextIO = sys.stdout):
    super().__init__(input, output)
    self.checkVersion("<file.ANTLRVersion>")
    self._interp = ParserATNSimulator(self, self.atn, self.decisionsToDFA, self.sharedContextCache)
    self._predicates = None

>>

/* This generates a private method since the actionIndex is generated, making an
* overriding implementation impossible to maintain.
*/
RuleActionFunction(r, actions) ::= <<

def <r.name>_action(self, localctx:<r.ctxType> , actionIndex:int):
<actions: {index|
<if(first(actions))>
    if actionIndex == <index>:
        <actions.(index)>

```

```

<elseif(rest(actions))>
  elif actionIndex == <index>:
    <actions.(index)>
<endif> }; separator="\n">
>>

/* This generates a private method since the predIndex is generated, making an
* overriding implementation impossible to maintain.
*/
RuleSempredFunction(r, actions) ::= <<
def <r.name>_sempred(self, localctx:<r.ctxType>, predIndex:int):
  <actions:{index|
<if(first(actions))>
  if predIndex == <index>:
    return <actions.(index)>
<elseif(rest(actions))>
  elif predIndex == <index>:
    return <actions.(index)>
<endif> }; separator="\n">

>>

RuleFunction(currentRule,args,code,locals,ruleCtx,altLabelCtxs,namedActions,finallyAction,postamble,exceptions)
::= <<

<ruleCtx>

<altLabelCtxs:{l | <altLabelCtxs.(l)>}; separator="\n">

def <currentRule.name>(self<currentRule.args:{a | , <a.name><if(a.type)>:<a.type><endif> }>):

  localctx = <parser.name>.<currentRule.ctxType>(self, self._ctx, self.state<currentRule.args:{a | , <a.name> }>)
  self.enterRule(localctx, <currentRule.startState>, self.RULE_<currentRule.name>)
  <namedActions.init>
  <locals; separator="\n">
  try:
    <code>
    <postamble; separator="\n">
    <namedActions.after>
  <if(exceptions)>
  <exceptions; separator="\n">
  <else>
  except RecognitionException as re:
    localctx.exception = re
    self._errHandler.reportError(self, re)
    self._errHandler.recover(self, re)
  <endif>
  finally:

```

```

    <finallyAction>
    self.exitRule()
    return localctx

>>

LeftRecursiveRuleFunction(currentRule,args,code,locals,ruleCtx,altLabelCtxs,
    namedActions,finallyAction,postamble) ::=
<<

<ruleCtx>
<altLabelCtxs:{l | <altLabelCtxs.(l)>} ; separator="\n">

def <currentRule.name>(self, _p:int=0<if(currentRule.args)>, <args:{ a | , <a> }><endif>):
    _parentctx = self._ctx
    _parentState = self.state
    localctx = <parser.name>.<currentRule.ctxType>(self, self._ctx, _parentState<args:{ a | , <a.name> }>)
    _prevctx = localctx
    _startState = <currentRule.startState>
    self.enterRecursionRule(localctx, <currentRule.startState>, self.RULE_<currentRule.name>, _p)
    <namedActions.init>
    <locals; separator="\n">
    try:
        <code>
        <postamble; separator="\n">
        <namedActions.after>
    except RecognitionException as re:
        localctx.exception = re
        self._errHandler.reportError(self, re)
        self._errHandler.recover(self, re)
    finally:
        <finallyAction>
        self.unrollRecursionContexts(_parentctx)
    return localctx

>>

CodeBlockForOuterMostAlt(currentOuterMostAltCodeBlock, locals, preamble, ops) ::= <<
<if(currentOuterMostAltCodeBlock.altLabel)>localctx = <parser.name>.<currentOuterMostAltCodeBlock.altLabel>;
format="cap">Context(self, localctx)<endif>
self.enterOuterAlt(localctx, <currentOuterMostAltCodeBlock.alt.altNum>)
<CodeBlockForAlt(currentAltCodeBlock=currentOuterMostAltCodeBlock, ...)>
>>

CodeBlockForAlt(currentAltCodeBlock, locals, preamble, ops) ::= <<
<locals; separator="\n">
<preamble; separator="\n">
<ops; separator="\n">

```

>>

```
LL1AltBlock(choice, preamble, alts, error) ::= <<
self.state = <choice.stateNumber>
self._errHandler.sync(self)
<if(choice.label)><labelref(choice.label)> = _input.LT(1)<endif>
<preamble; separator="\n">
token = self._input.LA(1)
<choice.altLook,alts:{look,alt| <cases(ttypes=look)>
  <alt>
  pass }; separator="\nel">
else:
  <error>
```

>>

```
LL1OptionalBlock(choice, alts, error) ::= <<
self.state = <choice.stateNumber>
self._errHandler.sync(self)
token = self._input.LA(1)
<choice.altLook,alts:{look,alt| <cases(ttypes=look)>
  <alt>
  pass }; separator="\nel">
else:
  pass
```

>>

```
LL1OptionalBlockSingleAlt(choice, expr, alts, preamble, error, followExpr) ::= <<
self.state = <choice.stateNumber>
self._errHandler.sync(self)
<preamble; separator="\n">
if <expr>:
  <alts; separator="\n">

<!else if ( !(<followExpr> ) ) <error>!>
>>
```

```
LL1StarBlockSingleAlt(choice, loopExpr, alts, preamble, iteration) ::= <<
self.state = <choice.stateNumber>
self._errHandler.sync(self)
<preamble; separator="\n">
while <loopExpr>:
  <alts; separator="\n">
  self.state = <choice.loopBackStateNumber>
  self._errHandler.sync(self)
  <iteration>
```

>>

```
LL1PlusBlockSingleAlt(choice, loopExpr, alts, preamble, iteration) ::= <<
self.state = <choice.blockStartStateNumber> <! alt block decision !>
self._errHandler.sync(self)
<preamble; separator="\n">
while True:
    <alts; separator="\n">
    self.state = <choice.stateNumber> <! loopback/exit decision !>
    self._errHandler.sync(self)
    <iteration>
    if not (<loopExpr>):
        break
```

>>

```
// LL(*) stuff
```

```
AltBlock(choice, preamble, alts, error) ::= <<
self.state = <choice.stateNumber>
self._errHandler.sync(self)
<if(choice.label)><labelref(choice.label)> = _input.LT(1)<endif>
<preamble; separator="\n">
la_ = self._interp.adaptivePredict(self._input,<choice.decision>,self._ctx)
<alts:{ alt |
if la_ == <i>:
    <alt>
    pass
}; separator="\nel">
```

>>

```
OptionalBlock(choice, alts, error) ::= <<
self.state = <choice.stateNumber>
self._errHandler.sync(self)
la_ = self._interp.adaptivePredict(self._input,<choice.decision>,self._ctx)
<alts:{ alt |
if la_ == <i><if(!choice.ast.greedy)>+1<endif>:
    <alt>
}; separator="\nel">
```

>>

```
StarBlock(choice, alts, sync, iteration) ::= <<
self.state = <choice.stateNumber>
self._errHandler.sync(self)
_alt = self._interp.adaptivePredict(self._input,<choice.decision>,self._ctx)
while _alt!=<choice.exitAlt> and _alt!=ATN.INVALID_ALT_NUMBER:
```

```

if _alt==1<if(!choice.ast.greedy)>+1<endif>:
  <iteration>
  <alts> <! should only be one !>
  self.state = <choice.loopBackStateNumber>
  self._errHandler.sync(self)
  _alt = self._interp.adaptivePredict(self._input,<choice.decision>,self._ctx)

>>

```

```

PlusBlock(choice, alts, error) ::= <<
self.state = <choice.blockStartStateNumber> <! alt block decision !>
self._errHandler.sync(self)
_alt = 1<if(!choice.ast.greedy)>+1<endif>
while _alt!=<choice.exitAlt> and _alt!=ATN.INVALID_ALT_NUMBER:
  <alts:{alt|
if _alt == <i><if(!choice.ast.greedy)>+1<endif>:
  <alt>
}; separator="\nел">
  else:
    <error>
  self.state = <choice.loopBackStateNumber> <! loopback/exit decision !>
  self._errHandler.sync(self)
  _alt = self._interp.adaptivePredict(self._input,<choice.decision>,self._ctx)

>>

```

```

Sync(s) ::= "sync(<s.expecting.name>)"

```

```

ThrowNoViableAlt(t) ::= "raise NoViableAltException(self)"

```

```

TestSetInline(s) ::= <<
<s.bitsets:{bits | <if(rest(rest(bits.ttypes)))><bitsetBitfieldComparison(s, bits)><else><bitsetInlineComparison(s,
bits)><endif>}; separator=" or ">
>>

```

```

// Java language spec 15.19 - shift operators mask operands rather than overflow to 0... need range test
testShiftInRange(shiftAmount) ::= <<
((<shiftAmount>) & ~0x3f) == 0
>>

```

```

// produces smaller bytecode only when bits.ttypes contains more than two items
bitsetBitfieldComparison(s, bits) ::= <%
(<testShiftInRange(<{<offsetShiftVar(s.varName, bits.shift)>}&>) and ((1 \<< <offsetShiftVar(s.varName,
bits.shift)>) & (<bits.ttypes:{ttype | (1 \<< <offsetShiftType(ttype, bits.shift)>)); separator=" | ">)) != 0)
%>

```

```

isZero ::= [
"0":true,

```

```

default:false
]

offsetShiftVar(shiftAmount, offset) ::= <%
<if(!isZero.(offset))><shiftAmount> - <offset><else><shiftAmount><endif>
%>

offsetShiftType(shiftAmount, offset) ::= <%
<if(!isZero.(offset))><parser.name>.<shiftAmount> - <offset><else><parser.name>.<shiftAmount><endif>
%>

// produces more efficient bytecode when bits.ttypes contains at most two items
bitsetInlineComparison(s, bits) ::= <%
<bits.ttypes:{ ttype | <s.varName>==<parser.name>.<ttype> }; separator=" or ">
%>

cases(ttypes) ::= <<
if token in [<ttypes:{ t | <parser.name>.<t> }; separator=", ">]:
>>

InvokeRule(r, argExprsChunks) ::= <<
self.state = <r.stateNumber>
<if(r.labels)><r.labels:{1 | <labelref(l)> =
}><endif>self.<r.name><(if(r.ast.options.p)><r.ast.options.p><(if(argExprsChunks)><endif><endif><argExprsChu
nks)>
>>

MatchToken(m) ::= <<
self.state = <m.stateNumber>
<if(m.labels)><m.labels:{1 | <labelref(l)> = }><endif>self.match(<parser.name>.<m.name>)
>>

MatchSet(m, expr, capture) ::= "<CommonSetStuff(m, expr, capture, false)>"

MatchNotSet(m, expr, capture) ::= "<CommonSetStuff(m, expr, capture, true)>"

CommonSetStuff(m, expr, capture, invert) ::= <<
self.state = <m.stateNumber>
<if(m.labels)><m.labels:{1 | <labelref(l)> = }>self._input.LT(1)<endif>
<capture>
<if(invert)>if <m.varName> \<= 0 or <expr><else>if not(<expr><endif>:
<if(m.labels)><m.labels:{1 | <labelref(l)> = }><else> <endif>self._errHandler.recoverInline(self)
else:
  self._errHandler.reportMatch(self)
  self.consume()
>>

Wildcard(w) ::= <<

```

```

self.state = <w.stateNumber>
<if(w.labels)><w.labels:{1 | <labelref(1)> = }><endif>self.matchWildcard()
>>

// ACTION STUFF

Action(a, foo, chunks) ::= "<chunks>"

ArgAction(a, chunks) ::= "<chunks>"

SemPred(p, chunks, failChunks) ::= <<
self.state = <p.stateNumber>
if not <chunks>:
    from antlr4.error.Errors import FailedPredicateException
    raise FailedPredicateException(self, <p.predicate><if(failChunks)>, <failChunks><elseif(p.msg)>,
<p.msg><endif>)
>>

ExceptionClause(e, catchArg, catchAction) ::= <<
catch (<catchArg>) {
    <catchAction>
}
>>

// lexer actions are not associated with model objects

LexerSkipCommand() ::= "skip()"
LexerMoreCommand() ::= "more()"
LexerPopModeCommand() ::= "popMode()"

LexerTypeCommand(arg, grammar) ::= "_type = <arg>"
LexerChannelCommand(arg, grammar) ::= "_channel = <arg>"
LexerModeCommand(arg, grammar) ::= "_mode = <arg>"
LexerPushModeCommand(arg, grammar) ::= "pushMode(<arg>)"

ActionText(t) ::= "<t.text>"
ActionTemplate(t) ::= "<t.st>"
ArgRef(a) ::= "localctx.<a.name>"
LocalRef(a) ::= "localctx.<a.name>"
RetValRef(a) ::= "localctx.<a.name>"
QRetValRef(a) ::= "<ctx(a)>.<a.dict>.<a.name>"
/** How to translate $tokenLabel */
TokenRef(t) ::= "<ctx(t)>.<t.name>"
LabelRef(t) ::= "<ctx(t)>.<t.name>"
ListLabelRef(t) ::= "<ctx(t)>.<ListLabelName(t.name)>"
SetAttr(s,rhsChunks) ::= "<ctx(s)>.<s.name> = <rhsChunks>"

TokenLabelType() ::= "<file.TokenLabelType; null={Token}>"

```

```

InputSymbolType() ::= "<file.InputSymbolType; null={Token}>"

TokenPropertyRef_text(t) ::= "(None if <ctx(t)>.<t.label> is None else <ctx(t)>.<t.label>.text)"
TokenPropertyRef_type(t) ::= "(0 if <ctx(t)>.<t.label> is None else <ctx(t)>.<t.label>.type()"
TokenPropertyRef_line(t) ::= "(0 if <ctx(t)>.<t.label> is None else <ctx(t)>.<t.label>.line)"
TokenPropertyRef_pos(t) ::= "(0 if <ctx(t)>.<t.label> is None else <ctx(t)>.<t.label>.column)"
TokenPropertyRef_channel(t) ::= "(0 if (<ctx(t)>.<t.label> is None else <ctx(t)>.<t.label>.channel)"
TokenPropertyRef_index(t) ::= "(0 if <ctx(t)>.<t.label> is None else <ctx(t)>.<t.label>.tokenIndex)"
TokenPropertyRef_int(t) ::= "(0 if <ctx(t)>.<t.label> is None else int(<ctx(t)>.<t.label>.text))"

RulePropertyRef_start(r) ::= "(None if <ctx(r)>.<r.label> is None else <ctx(r)>.<r.label>.start)"
RulePropertyRef_stop(r) ::= "(None if <ctx(r)>.<r.label> is None else <ctx(r)>.<r.label>.stop)"
RulePropertyRef_text(r) ::= "(None if <ctx(r)>.<r.label> is None else
self._input.getText(<ctx(r)>.<r.label>.start,<ctx(r)>.<r.label>.stop))"
RulePropertyRef_ctx(r) ::= "<ctx(r)>.<r.label>"
RulePropertyRef_parser(r) ::= "self"

ThisRulePropertyRef_start(r) ::= "localctx.start"
ThisRulePropertyRef_stop(r) ::= "localctx.stop"
ThisRulePropertyRef_text(r) ::= "self._input.getText(localctx.start, self._input.LT(-1))"
ThisRulePropertyRef_ctx(r) ::= "localctx"
ThisRulePropertyRef_parser(r) ::= "self"

NonLocalAttrRef(s) ::= "getInvokingContext(<s.ruleIndex>).<s.name>"
SetNonLocalAttr(s, rhsChunks) ::= "getInvokingContext(<s.ruleIndex>).<s.name> = <rhsChunks>"

AddToLabelList(a) ::= "<ctx(a.label)>.<a.listName>.append(<labelref(a.label)>)"

TokenDecl(t) ::= "self.<t.name> = None # <TokenLabelType()>"
TokenTypeDecl(t) ::= "self.<t.name> = 0 # <TokenLabelType()> type"
TokenListDecl(t) ::= "self.<t.name> = list() # of <TokenLabelType()>s"
RuleContextDecl(r) ::= "self.<r.name> = None # <r.ctxName>"
RuleContextListDecl(rdecl) ::= "self.<rdecl.name> = list() # of <rdecl.ctxName>s"

ContextTokenGetterDecl(t) ::= <<
def <t.name>(self):
    return self.getToken(<parser.name>.<t.name>, 0)
>>

// should never be called
ContextTokenListGetterDecl(t) ::= <<
def <t.name>_list(self):
    return self.getTokens(<parser.name>.<t.name>)
>>

ContextTokenListIndexedGetterDecl(t) ::= <<
def <t.name>(self, i:int=None):
    if i is None:

```

```

    return self.getTokens(<parser.name>.<t.name>)
  else:
    return self.getToken(<parser.name>.<t.name>, i)
>>

ContextRuleGetterDecl(r) ::= <<
def <r.name>(self):
  return self.getTypedRuleContext(<parser.name>.<r.ctxName>,0)

>>

// should never be called
ContextRuleListGetterDecl(r) ::= <<
def <r.name>_list(self):
  return self.getTypedRuleContexts(<parser.name>.<r.ctxName>)

>>

ContextRuleListIndexedGetterDecl(r) ::= <<
def <r.name>(self, i:int=None):
  if i is None:
    return self.getTypedRuleContexts(<parser.name>.<r.ctxName>)
  else:
    return self.getTypedRuleContext(<parser.name>.<r.ctxName>,i)

>>

LexerRuleContext() ::= "RuleContext"

/** The rule context name is the rule followed by a suffix; e.g.,
 * r becomes rContext.
 */
RuleContextNameSuffix() ::= "Context"

ImplicitTokenLabel(tokenName) ::= "_<tokenName>"
ImplicitRuleLabel(ruleName) ::= "_<ruleName>"
ImplicitSetLabel(id) ::= "_tset<id>"
ListLabelName(label) ::= "<label>"

CaptureNextToken(d) ::= "<d.varName> = self._input.LT(1)"
CaptureNextTokenType(d) ::= "<d.varName> = self._input.LA(1)"

StructDecl(struct,ctorAttrs,attrs, getters,dispatchMethods,interfaces,extensionMembers) ::= <<
class <struct.name>(<if(contextSuperClass)><contextSuperClass><else>ParserRuleContext<endif>):

  def __init__(self, parser, parent:ParserRuleContext=None, invokingState:int=-1<struct.ctorAttrs:{ a | ,
<a.name><if(a.type)>:<a.type><endif>=None }):
    super().__init__(parent, invokingState)

```

```

    self.parser = parser
    <attrs:{ a | <a>}; separator="\n">
    <struct.ctorAttrs:{ a | self.<a.name> = <a.name>}; separator="\n">

<getters:{ g | <g>}; separator="\n\n">

def getRuleIndex(self):
    return <parser.name>.RULE_<struct.derivedFromName>

<if(struct.provideCopyFrom)> <! don't need copy unless we have subclasses !>
def copyFrom(self, ctx:ParserRuleContext):
    super().copyFrom(ctx)
    <struct.attrs:{ a | self.<a.name> = ctx.<a.name>}; separator="\n">

<endif>
<dispatchMethods; separator="\n">
<extensionMembers; separator="\n">

>>

AltLabelStructDecl(struct,attrs,getters,dispatchMethods) ::= <<
class <struct.name>( <currentRule.name; format="cap">Context):

    def __init__(self, parser, ctx:ParserRuleContext): # actually a <parser.name>.<currentRule.name;
format="cap">Context
        super().__init__(parser)
        <attrs:{ a | <a>}; separator="\n">
        self.copyFrom(ctx)

    <getters:{ g | <g>}; separator="\n">

    <dispatchMethods; separator="\n">

>>

ListenerDispatchMethod(method) ::= <<
def <if(method.isEnter)>enter<else>exit<endif>Rule(self, listener:ParseTreeListener):
    if hasattr( listener, "<if(method.isEnter)>enter<else>exit<endif><struct.derivedFromName; format="cap">" ):
        listener.<if(method.isEnter)>enter<else>exit<endif><struct.derivedFromName; format="cap">(self)

>>

VisitorDispatchMethod(method) ::= <<
def accept(self, visitor:ParseTreeVisitor):
    if hasattr( visitor, "visit<struct.derivedFromName; format="cap">" ):
        return visitor.visit<struct.derivedFromName; format="cap">(self)
    else:
        return visitor.visitChildren(self)

```

```

>>

AttributeDecl(d) ::= "self.<d.name> = <if(d.initValue)><d.initValue><else>None<endif>"

/** If we don't know location of label def x, use this template */
labelref(x) ::= "<if(!x.isLocal)>localctx.<endif><x.name>"

/** For any action chunk, what is correctly-typed context struct ptr? */
ctx(actionChunk) ::= "localctx"

// used for left-recursive rules
recRuleAltPredicate(ruleName,opPrec) ::= "self.precpred(self._ctx, <opPrec>)"
recRuleSetReturnAction(src,name) ::= "$<name>=<src>.<name>"
recRuleSetStopToken() ::= "self._ctx.stop = self._input.LT(-1)"

recRuleAltStartAction(ruleName, ctxName, label) ::= <<
localctx = <parser.name>.<ctxName>Context(self, _parentctx, _parentState)
<if(label)>localctx.<label> = _prevctx<endif>
self.pushNewRecursionContext(localctx, _startState, self.RULE_<ruleName>)
>>

recRuleLabeledAltStartAction(ruleName, currentAltLabel, label, isListLabel) ::= <<
localctx = <parser.name>.<currentAltLabel; format="cap">Context(self, <parser.name>.<ruleName;
format="cap">Context(self, _parentctx, _parentState))
<if(label)>
<if(isListLabel)>
localctx.<label>.append(_prevctx)
<else>
localctx.<label> = _prevctx
<endif>
<endif>
self.pushNewRecursionContext(localctx, _startState, self.RULE_<ruleName>)
>>

recRuleReplaceContext(ctxName) ::= <<
localctx = <parser.name>.<ctxName>Context(self, localctx)
self._ctx = localctx
_prevctx = localctx
>>

recRuleSetPrevCtx() ::= <<
if self._parseListeners is not None:
    self.triggerExitRuleEvent()
_prevctx = localctx
>>

```

```

LexerFile(lexerFile, lexer, namedActions) ::= <<
<fileHeader(lexerFile.grammarFileName, lexerFile.ANTLRVersion)>
from antlr4 import *
from io import StringIO
from typing.io import TextIO
import sys

<namedActions.header>

<lexer>
>>

Lexer(lexer, atn, actionFuncs, sempredFuncs, superClass) ::= <<
<if(superClass)>
if __name__ is not None and "." in __name__:
    from .<superClass> import <superClass>
else:
    from <superClass> import <superClass>

<endif>
<atn>

class <lexer.name>(<if(superClass)><superClass><else>Lexer<endif>):

    atn = ATNDeserializer().deserialize(serializedATN())

    decisionsToDFA = [ DFA(ds, i) for i, ds in enumerate(atn.decisionToState) ]

    <if(lexer.channels)>
    <lexer.channels:{c| <c> = <lexer.channels.(c)>}; separator="\n">

    <endif>
    <if(rest(lexer.modes)>
    <rest(lexer.modes):{m| <m> = <i>}; separator="\n">

    <endif>
    <lexer.tokens:{k | <k> = <lexer.tokens.(k)>}; separator="\n", wrap, anchor>

    channelNames = [ u"DEFAULT_TOKEN_CHANNEL", u"HIDDEN"<if (lexer.channels)>, <lexer.channels:{c|
u"<c>"; separator=", ", wrap, anchor><endif> ]

    modeNames = [ <lexer.modes:{m| "<m>"; separator=", ", wrap, anchor> ]

    literalNames = [ "\<INVALID>",
    <lexer.literalNames:{t | <t>}; separator=", ", wrap, anchor> ]

    symbolicNames = [ "\<INVALID>",
    <lexer.symbolicNames:{t | <t>}; separator=", ", wrap, anchor> ]

```

```

ruleNames = [ <lexer.ruleNames:{r | "<r>"}; separator=", ", wrap, anchor> ]

grammarFileName = "<lexer.grammarFileName>"

def __init__(self, input=None, output:TextIO = sys.stdout):
    super().__init__(input, output)
    self.checkVersion("<lexerFile.ANTLRVersion>")
    self._interp = LexerATNSimulator(self, self.atn, self.decisionsToDFA, PredictionContextCache())
    self._actions = None
    self._predicates = None

<namedActions.members>

<dumpActions(lexer, "", actionFuncs, sempredFuncs)>

>>

SerializedATN(model) ::= <<
<! only one segment, can be inlined !>

def serializedATN():
    with StringIO() as buf:
        buf.write("<model.serialized; wrap={ }<\n>    buf.write({}>")
        return buf.getvalue()

>>

/** Using a type to init value map, try to init a type; if not in table
 * must be an object, default value is "null".
 */
initValue(typeName) ::= <<
<pythonTypeInitMap.(typeName)>
>>

codeFileExtension() ::= ".py"

Found in path(s):
* /opt/cola/permits/1473460068_1668481512.5494933/0/antlr4-4-8-1-
jar/org/antlr/v4/tool/templates/codegen/Python3/Python3.stg
No license file was found, but licenses were detected in source scan.

/*
[The "BSD licence"]
Copyright (c) 2006 Kay Roepke
All rights reserved.

Redistribution and use in source and binary forms, with or without

```

modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

/*

This file contains the actual layout of the messages emitted by ANTLR.

The text itself is coming out of the languages/*stg files, according to the chosen locale.

This file contains the default format ANTLR uses.

*/

location(file, line, column) ::= "<file>(<line>,<column>)"

message(id, text) ::= "error <id> : <text>"

report(location, message, type) ::= "<location> : <type> <message.id> : <message.text>"

wantsSingleLineMessage() ::= "true"

Found in path(s):

* /opt/cola/permits/1473460068_1668481512.5494933/0/antlr4-4-8-1-jar/org/antlr/v4/tool/templates/messages/formats/vs2005.stg

No license file was found, but licenses were detected in source scan.

/*

[The "BSD licence"]

Copyright (c) 2006 Kay Roepke

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

/*

This file contains the actual layout of the messages emitted by ANTLR.

This file contains the default format ANTLR uses.

*/

location(file, line, column) ::= "<file>:<line>:<column>:"

message(id, text) ::= "<id> <text>"

report(location, message, type) ::= "<type>(<message.id>): <location> <message.text>"

wantsSingleLineMessage() ::= "false"

Found in path(s):

* /opt/cola/permits/1473460068_1668481512.5494933/0/antlr4-4-8-1-jar/org/antlr/v4/tool/templates/messages/formats/antlr.stg

No license file was found, but licenses were detected in source scan.

/*

* [The "BSD license"]

* Copyright (c) 2012-2016 Terence Parr

* Copyright (c) 2012-2016 Sam Harwell

* All rights reserved.

*

* Redistribution and use in source and binary forms, with or without

* modification, are permitted provided that the following conditions

* are met:

*

* 1. Redistributions of source code must retain the above copyright

```

* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products
* derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

```

```

/** How to generate rules derived from left-recursive rules.
* These rely on recRuleAltPredicate(),
* recRuleArg(), recRuleSetResultAction(), recRuleSetReturnAction()
* templates in main language.stg
*/

```

```
group LeftRecursiveRules;
```

```
recRule(ruleName, argName, primaryAlts, opAlts, setResultAction,
        userRetvals, leftRecursiveRuleRefLabels) ::=
```

```

<<
<ruleName><if(userRetvals)> returns [<userRetvals>]<endif>
: ( { } <primaryAlts:{ alt | <alt.altText> }; separator="\n    | ">
  )
  (
    <opAlts; separator="\n    | ">
  )*
;
>>

```

```

recRuleAlt(alt, precOption, opPrec, pred) ::= <<
{<pred>}?<<precOption>=<opPrec>> <alt.altText>
>>

```

Found in path(s):

```

* /opt/cola/permits/1473460068_1668481512.5494933/0/antlr4-4-8-1-
jar/org/antlr/v4/tool/templates/LeftRecursiveRules.stg

```

No license file was found, but licenses were detected in source scan.

```
/*
```

```

* [The "BSD license"]
* Copyright (c) 2012-2016 Terence Parr
* Copyright (c) 2012-2016 Sam Harwell
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products
* derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

```

```

javaTypeInitMap ::= [
  "int":"0",
  "long":"0",
  "float":"0.0f",
  "double":"0.0",
  "boolean":"false",
  "byte":"0",
  "short":"0",
  "char":"0",
  default:"null" // anything other than a primitive type is an object
]

```

```

// args must be <object-model-object>, <fields-resulting-in-STs>

```

```

ParserFile(file, parser, namedActions, contextSuperClass) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
<if(file.genPackage)>
package <file.genPackage>;
<endif>

```

```

<namedActions.header>
import org.antlr.v4.runtime.atn.*;
import org.antlr.v4.runtime.dfa.DFA;
import org.antlr.v4.runtime.*;
import org.antlr.v4.runtime.misc.*;
import org.antlr.v4.runtime.tree.*;
import java.util.List;
import java.util.Iterator;
import java.util.ArrayList;

```

```

<parser>
>>

```

```

ListenerFile(file, header, namedActions) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
<if(file.genPackage)>
package <file.genPackage>;
<endif>
<header>
import org.antlr.v4.runtime.tree.ParseTreeListener;

```

```

/**
 * This interface defines a complete listener for a parse tree produced by
 * { @link <file.parserName> }.
 */
public interface <file.grammarName>Listener extends ParseTreeListener {
    <file.listenerNames: {lname |
/**
<if(file.listenerLabelRuleNames.(lname))>
 * Enter a parse tree produced by the { @code <lname>\}
 * labeled alternative in { @link <file.parserName>#<file.listenerLabelRuleNames.(lname)>\}.
<else>
 * Enter a parse tree produced by { @link <file.parserName>#<lname>\}.
<endif>
 * @param ctx the parse tree
 */
void enter<lname; format="cap">(<file.parserName>.<lname; format="cap">Context ctx);
/**
<if(file.listenerLabelRuleNames.(lname))>
 * Exit a parse tree produced by the { @code <lname>\}
 * labeled alternative in { @link <file.parserName>#<file.listenerLabelRuleNames.(lname)>\}.
<else>
 * Exit a parse tree produced by { @link <file.parserName>#<lname>\}.
<endif>
 * @param ctx the parse tree
 */
void exit<lname; format="cap">(<file.parserName>.<lname; format="cap">Context ctx);}; separator="\n">
}

```

>>

```
BaseListenerFile(file, header, namedActions) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
<if(file.genPackage)>
package <file.genPackage>;
<endif>
<header>

import org.antlr.v4.runtime.ParserRuleContext;
import org.antlr.v4.runtime.tree.ErrorNode;
import org.antlr.v4.runtime.tree.TerminalNode;

/**
 * This class provides an empty implementation of { @link <file.grammarName>Listener},
 * which can be extended to create a listener which only needs to handle a subset
 * of the available methods.
 */
public class <file.grammarName>BaseListener implements <file.grammarName>Listener {
    <file.listenerNames:{lname |
/**
 * { @inheritDoc\}
 *
 * \<p>The default implementation does nothing.\</p>
 */
@Override public void enter<lname; format="cap">(<file.parserName>.<lname; format="cap">Context ctx) { \}
/**
 * { @inheritDoc\}
 *
 * \<p>The default implementation does nothing.\</p>
 */
@Override public void exit<lname; format="cap">(<file.parserName>.<lname; format="cap">Context ctx) { \} };
separator="\n">

/**
 * { @inheritDoc\}
 *
 * \<p>The default implementation does nothing.\</p>
 */
@Override public void enterEveryRule(ParserRuleContext ctx) { }
/**
 * { @inheritDoc\}
 *
 * \<p>The default implementation does nothing.\</p>
 */
@Override public void exitEveryRule(ParserRuleContext ctx) { }
/**
 * { @inheritDoc\}
```

```

*
* \<p>The default implementation does nothing.\</p>
*/
@Override public void visitTerminal(TerminalNode node) { }
/**
* { @inheritDoc\}
*
* \<p>The default implementation does nothing.\</p>
*/
@Override public void visitErrorNode(ErrorNode node) { }
}
>>

```

```

VisitorFile(file, header, namedActions) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
<if(file.genPackage)>
package <file.genPackage>;
<endif>
<header>
import org.antlr.v4.runtime.tree.ParseTreeVisitor;

/**
* This interface defines a complete generic visitor for a parse tree produced
* by { @link <file.parserName>}.
*
* @param \<T> The return type of the visit operation. Use { @link Void} for
* operations with no return type.
*/
public interface <file.grammarName>Visitor\<T> extends ParseTreeVisitor\<T> {
  <file.visitorNames:{ Iname |
/**
<if(file.visitorLabelRuleNames.(Iname))>
* Visit a parse tree produced by the { @code <Iname>\}
* labeled alternative in { @link <file.parserName>#\<file.visitorLabelRuleNames.(Iname)>\}.
<else>
* Visit a parse tree produced by { @link <file.parserName>#\<Iname>\}.
<endif>
* @param ctx the parse tree
* @return the visitor result
*/
T visit<Iname; format="cap">(<file.parserName>.<Iname; format="cap">Context ctx);}; separator="\n">
}
>>

```

```

BaseVisitorFile(file, header, namedActions) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
<if(file.genPackage)>
package <file.genPackage>;

```

```

<endif>
<header>
import org.antlr.v4.runtime.tree.AbstractParseTreeVisitor;

/**
 * This class provides an empty implementation of {@link <file.grammarName>Visitor},
 * which can be extended to create a visitor which only needs to handle a subset
 * of the available methods.
 *
 * @param \<T> The return type of the visit operation. Use {@link Void} for
 * operations with no return type.
 */
public class <file.grammarName>BaseVisitor<T> extends AbstractParseTreeVisitor<T> implements
<file.grammarName>Visitor<T> {
    <file.visitorNames:{lname |
/**
 * {@inheritDoc}
 *
 * \<p>The default implementation returns the result of calling
 * {@link #visitChildren} on {@code ctx}.\</p>
 */
    @Override public T visit<lname; format="cap">(<file.parserName>.<lname; format="cap">Context ctx) { return
visitChildren(ctx); \}}; separator="\n"
    }
>>

fileHeader(grammarFileName, ANTLRVersion) ::= <<
// Generated from <grammarFileName; format="java-escape"> by ANTLR <ANTLRVersion>
>>

Parser(parser, funcs, atn, sempredFuncs, superClass) ::= <<
<Parser_(ctor="parser_ctor", ...)>
>>

Parser_(parser, funcs, atn, sempredFuncs, ctor, superClass) ::= <<
@SuppressWarnings({"all", "warnings", "unchecked", "unused", "cast"})
public class <parser.name> extends <superClass; null="Parser"> {
    static { RuntimeMetaData.checkVersion("<file.ANTLRVersion>", RuntimeMetaData.VERSION); }

    protected static final DFA[] _decisionToDFA;
    protected static final PredictionContextCache _sharedContextCache =
        new PredictionContextCache();
    <if(parser.tokens)>
    public static final int
    <parser.tokens:{k | <k>=<parser.tokens.(k)>}; separator=", ", wrap, anchor>;
    <endif>
    public static final int
    <parser.rules:{r | RULE_<r.name> = <r.index>}; separator=", ", wrap, anchor>;

```

```

private static String[] makeRuleNames() {
    return new String[] {
        <parser.ruleNames:{r | "<r>"}; separator=", ", wrap, anchor>
    };
}
public static final String[] ruleNames = makeRuleNames();

<vocabulary(parser.literalNames, parser.symbolicNames)>

@Override
public String getGrammarFileName() { return "<parser.grammarFileName; format="java-escape">"; }

@Override
public String[] getRuleNames() { return ruleNames; }

@Override
public String getSerializedATN() { return _serializedATN; }

@Override
public ATN getATN() { return _ATN; }

<namedActions.members>
<parser:(ctor)()>
<funcs; separator="\n">

<if(sempredFuncs)>
public boolean sempred(RuleContext _localctx, int ruleIndex, int predIndex) {
    switch (ruleIndex) {
        <parser.sempredFuncs.values:{f}
case <f.ruleIndex>:
return <f.name>_sempred((<f.ctxType>)_localctx, predIndex);}; separator="\n">
    }
    return true;
}
<sempredFuncs.values; separator="\n">
<endif>

<atn>
}
>>

vocabulary(literalNames, symbolicNames) ::= <<
private static String[] makeLiteralNames() {
    return new String[] {
        <literalNames:{t | <t>}; null="null", separator=", ", wrap, anchor>
    };
}
private static final String[] _LITERAL_NAMES = makeLiteralNames();

```

```

private static String[] makeSymbolicNames() {
    return new String[] {
        <symbolicNames: {t | <t>}; null="null", separator=", ", wrap, anchor>
    };
}
private static final String[] _SYMBOLIC_NAMES = makeSymbolicNames();
public static final Vocabulary VOCABULARY = new VocabularyImpl(_LITERAL_NAMES,
    _SYMBOLIC_NAMES);

/**
 * @deprecated Use { @link #VOCABULARY } instead.
 */
@Deprecated
public static final String[] tokenNames;
static {
    tokenNames = new String[_SYMBOLIC_NAMES.length];
    for (int i = 0; i < tokenNames.length; i++) {
        tokenNames[i] = VOCABULARY.getLiteralName(i);
        if (tokenNames[i] == null) {
            tokenNames[i] = VOCABULARY.getSymbolicName(i);
        }

        if (tokenNames[i] == null) {
            tokenNames[i] = "\<INVALID>";
        }
    }
}

@Override
@Deprecated
public String[] getTokenNames() {
    return tokenNames;
}

@Override

public Vocabulary getVocabulary() {
    return VOCABULARY;
}
>>

dumpActions(recog, argFuncs, actionFuncs, sempredFuncs) ::= <<
<if(actionFuncs)>
@Override
public void action(RuleContext _localctx, int ruleIndex, int actionIndex) {
    switch (ruleIndex) {
        <recog.actionFuncs.values: {f}
    case <f.ruleIndex>:

```

```

    <f.name>_action((<f.ctxType>)_localctx, actionIndex);
    break;}; separator="\n">
    }
    }
    <actionFuncs.values; separator="\n">
    <endif>
    <if(sempredFuncs)>
    @Override
    public boolean sempred(RuleContext _localctx, int ruleIndex, int predIndex) {
    switch (ruleIndex) {
    <recog.sempredFuncs.values: {f|
    case <f.ruleIndex>:
    return <f.name>_sempred((<f.ctxType>)_localctx, predIndex);}; separator="\n">
    }
    return true;
    }
    <sempredFuncs.values; separator="\n">
    <endif>
    >>

    parser_ctor(p) ::= <<
    public <p.name>(TokenStream input) {
    super(input);
    _interp = new ParserATNSimulator(this, _ATN, _decisionToDFA, _sharedContextCache);
    }
    >>

    /* This generates a private method since the actionIndex is generated, making an
    * overriding implementation impossible to maintain.
    */
    RuleActionFunction(r, actions) ::= <<
    private void <r.name>_action(<r.ctxType> _localctx, int actionIndex) {
    switch (actionIndex) {
    <actions: {index|
    case <index>:
    <actions.(index)>
    break;}; separator="\n">
    }
    }
    >>

    /* This generates a private method since the predIndex is generated, making an
    * overriding implementation impossible to maintain.
    */
    RuleSempredFunction(r, actions) ::= <<
    private boolean <r.name>_sempred(<r.ctxType> _localctx, int predIndex) {
    switch (predIndex) {
    <actions: {index|

```

```

case <index>:
return <actions.(index)>;}; separator="\n">
}
return true;
}
>>

```

```

RuleFunction(currentRule,args,code,locals,ruleCtx,altLabelCtxs,namedActions,finallyAction,postamble,exceptions)
::= <<

```

```

<ruleCtx>
<altLabelCtxs:{l | <altLabelCtxs.(l)>}; separator="\n">

<if(currentRule.modifiers)><currentRule.modifiers:{f | <f> }><else>public final <endif><currentRule.ctxType>
<currentRule.name><(args; separator=",") throws RecognitionException {
<currentRule.ctxType> _localctx = new <currentRule.ctxType>(_ctx, getState())<currentRule.args:{a | ,
<a.name>}>>);
enterRule(_localctx, <currentRule.startState>, RULE_<currentRule.name>);
<namedActions.init>
<locals; separator="\n">
try {
<if(currentRule.hasLookaheadBlock)>
int _alt;
<endif>
<code>
<postamble; separator="\n">
<namedActions.after>
}
<if(exceptions)>
<exceptions; separator="\n">
<else>
catch (RecognitionException re) {
_localctx.exception = re;
_errHandler.reportError(this, re);
_errHandler.recover(this, re);
}
<endif>
finally {
<finallyAction>
exitRule();
}
return _localctx;
}
>>

```

```

LeftRecursiveRuleFunction(currentRule,args,code,locals,ruleCtx,altLabelCtxs,
namedActions,finallyAction,postamble) ::=
<<

```

```

<ruleCtx>
<altLabelCtxs:{l | <altLabelCtxs.(l)>}; separator="\n">

<if(currentRule.modifiers)><currentRule.modifiers:{f | <f> }><else>public final <endif><currentRule.ctxType>
<currentRule.name><(<args; separator=", ">) throws RecognitionException {
return <currentRule.name>(0<currentRule.args:{a | , <a.name>}>);
}

```

```

private <currentRule.ctxType> <currentRule.name>(int _p<args:{a | , <a>}>) throws RecognitionException {
ParserRuleContext _parentctx = _ctx;
int _parentState = getState();
<currentRule.ctxType> _localctx = new <currentRule.ctxType>(_ctx, _parentState<currentRule.args:{a | ,
<a.name>}>);
<currentRule.ctxType> _prevctx = _localctx;
int _startState = <currentRule.startState>;
enterRecursionRule(_localctx, <currentRule.startState>, RULE_<currentRule.name>, _p);
<namedActions.init>
<locals; separator="\n">
try {
<if(currentRule.hasLookaheadBlock)>
int _alt;
<endif>
<code>
<postamble; separator="\n">
<namedActions.after>
}
catch (RecognitionException re) {
_localctx.exception = re;
_errHandler.reportError(this, re);
_errHandler.recover(this, re);
}
finally {
<finallyAction>
unrollRecursionContexts(_parentctx);
}
return _localctx;
}
>>

```

```

CodeBlockForOuterMostAlt(currentOuterMostAltCodeBlock, locals, preamble, ops) ::= <<
<if(currentOuterMostAltCodeBlock.altLabel)>_localctx = new <currentOuterMostAltCodeBlock.altLabel;
format="cap">Context(_localctx);<endif>
enterOuterAlt(_localctx, <currentOuterMostAltCodeBlock.alt.altNum>);
<CodeBlockForAlt(currentAltCodeBlock=currentOuterMostAltCodeBlock, ...)>
>>

```

```

CodeBlockForAlt(currentAltCodeBlock, locals, preamble, ops) ::= <<

```

```

{
<locals; separator="\n">
<preamble; separator="\n">
<ops; separator="\n">
}
>>

```

```

LL1AltBlock(choice, preamble, alts, error) ::= <<
setState(<choice.stateNumber>);
_errHandler.sync(this);
<if(choice.label)><labelref(choice.label)> = _input.LT(1);<endif>
<preamble; separator="\n">
switch (_input.LA(1)) {
<choice.altLook,alts:{look,alt| <cases(ttypes=look)>
<alt>
break;}; separator="\n">
default:
<error>
}
>>

```

```

LL1OptionalBlock(choice, alts, error) ::= <<
setState(<choice.stateNumber>);
_errHandler.sync(this);
switch (_input.LA(1)) {
<choice.altLook,alts:{look,alt| <cases(ttypes=look)>
<alt>
break;}; separator="\n">
default:
break;
}
>>

```

```

LL1OptionalBlockSingleAlt(choice, expr, alts, preamble, error, followExpr) ::= <<
setState(<choice.stateNumber>);
_errHandler.sync(this);
<preamble; separator="\n">
if (<expr>) {
<alts; separator="\n">
}
<!else if ( !(<followExpr> ) <error>!>
>>

```

```

LL1StarBlockSingleAlt(choice, loopExpr, alts, preamble, iteration) ::= <<
setState(<choice.stateNumber>);
_errHandler.sync(this);
<preamble; separator="\n">
while (<loopExpr>) {

```

```

<alts; separator="\n">
setState(<choice.loopBackStateNumber>);
_errHandler.sync(this);
<iteration>
}
>>

LL1PlusBlockSingleAlt(choice, loopExpr, alts, preamble, iteration) ::= <<
setState(<choice.blockStartStateNumber>); <! alt block decision !>
_errHandler.sync(this);
<preamble; separator="\n">
do {
<alts; separator="\n">
setState(<choice.stateNumber>); <! loopback/exit decision !>
_errHandler.sync(this);
<iteration>
} while ( <loopExpr> );
>>

// LL(*) stuff

AltBlock(choice, preamble, alts, error) ::= <<
setState(<choice.stateNumber>);
_errHandler.sync(this);
<if(choice.label)><labelref(choice.label)> = _input.LT(1);<endif>
<preamble; separator="\n">
switch ( getInterpreter().adaptivePredict(_input,<choice.decision>,_ctx) ) {
<alts:{alt |
case <i>:
<alt>
break;}; separator="\n">
}
>>

OptionalBlock(choice, alts, error) ::= <<
setState(<choice.stateNumber>);
_errHandler.sync(this);
switch ( getInterpreter().adaptivePredict(_input,<choice.decision>,_ctx) ) {
<alts:{alt |
case <i><if(!choice.ast.greedy)>+1<endif>:
<alt>
break;}; separator="\n">
}
>>

StarBlock(choice, alts, sync, iteration) ::= <<
setState(<choice.stateNumber>);
_errHandler.sync(this);

```

```

_alt = getInterpreter().adaptivePredict(_input,<choice.decision>,_ctx);
while ( _alt!=<choice.exitAlt> && _alt!=org.antlr.v4.runtime.atn.ATN.INVALID_ALT_NUMBER ) {
if ( _alt==1<if(!choice.ast.greedy)>+1<endif> ) {
<iteration>
<alts> <! should only be one !>
}
setState(<choice.loopBackStateNumber>);
_errHandler.sync(this);
_alt = getInterpreter().adaptivePredict(_input,<choice.decision>,_ctx);
}
>>

```

```

PlusBlock(choice, alts, error) ::= <<
setState(<choice.blockStartStateNumber>); <! alt block decision !>
_errHandler.sync(this);
_alt = 1<if(!choice.ast.greedy)>+1<endif>;
do {
switch ( _alt ) {
<alts:{alt|
case <i><if(!choice.ast.greedy)>+1<endif>:
<alt>
break;}; separator="\n">
default:
<error>
}
setState(<choice.loopBackStateNumber>); <! loopback/exit decision !>
_errHandler.sync(this);
_alt = getInterpreter().adaptivePredict(_input,<choice.decision>,_ctx);
} while ( _alt!=<choice.exitAlt> && _alt!=org.antlr.v4.runtime.atn.ATN.INVALID_ALT_NUMBER );
>>

```

```

Sync(s) ::= "sync(<s.expecting.name>);"

```

```

ThrowNoViableAlt(t) ::= "throw new NoViableAltException(this);"

```

```

TestSetInline(s) ::= <<
<s.bitsets:{bits | <if(rest(rest(bits.ttypes)))><bitsetBitfieldComparison(s, bits)><else><bitsetInlineComparison(s,
bits)><endif>}; separator=" || ">
>>

```

```

// Java language spec 15.19 - shift operators mask operands rather than overflow to 0... need range test
testShiftInRange(shiftAmount) ::= <<
((<shiftAmount>) & ~0x3f) == 0
>>

```

```

// produces smaller bytecode only when bits.ttypes contains more than two items
bitsetBitfieldComparison(s, bits) ::= <%
(<testShiftInRange({<offsetShift(s.varName, bits.shift)>})> && ((1L \<< <offsetShift(s.varName, bits.shift)>) &

```

```

(<bits.ttypes:{ ttype | (1L \<< <offsetShift(ttype, bits.shift)>)}; separator=" | ">) != 0
%>

isZero ::= [
"0":true,
default:false
]

offsetShift(shiftAmount, offset) ::= <%
<if(!isZero.(offset))><shiftAmount> - <offset><else><shiftAmount><endif>
%>

// produces more efficient bytecode when bits.ttypes contains at most two items
bitsetInlineComparison(s, bits) ::= <%
<bits.ttypes:{ ttype | <s.varName>===<ttype>}; separator=" || ">
%>

cases(ttypes) ::= <<
<ttypes:{ t | case <t>:}; separator="\n">
>>

InvokeRule(r, argExprsChunks) ::= <<
setState(<r.stateNumber>);
<if(r.labels)><r.labels:{1 | <labelref(l)> =
}><endif><r.name><(<if(r.ast.options.p)><r.ast.options.p><if(argExprsChunks)><endif><endif><argExprsChunks>
);
>>

MatchToken(m) ::= <<
setState(<m.stateNumber>);
<if(m.labels)><m.labels:{1 | <labelref(l)> = }><endif>match(<m.name>);
>>

MatchSet(m, expr, capture) ::= "<CommonSetStuff(m, expr, capture, false)>"

MatchNotSet(m, expr, capture) ::= "<CommonSetStuff(m, expr, capture, true)>"

CommonSetStuff(m, expr, capture, invert) ::= <<
setState(<m.stateNumber>);
<if(m.labels)><m.labels:{1 | <labelref(l)> = }>_input.LT(1);<endif>
<capture>
if ( <if(invert)><m.varName> \<= 0 || <else>!<endif><(<expr>) ) {
<if(m.labels)><m.labels:{1 | <labelref(l)> = (Token)}><endif>_errHandler.recoverInline(this);
}
else {
if ( _input.LA(1)==Token.EOF ) matchedEOF = true;
_errHandler.reportMatch(this);
consume();
}
}

```

```

}
>>

Wildcard(w) ::= <<
setState(<w.stateNumber>);
<if(w.labels)><w.labels:{1 | <labelref(l)> = }><endif>matchWildcard();
>>

// ACTION STUFF

Action(a, foo, chunks) ::= "<chunks>"

ArgAction(a, chunks) ::= "<chunks>"

SemPred(p, chunks, failChunks) ::= <<
setState(<p.stateNumber>);
if (!(<chunks>)) throw new FailedPredicateException(this, <p.predicate><if(failChunks)>,
<failChunks><elseif(p.msg)>, <p.msg><endif>);
>>

ExceptionClause(e, catchArg, catchAction) ::= <<
catch (<catchArg>) {
  <catchAction>
}
>>

// lexer actions are not associated with model objects

LexerSkipCommand() ::= "skip();"
LexerMoreCommand() ::= "more();"
LexerPopModeCommand() ::= "popMode();"

LexerTypeCommand(arg, grammar) ::= "_type = <arg>";
LexerChannelCommand(arg, grammar) ::= "_channel = <arg>";
LexerModeCommand(arg, grammar) ::= "_mode = <arg>";
LexerPushModeCommand(arg, grammar) ::= "pushMode(<arg>);"

ActionText(t) ::= "<t.text>"
ActionTemplate(t) ::= "<t.st>"
ArgRef(a) ::= "_localctx.<a.name>"
LocalRef(a) ::= "_localctx.<a.name>"
RetValRef(a) ::= "_localctx.<a.name>"
QRetValRef(a) ::= "<ctx(a)>.<a.dict>.<a.name>"
/** How to translate $tokenLabel */
TokenRef(t) ::= "<ctx(t)>.<t.name>"
LabelRef(t) ::= "<ctx(t)>.<t.name>"
ListLabelRef(t) ::= "<ctx(t)>.<ListLabelName(t.name)>"
SetAttr(s,rhsChunks) ::= "<ctx(s)>.<s.name> = <rhsChunks>";

```

```

TokenLabelType() ::= "<file.TokenLabelType; null={Token}>"
InputSymbolType() ::= "<file.InputSymbolType; null={Token}>"

TokenPropertyRef_text(t) ::= "<ctx(t)>.<t.label>!<null?<ctx(t)>.<t.label>.getText():null)"
TokenPropertyRef_type(t) ::= "<ctx(t)>.<t.label>!<null?<ctx(t)>.<t.label>.getType():0)"
TokenPropertyRef_line(t) ::= "<ctx(t)>.<t.label>!<null?<ctx(t)>.<t.label>.getLine():0)"
TokenPropertyRef_pos(t) ::= "<ctx(t)>.<t.label>!<null?<ctx(t)>.<t.label>.getCharPositionInLine():0)"
TokenPropertyRef_channel(t) ::= "<ctx(t)>.<t.label>!<null?<ctx(t)>.<t.label>.getChannel():0)"
TokenPropertyRef_index(t) ::= "<ctx(t)>.<t.label>!<null?<ctx(t)>.<t.label>.getTokenIndex():0)"
TokenPropertyRef_int(t) ::= "<ctx(t)>.<t.label>!<null?Integer.valueOf(<ctx(t)>.<t.label>.getText():0)"

RulePropertyRef_start(r) ::= "<ctx(r)>.<r.label>!<null?<ctx(r)>.<r.label>.start):null)"
RulePropertyRef_stop(r) ::= "<ctx(r)>.<r.label>!<null?<ctx(r)>.<r.label>.stop):null)"
RulePropertyRef_text(r) ::=
"<ctx(r)>.<r.label>!<null?_input.getText(<ctx(r)>.<r.label>.start,<ctx(r)>.<r.label>.stop):null)"
RulePropertyRef_ctx(r) ::= "<ctx(r)>.<r.label>"
RulePropertyRef_parser(r) ::= "this"

ThisRulePropertyRef_start(r) ::= "_localctx.start"
ThisRulePropertyRef_stop(r) ::= "_localctx.stop"
ThisRulePropertyRef_text(r) ::= "_input.getText(_localctx.start, _input.LT(-1))"
ThisRulePropertyRef_ctx(r) ::= "_localctx"
ThisRulePropertyRef_parser(r) ::= "this"

NonLocalAttrRef(s) ::= "(<<s.ruleName; format=\"cap\">Context)getInvokingContext(<s.ruleIndex>)).<s.name>"
SetNonLocalAttr(s, rhsChunks) ::=
"(<<s.ruleName; format=\"cap\">Context)getInvokingContext(<s.ruleIndex>)).<s.name> = <rhsChunks>);"

AddToLabelList(a) ::= "<ctx(a.label)>.<a.listName>.add(<labelref(a.label)>);"

TokenDecl(t) ::= "<TokenLabelType()> <t.name>"
TokenTypeDecl(t) ::= "int <t.name>);"
TokenListDecl(t) ::= "List<Token> <t.name> = new ArrayList<Token>()"
RuleContextDecl(r) ::= "<r.ctxName> <r.name>"
RuleContextListDecl(rdecl) ::= "List<<rdecl.ctxName>> <rdecl.name> = new ArrayList<<rdecl.ctxName>>()"

ContextTokenGetterDecl(t) ::=
"public TerminalNode <t.name>() { return getToken(<parser.name>.<t.name>, 0); }"
ContextTokenListGetterDecl(t) ::=
"public List<TerminalNode> <t.name>() { return getTokens(<parser.name>.<t.name>); }"
ContextTokenListIndexedGetterDecl(t) ::= <<
public TerminalNode <t.name>(int i) {
return getToken(<parser.name>.<t.name>, i);
}
>>

ContextRuleGetterDecl(r) ::= <<
public <r.ctxName> <r.name>() {

```

```

return getRuleContext(<r.ctxName>.class,0);
}
>>
ContextRuleListGetterDecl(r) ::= <<
public List<<r.ctxName>> <r.name>() {
return getRuleContexts(<r.ctxName>.class);
}
>>
ContextRuleListIndexedGetterDecl(r) ::= <<
public <r.ctxName> <r.name>(int i) {
return getRuleContext(<r.ctxName>.class,i);
}
>>

LexerRuleContext() ::= "RuleContext"

/** The rule context name is the rule followed by a suffix; e.g.,
 * r becomes rContext.
 */
RuleContextNameSuffix() ::= "Context"

ImplicitTokenLabel(tokenName) ::= "<tokenName>"
ImplicitRuleLabel(ruleName) ::= "<ruleName>"
ImplicitSetLabel(id) ::= "_tset<id>"
ListLabelName(label) ::= "<label>"

CaptureNextToken(d) ::= "<d.varName> = _input.LT(1);"
CaptureNextTokenType(d) ::= "<d.varName> = _input.LA(1);"

StructDecl(struct,ctorAttrs,attrs, getters,dispatchMethods,interfaces,extensionMembers)
::= <<
public static class <struct.name> extends
<if(contextSuperClass)><contextSuperClass><else>ParserRuleContext<endif><if(interfaces)> implements
<interfaces; separator=", "><endif> {
<attrs:{ a | public <a>;}; separator="\n">
<getters:{ g | <g>;}; separator="\n">
<if(ctorAttrs)>public <struct.name>(ParserRuleContext parent, int invokingState) { super(parent, invokingState);
}<endif>
public <struct.name>(ParserRuleContext parent, int invokingState<ctorAttrs:{ a | , <a>>}) {
super(parent, invokingState);
<struct.ctorAttrs:{ a | this.<a.name> = <a.name>;}; separator="\n">
}
@Override public int getRuleIndex() { return RULE_<struct.derivedFromName>; }
<if(struct.provideCopyFrom)> <! don't need copy unless we have subclasses !>
public <struct.name>() { }
public void copyFrom(<struct.name> ctx) {
super.copyFrom(ctx);
<struct.attrs:{ a | this.<a.name> = ctx.<a.name>;}; separator="\n">

```

```

}
<endif>
<dispatchMethods; separator="\n">
<extensionMembers; separator="\n">
}
>>

AltLabelStructDecl(struct,attrs, getters,dispatchMethods) ::= <<
public static class <struct.name> extends <currentRule.name; format="cap">Context {
<attrs:{a | public <a>;}; separator="\n">
<getters:{g | <g>;}; separator="\n">
public <struct.name>(<currentRule.name; format="cap">Context ctx) { copyFrom(ctx); }
<dispatchMethods; separator="\n">
}
>>

ListenerDispatchMethod(method) ::= <<
@Override
public void <if(method.isEnter)>enter<else>exit<endif>Rule(ParseTreeListener listener) {
if ( listener instanceof <parser.grammarName>Listener )
((<parser.grammarName>Listener)listener).<if(method.isEnter)>enter<else>exit<endif><struct.derivedFromName;
format="cap">(this);
}
>>

VisitorDispatchMethod(method) ::= <<
@Override
public <T> T accept(ParseTreeVisitor<? extends T> visitor) {
if ( visitor instanceof <parser.grammarName>Visitor ) return ((<parser.grammarName>Visitor<? extends
T>)visitor).visit<struct.derivedFromName; format="cap">(this);
else return visitor.visitChildren(this);
}
>>

AttributeDecl(d) ::= "<d.type> <d.name><if(d.initValue)> = <d.initValue><endif>"

/** If we don't know location of label def x, use this template */
labelref(x) ::= "<if(!x.isLocal)>((<x.ctx.name>)_localctx).<endif><x.name>"

/** For any action chunk, what is correctly-typed context struct ptr? */
ctx(actionChunk) ::= "((<actionChunk.ctx.name>)_localctx)"

// used for left-recursive rules
recRuleAltPredicate(ruleName,opPrec) ::= "precpred(_ctx, <opPrec>)"
recRuleSetReturnAction(src,name) ::= "$<name>=<src>.<name>";
recRuleSetStopToken() ::= "_ctx.stop = _input.LT(-1);"

recRuleAltStartAction(ruleName, ctxName, label, isListLabel) ::= <<

```

```

_localctx = new <ctxName>Context(_parentctx, _parentState);
<if(label)>
<if(isListLabel)>
_localctx.<label>.add(_prevctx);
<else>
_localctx.<label> = _prevctx;
<endif>
<endif>
<if(label)>_localctx.<label> = _prevctx;<endif>
pushNewRecursionContext(_localctx, _startState, RULE_<ruleName>);
>>

recRuleLabeledAltStartAction(ruleName, currentAltLabel, label, isListLabel) ::= <<
_localctx = new <currentAltLabel; format="cap">Context(new <ruleName; format="cap">Context(_parentctx,
_parentState));
<if(label)>
<if(isListLabel)>
((<currentAltLabel; format="cap">Context)_localctx).<label>.add(_prevctx);
<else>
((<currentAltLabel; format="cap">Context)_localctx).<label> = _prevctx;
<endif>
<endif>
pushNewRecursionContext(_localctx, _startState, RULE_<ruleName>);
>>

recRuleReplaceContext(ctxName) ::= <<
_localctx = new <ctxName>Context(_localctx);
_ctx = _localctx;
_prevctx = _localctx;
>>

recRuleSetPrevCtx() ::= <<
if ( _parseListeners!=null ) triggerExitRuleEvent();
_prevctx = _localctx;
>>

LexerFile(lexerFile, lexer, namedActions) ::= <<
<fileHeader(lexerFile.grammarFileName, lexerFile.ANTLRVersion)>
<if(lexerFile.genPackage)>
package <lexerFile.genPackage>;
<endif>
<namedActions.header>
import org.antlr.v4.runtime.Lexer;
import org.antlr.v4.runtime.CharStream;
import org.antlr.v4.runtime.Token;
import org.antlr.v4.runtime.TokenStream;
import org.antlr.v4.runtime.*;

```

```

import org.antlr.v4.runtime.atn.*;
import org.antlr.v4.runtime.dfa.DFA;
import org.antlr.v4.runtime.misc.*;

<lexer>
>>

Lexer(lexer, atn, actionFuncs, sempredFuncs, superClass) ::= <<
@SuppressWarnings({"all", "warnings", "unchecked", "unused", "cast"})
public class <lexer.name> extends <superClass; null="Lexer"> {
    static { RuntimeMetaData.checkVersion("<lexerFile.ANTLRVersion>", RuntimeMetaData.VERSION); }

    protected static final DFA[] _decisionToDFA;
    protected static final PredictionContextCache _sharedContextCache =
        new PredictionContextCache();
    <if(lexer.tokens)>
    public static final int
        <lexer.tokens:{k | <k>=<lexer.tokens.(k)>}; separator=", ", wrap, anchor>;
    <endif>
    <if(lexer.channels)>
    public static final int
        <lexer.channels:{c | <c>=<lexer.channels.(c)>}; separator=", ", wrap, anchor>;
    <endif>
    <if(rest(lexer.modes)>
    public static final int
        <rest(lexer.modes):{m | <m>=<i>}; separator=", ", wrap, anchor>;
    <endif>
    public static String[] channelNames = {
        "DEFAULT_TOKEN_CHANNEL", "HIDDEN"<if (lexer.channels)>, <lexer.channels:{c | "<c>"}; separator=", ",
wrap, anchor><endif>
    };

    public static String[] modeNames = {
        <lexer.modes:{m | "<m>"}; separator=", ", wrap, anchor>
    };

    private static String[] makeRuleNames() {
        return new String[] {
            <lexer.ruleNames:{r | "<r>"}; separator=", ", wrap, anchor>
        };
    }
    public static final String[] ruleNames = makeRuleNames();

    <vocabulary(lexer.literalNames, lexer.symbolicNames)>

    <namedActions.members>

    public <lexer.name>(CharStream input) {

```

```

super(input);
_interp = new LexerATNSimulator(this, _ATN, _decisionToDFA, _sharedContextCache);
}

@Override
public String getGrammarFileName() { return "<lexer.grammarFileName>"; }

@Override
public String[] getRuleNames() { return ruleNames; }

@Override
public String getSerializedATN() { return _serializedATN; }

@Override
public String[] getChannelNames() { return channelNames; }

@Override
public String[] getModeNames() { return modeNames; }

@Override
public ATN getATN() { return _ATN; }

<dumpActions(lexer, "", actionFuncs, sempredFuncs)>
<atn>
}
>>

SerializedATN(model) ::= <<
<if(rest(model.segments))>
<! requires segmented representation !>
private static final int _serializedATNSegments = <length(model.segments)>;
<model.segments:{segment|private static final String _serializedATNSegment<i0> =
"<segment; wrap={"+<\n><\t>"}>"; separator="\n">
public static final String _serializedATN = Utils.join(
new String[] {
<model.segments:{segment | _serializedATNSegment<i0>}; separator=",\n">
},
""
);
<else>
<! only one segment, can be inlined !>
public static final String _serializedATN =
"<model.serialized; wrap={"+<\n><\t>"}>";
<endif>
public static final ATN _ATN =
new ATNDeserializer().deserialize(_serializedATN.toCharArray());
static {
_interp = new DFA[_ATN.getNumberOfDecisions()];

```

```

for (int i = 0; i < _ATN.getNumberOfDecisions(); i++) {
    _decisionToDFA[i] = new DFA(_ATN.getDecisionState(i), i);
}
<! org.antlr.v4.tool.DOTGenerator dot = new org.antlr.v4.tool.DOTGenerator(null);!>
<! System.out.println(dot.getDOT(_ATN.decisionToState.get(0), ruleNames, false));!>
<! System.out.println(dot.getDOT(_ATN.ruleToStartState[2], ruleNames, false));!>
}
>>

```

```

/** Using a type to init value map, try to init a type; if not in table

```

```

* must be an object, default value is "null".

```

```

*/

```

```

initValue(typeName) ::= <<

```

```

<javaTypeInitMap.(typeName)>

```

```

>>

```

```

codeFileExtension() ::= ".java"

```

```

Found in path(s):

```

```

* /opt/cola/permits/1473460068_1668481512.5494933/0/antlr4-4-8-1-

```

```

jar/org/antlr/v4/tool/templates/codegen/Java/Java.stg

```

```

No license file was found, but licenses were detected in source scan.

```

```

/*

```

```

* [The "BSD license"]

```

```

* Copyright (c) 2012-2016 Terence Parr

```

```

* Copyright (c) 2012-2016 Sam Harwell

```

```

* All rights reserved.

```

```

*

```

```

* Redistribution and use in source and binary forms, with or without

```

```

* modification, are permitted provided that the following conditions

```

```

* are met:

```

```

*

```

```

* 1. Redistributions of source code must retain the above copyright

```

```

* notice, this list of conditions and the following disclaimer.

```

```

* 2. Redistributions in binary form must reproduce the above copyright

```

```

* notice, this list of conditions and the following disclaimer in the

```

```

* documentation and/or other materials provided with the distribution.

```

```

* 3. The name of the author may not be used to endorse or promote products

```

```

* derived from this software without specific prior written permission.

```

```

*

```

```

* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR

```

```

* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES

```

```

* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.

```

```

* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,

```

```

* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT

```

```

* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,

```

```

* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY

```

```

* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

```

```

phpTypeInitMap ::= [
  "int":"0",
  "long":"0",
  "float":"0.0",
  "double":"0.0",
  "boolean":"false",
  default:"null"
]

```

```

// args must be <object-model-object>, <fields-resulting-in-STs>

```

```

ParserFile(file, parser, namedActions, contextSuperClass) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
<parser>
>>

```

```

ListenerFile(file, header, namedActions) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
<if(file.genPackage)>
namespace <file.genPackage>;
<endif>
<header>
use Antlr\Antlr4\Runtime\Tree\ParseTreeListener;

```

```

/**
 * This interface defines a complete listener for a parse tree produced by
 * { @see <file.parserName> }.
 */
interface <file.grammarName>Listener extends ParseTreeListener {
  <file.listenerNames:{lname |
/**
<if(file.listenerLabelRuleNames.(lname))>
 * Enter a parse tree produced by the `<lname>`
 * labeled alternative in { @see <file.parserName>::<file.listenerLabelRuleNames.(lname)>() }.
<else>
 * Enter a parse tree produced by { @see <file.parserName>::<lname>() }.
<endif>
 * @param $context The parse tree.
 */
public function enter<lname; format="cap">(Context\<lname; format="cap">Context $context) : void;
/**
<if(file.listenerLabelRuleNames.(lname))>
 * Exit a parse tree produced by the `<lname>` labeled alternative

```

```

* in { @see <file.parserName>::<file.listenerLabelRuleNames.<lname>>()}.
<else>
* Exit a parse tree produced by { @see <file.parserName>::<lname>()}.
<endif>
* @param $context The parse tree.
*/
public function exit<lname; format="cap">(Context\<lname; format="cap">Context $context) : void;
separator="\n"
}
>>

BaseListenerFile(file, header, namedActions) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
<if(file.genPackage)>
namespace <file.genPackage>;
<endif>
<header>

use Antlr\Antlr4\Runtime\ParserRuleContext;
use Antlr\Antlr4\Runtime\Tree\ErrorNode;
use Antlr\Antlr4\Runtime\Tree\TerminalNode;

/**
 * This class provides an empty implementation of { @see <file.grammarName>Listener},
 * which can be extended to create a listener which only needs to handle a subset
 * of the available methods.
 */
class <file.grammarName>BaseListener implements <file.grammarName>Listener
{
    <file.listenerNames: {lname |
/**
 * { @inheritdoc}
 *
 * The default implementation does nothing.
 */
public function enter<lname; format="cap">(Context\<lname; format="cap">Context $context) : void {}

/**
 * { @inheritdoc}
 *
 * The default implementation does nothing.
 */
public function exit<lname; format="cap">(Context\<lname; format="cap">Context $context) : void {}};
separator="\n"

/**
 * { @inheritdoc}
 *

```

```

* The default implementation does nothing.
*/
public function enterEveryRule(ParserRuleContext $context) : void {}

/**
 * {@inheritdoc}
 *
 * The default implementation does nothing.
 */
public function exitEveryRule(ParserRuleContext $context) : void {}

/**
 * {@inheritdoc}
 *
 * The default implementation does nothing.
 */
public function visitTerminal(TerminalNode $node) : void {}

/**
 * {@inheritdoc}
 *
 * The default implementation does nothing.
 */
public function visitErrorNode(ErrorNode $node) : void {}
}
>>

```

```

VisitorFile(file, header, namedActions) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
<if(file.genPackage)>
namespace <file.genPackage>;
<endif>

<header>
use Antlr\Antlr4\Runtime\Tree\ParseTreeVisitor;

/**
 * This interface defines a complete generic visitor for a parse tree produced by {@see <file.parserName>}.
 */
interface <file.grammarName>Visitor extends ParseTreeVisitor
{
<file.visitorNames>: { Iname |
/**
<if(file.visitorLabelRuleNames.(Iname))>
 * Visit a parse tree produced by the `<Iname>` labeled alternative
 * in {@see <file.parserName>::<file.visitorLabelRuleNames.(Iname)>()}.
<else>
 * Visit a parse tree produced by {@see <file.parserName>::<Iname>()}.

```

```

<endif>
*
* @param Context\<Iname; format="cap">Context $context The parse tree.
*
* @return mixed The visitor result.
*/
public function visit<Iname; format="cap">(Context\<Iname; format="cap">Context $context);};
separator="\n\n">
}
>>

```

```

BaseVisitorFile(file, header, namedActions) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
<if(file.genPackage)>
namespace <file.genPackage>;
<endif>
<header>
use Antlr\Antlr4\Runtime\Tree\AbstractParseTreeVisitor;

/**
 * This class provides an empty implementation of { @see <file.grammarName>Visitor },
 * which can be extended to create a visitor which only needs to handle a subset
 * of the available methods.
 */
class <file.grammarName>BaseVisitor extends AbstractParseTreeVisitor implements <file.grammarName>Visitor
{
    <file.visitorNames:{ Iname |
/**
 * { @inheritdoc\}
 *
 * The default implementation returns the result of calling
 * { @see self::visitChildren()\} on `context`.
 */
public function visit<Iname; format="cap">(Context\<Iname; format="cap">Context $context)
{
    return $this->visitChildren($context);
\}}; separator="\n\n">
}
>>

```

```

fileHeader(grammarFileName, ANTLRVersion) ::= <<
\<?php

/*
 * Generated from <grammarFileName> by ANTLR <ANTLRVersion>
 */

>>

```

```

Parser(parser, funcs, atn, sempredFuncs, superClass) ::= <<
<Parser_(ctor="parser_ctor", ...)>
>>

```

```

Parser_(parser, funcs, atn, sempredFuncs, ctor, superClass) ::= <<
namespace<if(file.genPackage)> <file.genPackage><endif> {
<if(namedActions.header)><namedActions.header><endif>
use Antlr\Antlr4\Runtime\Atn\ATN;
use Antlr\Antlr4\Runtime\Atn\ATNDeserializer;
use Antlr\Antlr4\Runtime\Atn\ParserATNSimulator;
use Antlr\Antlr4\Runtime\Dfa\DFA;
use Antlr\Antlr4\Runtime\Error\Exceptions\FailedPredicateException;
use Antlr\Antlr4\Runtime\Error\Exceptions\NoViableAltException;
use Antlr\Antlr4\Runtime\PredictionContexts\PredictionContextCache;
use Antlr\Antlr4\Runtime\Error\Exceptions\RecognitionException;
use Antlr\Antlr4\Runtime\RuleContext;
use Antlr\Antlr4\Runtime\Token;
use Antlr\Antlr4\Runtime\TokenStream;
use Antlr\Antlr4\Runtime\Vocabulary;
use Antlr\Antlr4\Runtime\VocabularyImpl;
use Antlr\Antlr4\Runtime\RuntimeMetaData;
use Antlr\Antlr4\Runtime\Parser;
<if(namedActions.definitions)><namedActions.definitions><endif>

```

```

final class <parser.name> extends <superClass; null="Parser">
{
<if(parser.tokens)>
public const <parser.tokens:{k | <k> = <parser.tokens.(k)>}; separator=", ", wrap, anchor>;
<endif>

public const <parser.rules:{r | RULE_<r.name> = <r.index>}; separator=", ", wrap, anchor>;

/**
 * @var array<string>
 */
public const RULE_NAMES = [
<parser.ruleNames:{r | '<r>'}; separator=", ", wrap, anchor>
];

<vocabulary(parser.literalNames, parser.symbolicNames)>

<atn>
protected static $atn;
protected static $decisionToDFA;
protected static $sharedContextCache;
<if(namedActions.members)>

<namedActions.members>

```

<endif>

<parser:(ctor())>

private static function initialize() : void

```
{
  if (self::$atn !== null) {
    return;
  }
```

RuntimeMetaData::checkVersion('<file.ANTLRVersion>', RuntimeMetaData::VERSION);

\$atn = (new ATNDeserializer())->deserialize(self::SERIALIZED_ATN);

\$decisionToDFA = [];

```
for ($i = 0, $count = $atn->getNumberOfDecisions(); $i < $count; $i++) {
  $decisionToDFA[] = new DFA($atn->getDecisionState($i), $i);
}
```

self::\$atn = \$atn;

self::\$decisionToDFA = \$decisionToDFA;

self::\$sharedContextCache = new PredictionContextCache();

```
}
```

public function getGrammarFileName() : string

```
{
  return "<parser.grammarFileName>";
}
```

public function getRuleNames() : array

```
{
  return self::RULE_NAMES;
}
```

public function getSerializedATN() : string

```
{
  return self::SERIALIZED_ATN;
}
```

public function getATN() : ATN

```
{
  return self::$atn;
}
```

public function getVocabulary() : Vocabulary

```
{
  static $vocabulary;
```

```

    return $vocabulary = $vocabulary ?? new VocabularyImpl(self::LITERAL_NAMES, self::SYMBOLIC_NAMES);
    }
<endif(funcs)>

<funcs; separator="\n\n">
<endif>
<if(sempredFuncs)>

public function sempred(?RuleContext $localContext, int $ruleIndex, int $predicateIndex) : bool
{
    switch ($ruleIndex) {
    <parser.sempredFuncs.values:{f}
    case <f.ruleIndex>:
    return $this->sempred<f.name; format="cap">($localContext, $predicateIndex);}; separator="\n\n">

    default:
    return true;
    }
}

<sempredFuncs.values; separator="\n\n">
<endif>
}
}

namespace <if(file.genPackage)><file.genPackage>\\<endif>Context {
    use Antlr\\Antlr4\\Runtime\\ParserRuleContext;
    use Antlr\\Antlr4\\Runtime\\Token;
    use Antlr\\Antlr4\\Runtime\\Tree\\ParseTreeVisitor;
    use Antlr\\Antlr4\\Runtime\\Tree\\TerminalNode;
    use Antlr\\Antlr4\\Runtime\\Tree\\ParseTreeListener;
    use <if(file.genPackage)><file.genPackage>\\<endif><parser.name>;
    <if (file.genVisitor)>use <if(file.genPackage)><file.genPackage>\\<endif><file.grammarName>Visitor;<endif>
    <if (file.genListener)>use <if(file.genPackage)><file.genPackage>\\<endif><file.grammarName>Listener;<endif>
    <namedActions.contexts>

    <funcs :{ func | <func.ruleCtx><if(func.altLabelCtxs)>

    <func.altLabelCtxs:{1 | <func.altLabelCtxs.(1)>}; separator="\n\n"><endif> }; separator="\n\n">
    }
>>

vocabulary(literalNames, symbolicNames) ::= <<
/**
 * @var array<string|null>
 */
private const LITERAL_NAMES = [

```

```

    <literalNames:{t | <t>}; null="null", separator=", ", wrap, anchor>
];

/**
 * @var array<string>
 */
private const SYMBOLIC_NAMES = [
    <symbolicNames:{t | <t>}; null="null", separator=", ", wrap, anchor>
];
>>

dumpActions(recog, argFuncs, actionFuncs, sempredFuncs) ::= <<
<if(actionFuncs)>

public function action(?RuleContext $localContext, int $ruleIndex, int $actionIndex) : void
{
    switch ($ruleIndex) {
        <recog.actionFuncs.values:{f}
        case <f.ruleIndex>:
            $this->action<f.name; format="cap">($localContext, $actionIndex);
            break;}; separator="\n\n">
    }
}

<actionFuncs.values; separator="\n">
<endif>
<if(sempredFuncs)>

public function sempred(?RuleContext $localContext, int $ruleIndex, int $predicateIndex) : bool
{
    switch ($ruleIndex) {
        <recog.sempredFuncs.values:{f}
        case <f.ruleIndex>:
            return $this->sempred<f.name; format="cap">($localContext, $predicateIndex);}; separator="\n\n">
    }

    return true;
}
<sempredFuncs.values; separator="\n\n">
<endif>
>>

parser_ctor(p) ::= <<
public function __construct(TokenStream $input)
{
    parent::__construct($input);

    self::initialize();
}
}

```

```

    $this->interp = new ParserATNSimulator($this, self::$atn, self::$decisionToDFA, self::$sharedContextCache);
}
>>

/**
 * This generates a private method since the actionIndex is generated, making
 * an overriding implementation impossible to maintain.
 */
RuleActionFunction(r, actions) ::= <<
private function action<r.name; format="cap">(Context<r.ctxType> $localContext, int $actionIndex) : void
{
    switch ($actionIndex) {
    <actions:{index|
    case <index>:
    <actions.(index)>

    break;}; separator="\n\n">
    }
}
>>

/**
 * This generates a private method since the predicateIndex is generated, making
 * an overriding implementation impossible to maintain.
 */
RuleSempredFunction(r, actions) ::= <<
private function sempred<r.name; format="cap">(Context<r.ctxType> $localContext, int $predicateIndex) : bool
{
    switch ($predicateIndex) {
    <actions:{index|
    case <index>:
        return <actions.(index)>;}; separator="\n\n">
    }

    return true;
}
>>

RuleFunction(currentRule, args, code, locals, ruleCtx, altLabelCtxs, namedActions, finallyAction, exceptions, postamble)
::= <<
/**
 * @throws RecognitionException
 */
<if(currentRule.modifiers)><currentRule.modifiers:{f | <f> }><endif>public function <currentRule.name>(args;
separator=", ">) : Context<r.ctxType>
{
    $localContext = new Context<r.ctxType>($this->ctx, $this->getState())<currentRule.args:{a | ,

```

```
$<a.name>}>);
```

```
    $this->enterRule($localContext, <currentRule.startState>, self::RULE_<currentRule.name>);  
    <namedActions.init>  
    <locals; separator="\n">  
  
    try {  
        <code>  
        <postamble; separator="\n">  
        <namedActions.after>  
    }<if(exceptions)><exceptions; separator="\n"><else> catch (RecognitionException $exception) {  
        $localContext->exception = $exception;  
        $this->errorHandler->reportError($this, $exception);  
        $this->errorHandler->recover($this, $exception);  
    }<endif> finally {  
        <finallyAction>  
        $this->exitRule();  
    }  
  
    return $localContext;  
}  
>>
```

```
LeftRecursiveRuleFunction(currentRule,args,code,locals,ruleCtx,altLabelCtxs,namedActions,finallyAction,postamble) ::= <<
```

```
/**
```

```
 * @throws RecognitionException
```

```
 */
```

```
<if(currentRule.modifiers)><currentRule.modifiers:{f | <f> }><endif>public function <currentRule.name>(<args;  
separator=", ">) : Context\<currentRule.ctxType>
```

```
{  
    return $this->recursive<currentRule.name; format="cap">(0<currentRule.args:{a | , <a.name>}>);  
}
```

```
/**
```

```
 * @throws RecognitionException
```

```
 */
```

```
private function recursive<currentRule.name; format="cap">(int $precedence<args:{a | , <a>}>) :  
Context\<currentRule.ctxType>
```

```
{  
    $parentContext = $this->ctx;  
    $parentState = $this->getState();  
    $localContext = new Context\<currentRule.ctxType>($this->ctx, $parentState<currentRule.args:{a | ,  
<a.name>}>);  
    $previousContext = $localContext;  
    $startState = <currentRule.startState>;  
    $this->enterRecursionRule($localContext, <currentRule.startState>, self::RULE_<currentRule.name>,  
$precedence);
```

```

<namedActions.init>
<locals; separator="\n">

try {
  <code>
  <postamble; separator="\n">
  <namedActions.after>
} catch (RecognitionException $exception) {
  $localContext->exception = $exception;
  $this->errorHandler->reportError($this, $exception);
  $this->errorHandler->recover($this, $exception);
} finally {
  <finallyAction>
  $this->unrollRecursionContexts($parentContext);
}

return $localContext;
}
>>

```

```

CodeBlockForOuterMostAlt(currentOuterMostAltCodeBlock, locals, preamble, ops) ::= <<
<if(currentOuterMostAltCodeBlock.altLabel)>$localContext = new
Context\|<currentOuterMostAltCodeBlock.altLabel; format="cap">Context($localContext);<endif>
$this->enterOuterAlt($localContext, <currentOuterMostAltCodeBlock.alt.altNum>);
<CodeBlockForAlt(currentAltCodeBlock=currentOuterMostAltCodeBlock, ...)>
>>

```

```

CodeBlockForAlt(currentAltCodeBlock, locals, preamble, ops) ::= <<
<locals; separator="\n">
<preamble; separator="\n">
<ops; separator="\n">
>>

```

```

LL1AltBlock(choice, preamble, alts, error) ::= <<
$this->setState(<choice.stateNumber>);
$this->errorHandler->sync($this);
<if(choice.label)><labelref(choice.label)> = $this->input->LT(1);<endif>
<preamble; separator="\n">

switch ($this->input->LA(1)) {
  <choice.altLook,alts:{look,alt| <cases(ttypes=look)>
  <alt>
  break;}; separator="\n\n">

default:
  <error>
}
>>

```

```

LL1OptionalBlock(choice, alts, error) ::= <<
$this->setState(<choice.stateNumber>);
$this->errorHandler->sync($this);

switch ($this->input->LA(1)) {
  <choice.altLook,alts:{look,alt| <cases(ttypes=look)>
  <alt>
  break;}; separator="\n\n">

default:
  break;
}
>>

LL1OptionalBlockSingleAlt(choice, expr, alts, preamble, error, followExpr) ::= <<
$this->setState(<choice.stateNumber>);
$this->errorHandler->sync($this);
<preamble; separator="\n">

if (<expr>) {
  <alts; separator="\n">
}
>>

LL1StarBlockSingleAlt(choice, loopExpr, alts, preamble, iteration) ::= <<
$this->setState(<choice.stateNumber>);
$this->errorHandler->sync($this);

<preamble; separator="\n">
while (<loopExpr>) {
  <alts; separator="\n">
  $this->setState(<choice.loopBackStateNumber>);
  $this->errorHandler->sync($this);
  <iteration>
}
>>

LL1PlusBlockSingleAlt(choice, loopExpr, alts, preamble, iteration) ::= <<
$this->setState(<choice.blockStartStateNumber>); <! alt block decision !>
$this->errorHandler->sync($this);

<preamble; separator="\n">
do {
  <alts; separator="\n">
  $this->setState(<choice.stateNumber>); <! loopback/exit decision !>
  $this->errorHandler->sync($this);
  <iteration>
}

```

```

} while (<loopExpr>);
>>

// LL(*) stuff

AltBlock(choice, preamble, alts, error) ::= <<
$this->setState(<choice.stateNumber>);
$this->errorHandler->sync($this);
<if(choice.label)><labelref(choice.label)> = $this->input->LT(1);<endif>
<preamble; separator="\n">

switch ($this->getInterpreter()->adaptivePredict($this->input, <choice.decision>, $this->ctx)) {
  <alts:{ alt |
case <i>:
  <alt>
break; }; separator="\n\n">
}
>>

OptionalBlock(choice, alts, error) ::= <<
$this->setState(<choice.stateNumber>);
$this->errorHandler->sync($this);

switch ($this->getInterpreter()->adaptivePredict($this->input, <choice.decision>, $this->ctx)) {
<alts:{ alt |
  case <i><if(!choice.ast.greedy)>+1<endif>:
  <alt>
break; }; separator="\n\n">
}
>>

StarBlock(choice, alts, sync, iteration) ::= <<
$this->setState(<choice.stateNumber>);
$this->errorHandler->sync($this);

$alt = $this->getInterpreter()->adaptivePredict($this->input, <choice.decision>, $this->ctx);

while ($alt !== <choice.exitAlt> && $alt !== ATN::INVALID_ALT_NUMBER) {
if ($alt === 1<if(!choice.ast.greedy)>+1<endif>) {
  <iteration>
  <alts> <! should only be one !>
}

$this->setState(<choice.loopBackStateNumber>);
$this->errorHandler->sync($this);

$alt = $this->getInterpreter()->adaptivePredict($this->input, <choice.decision>, $this->ctx);
}

```

>>

```
PlusBlock(choice, alts, error) ::= <<
$this->setState(<choice.blockStartStateNumber>); <! alt block decision !>
$this->errorHandler->sync($this);
```

```
$alt = 1<if(!choice.ast.greedy)>+1<endif>;
```

```
do {
  switch ($alt) {
    <alts:{alt|
  case <i><if(!choice.ast.greedy)>+1<endif>:
    <alt>
    break;}; separator="\n\n">
  default:
    <error>
  }
}
```

```
$this->setState(<choice.loopBackStateNumber>); <! loopback/exit decision !>
$this->errorHandler->sync($this);
```

```
$alt = $this->getInterpreter()->adaptivePredict($this->input, <choice.decision>, $this->ctx);
} while ($alt !== <choice.exitAlt> && $alt !== ATN::INVALID_ALT_NUMBER);
```

>>

```
Sync(s) ::= "sync(<s.expecting.name>);"
```

```
ThrowNoViableAlt(t) ::= "throw new NoViableAltException($this);"
```

```
TestSetInline(s) ::= <<
<s.bitsets:{bits | <if(rest(rest(bits.ttypes)))><bitsetBitfieldComparison(s, bits)><else><bitsetInlineComparison(s,
bits)><endif>}; separator=" || ">
```

>>

```
// Java language spec 15.19 - shift operators mask operands rather than overflow to 0... need range test
```

```
testShiftInRange(shiftAmount) ::= <<
```

```
((<shiftAmount>) & ~0x3f) === 0
```

>>

```
// produces smaller bytecode only when bits.ttypes contains more than two items
```

```
bitsetBitfieldComparison(s, bits) ::= <%
```

```
(<testShiftInRange({<offsetShiftVar(s.varName, bits.shift)>})> && ((1 \<< <offsetShiftVar(s.varName,
bits.shift)>) & (<bits.ttypes:{ttype | (1 \<< <offsetShiftConst(ttype, bits.shift)>}); separator=" | ">)) !== 0)
%>
```

```
isZero ::= [
```

```
"0":true,
```

```
default:false
```

]

```
offsetShiftVar(shiftAmount, offset) ::= <%
<if(!isZero.(offset))>(<$<shiftAmount> - <offset></else><$<shiftAmount></endif>
%>
offsetShiftConst(shiftAmount, offset) ::= <%
<if(!isZero.(offset))>(self::<shiftAmount> - <offset></else>self::<shiftAmount></endif>
%>

// produces more efficient bytecode when bits.ttypes contains at most two items
bitsetInlineComparison(s, bits) ::= <%
<bits.ttypes: { ttype | $<s.varName> === self::<ttype> }; separator=" || ">
%>

cases(ttypes) ::= <<
<ttypes: { t | case self::<t> }; separator="\n">
>>

InvokeRule(r, argExprsChunks) ::= <<
$this->setState(<r.stateNumber>);
<if(r.labels)><r.labels: {1 | <labelref(1)> = }></endif>$this-><if(r.ast.options.p)>recursive<r.name>;
format="cap"></else><r.name></endif>(<if(r.ast.options.p)><r.ast.options.p></if><argExprsChunks>,</endif></endif>
<argExprsChunks>);
>>

MatchToken(m) ::= <<
$this->setState(<m.stateNumber>);
<if(m.labels)><m.labels: {1 | <labelref(1)> = }></endif>$this->match(self::<m.name>);
>>

MatchSet(m, expr, capture) ::= "<CommonSetStuff(m, expr, capture, false)>"

MatchNotSet(m, expr, capture) ::= "<CommonSetStuff(m, expr, capture, true)>"

CommonSetStuff(m, expr, capture, invert) ::= <<
$this->setState(<m.stateNumber>);

<if(m.labels)><m.labels: {1 | <labelref(1)> = }>$this->input->LT(1);</endif>
<capture>

if (<if(invert)><$m.varName> \<= 0 || </else>!</endif><(<expr>)) {
    <if(m.labels)><m.labels: {1 | <labelref(1)> = }></endif>$this->errorHandler->recoverInline($this);
} else {
    if ($this->input->LA(1) === Token::EOF) {
        $this->matchedEOF = true;
    }

    $this->errorHandler->reportMatch($this);
```

```

    $this->consume();
}
>>

Wildcard(w) ::= <<
$this->setState(<w.stateNumber>);
<if(w.labels)><w.labels:{1 | <labelref(l)> = }><endif>$this->matchWildcard();
>>

// ACTION STUFF

Action(a, foo, chunks) ::= "<chunks>"

ArgAction(a, chunks) ::= "<chunks>"

SemPred(p, chunks, failChunks) ::= <<
$this->setState(<p.stateNumber>);

if (!(<chunks>)) {
    throw new FailedPredicateException($this, <p.predicate><if(failChunks)>, <failChunks><elseif(p.msg)>,
    <p.msg><endif>);
}
>>

ExceptionClause(e, catchArg, catchAction) ::= <<
catch (<catchArg>) {
    <catchAction>
}
>>

// lexer actions are not associated with model objects

LexerSkipCommand() ::= "$this->skip();"
LexerMoreCommand() ::= "$this->more();"
LexerPopModeCommand() ::= "$this->popMode();"

LexerTypeCommand(arg, grammar) ::= "$this->type = <arg>";
LexerChannelCommand(arg, grammar) ::= "$this->channel = <arg>";
LexerModeCommand(arg, grammar) ::= "$this->mode = <arg>";
LexerPushModeCommand(arg, grammar) ::= "$this->pushMode(<arg>);"

ActionText(t) ::= "<t.text>"
ActionTemplate(t) ::= "<t.st>"
ArgRef(a) ::= "$localContext-><a.name>"
LocalRef(a) ::= "$localContext-><a.name>"
RetValRef(a) ::= "$localContext-><a.name>"
QRetValRef(a) ::= "<ctx(a)>-><a.dict>-><a.name>"
/** How to translate $tokenLabel */

```

```

TokenRef(t) ::= "<ctx(t)>-><t.name>"
LabelRef(t) ::= "<ctx(t)>-><t.name>"
ListLabelRef(t) ::= "<ctx(t)>-><ListLabelName(t.name)>"
SetAttr(s,rhsChunks) ::= "<ctx(s)>-><s.name> = <rhsChunks>;"

TokenLabelType() ::= "<file.TokenLabelType; null={Token}>"
InputSymbolType() ::= "<file.InputSymbolType; null={Token}>"

TokenPropertyRef_text(t) ::= "<ctx(t)>-><t.label> !== null ? <ctx(t)>-><t.label>->getText() : null)"
TokenPropertyRef_type(t) ::= "<ctx(t)>-><t.label> !== null ? <ctx(t)>-><t.label>->getType() : 0)"
TokenPropertyRef_line(t) ::= "<ctx(t)>-><t.label> !== null ? <ctx(t)>-><t.label>->getLine() : 0)"
TokenPropertyRef_pos(t) ::= "<ctx(t)>-><t.label> !== null ? <ctx(t)>-><t.label>->getCharPositionInLine() : 0)"
TokenPropertyRef_channel(t) ::= "<ctx(t)>-><t.label> !== null ? <ctx(t)>-><t.label>->getChannel() : 0)"
TokenPropertyRef_index(t) ::= "<ctx(t)>-><t.label> !== null ? <ctx(t)>-><t.label>->getTokenIndex() : 0)"
TokenPropertyRef_int(t) ::= "<ctx(t)>-><t.label> !== null ? (int) <ctx(t)>-><t.label>->getText() : 0)"

RulePropertyRef_start(r) ::= "<ctx(r)>-><r.label> !== null ? (<ctx(r)>-><r.label>->start) : null)"
RulePropertyRef_stop(r) ::= "<ctx(r)>-><r.label> !== null ? (<ctx(r)>-><r.label>->stop) : null)"
RulePropertyRef_text(r) ::= "<ctx(r)>-><r.label> !== null ? $this->input->getTextByTokens(<ctx(r)>-><r.label>->start, <ctx(r)>-><r.label>->stop) : null)"
RulePropertyRef_ctx(r) ::= "<ctx(r)>-><r.label>"
RulePropertyRef_parser(r) ::= "\$this"

ThisRulePropertyRef_start(r) ::= "$localContext->start"
ThisRulePropertyRef_stop(r) ::= "$localContext->stop"
ThisRulePropertyRef_text(r) ::= "$this->input->getTextByTokens($localContext->start, $this->input->LT(-1))"
ThisRulePropertyRef_ctx(r) ::= "$localContext"
ThisRulePropertyRef_parser(r) ::= "$this"

NonLocalAttrRef(s) ::= "\$this->getInvokingContext(<s.ruleIndex>)-><s.name>"
SetNonLocalAttr(s, rhsChunks) ::= "\$this->getInvokingContext(<s.ruleIndex>)-><s.name> = <rhsChunks>;"

AddToLabelList(a) ::= "<ctx(a.label)>-><a.listName>[] = <labelref(a.label)>;"

TokenDecl(t) ::= "<TokenLabelType()> $<t.name>"
TokenTypeDecl(t) ::= ""
TokenListDecl(t) ::= "array $<t.name> = []"
RuleContextDecl(r) ::= "<r.ctxName> $<r.name>"
RuleContextListDecl(rdecl) ::= "array $<rdecl.name> = []"
AttributeDecl(d) ::= "<d.type> $<d.name><if(d.initValue)> = <d.initValue><endif>"

PropertiesDecl(struct) ::= <<
<if(struct.tokenListDecls)>
<struct.tokenListDecls : { d | /**
* @var array<Token>|null $<d.name>
*/
public $<d.name>;}; separator="\n\n">
<endif>

```

```

<if(struct.tokenDecls)>
<if(struct.tokenListDecls)>

<endif>
<struct.tokenDecls : {d | /**
* @var <TokenLabelType()>|null $<d.name>
*/
public $<d.name>;}; separator="\n\n">
<endif>
<if(struct.ruleContextDecls)>
<if(struct.tokenListDecls || struct.tokenDecls)>

<endif>
<struct.ruleContextDecls : {d | /**
* @var <d.ctxName>|null $<d.name>
*/
public $<d.name>;}; separator="\n\n">
<endif>
<if(struct.ruleContextListDecls)>
<if(struct.tokenListDecls || struct.tokenDecls || struct.ruleContextDecls)>

<endif>
<struct.ruleContextListDecls : {d | /**
* @var array\<<d.ctxName>\>|null $<d.name>
*/
public $<d.name>;}; separator="\n\n">
<endif>
<if(struct.attributeDecls)>
<if(struct.tokenListDecls || struct.tokenDecls || struct.ruleContextDecls || struct.ruleContextListDecls)>

<endif>
<struct.attributeDecls : {d | /**
* @var <d.type><if(!d.initValue)>|null<endif> $<d.name>
*/
public $<d.name><if(d.initValue)> = <d.initValue><endif>;}; separator="\n\n">
<endif>

>>

ContextTokenGetterDecl(t) ::= <<
public function <t.name>() : ?TerminalNode
{
    return $this->getToken(<parser.name>::<t.name>, 0);
}
>>

ContextTokenListGetterDecl(t) ::= <<
>>

```

```

ContextTokenListIndexedGetterDecl(t) ::= <<
/**
 * @return array<TerminalNode>|TerminalNode|null
 */
public function <t.name>(int $index = null)
{
    if ($index === null) {
        return $this->getTokens(<parser.name>::<t.name>);
    }

    return $this->getToken(<parser.name>::<t.name>, $index);
}
>>

ContextRuleGetterDecl(r) ::= <<
public function <r.name>() : ?<r.ctxName>
{
    return $this->getTypedRuleContext(<r.ctxName>::class, 0);
}
>>

ContextRuleListGetterDecl(r) ::= <<
>>

ContextRuleListIndexedGetterDecl(r) ::= <<
/**
 * @return array<<r.ctxName>>|<r.ctxName>|null
 */
public function <r.name>(int $index = null)
{
    if ($index === null) {
        return $this->getTypedRuleContexts(<r.ctxName>::class);
    }

    return $this->getTypedRuleContext(<r.ctxName>::class, $index);
}
>>

LexerRuleContext() ::= "RuleContext"

/**
 * The rule context name is the rule followed by a suffix; e.g., r becomes rContext.
 */
RuleContextNameSuffix() ::= "Context"

ImplicitTokenLabel(tokenName) ::= "<tokenName>"
ImplicitRuleLabel(ruleName) ::= "<ruleName>"

```

```

ImplicitsetLabel(id) ::= "_tset<id>"
ListLabelName(label) ::= "<label>"

CaptureNextToken(d) ::= "$<d.varName> = \${this->input->LT(1)};"
CaptureNextTokenType(d) ::= "$<d.varName> = \${this->input->LA(1)};"

StructDecl(struct,ctorAttrs,attrs, getters,dispatchMethods,interfaces,extensionMembers) ::= <<
class <struct.name> extends
<if(contextSuperClass)><contextSuperClass><else>ParserRuleContext<endif><if(interfaces)> implements
<interfaces; separator=", "><endif>
{
<PropertiesDecl(struct)>
public function __construct(?ParserRuleContext $parent, ?int $invokingState = null<ctorAttrs:{a | , ?<a> = null}>)
{
parent::__construct($parent, $invokingState);
<if(struct.ctorAttrs)>

<struct.ctorAttrs:{a | \${this-><a.name> = \${<a.name> ?? \${this-><a.name>}}; separator="\n">
<endif>
}

public function getRuleIndex() : int
{
return <parser.name>::RULE_<struct.derivedFromName>;
}
<if(getters)>

<getters:{g | <g>}; separator="\n\n">
<endif>
<if(struct.provideCopyFrom)> <! don't need copy unless we have subclasses !>
public function copyFrom(ParserRuleContext $context) : void
{
parent::copyFrom($context);

<struct.attrs:{a | \${this-><a.name> = \${<a.name> ?? \${<a.name>}}; separator="\n">
}
<endif>
<if(dispatchMethods)>

<dispatchMethods; separator="\n\n">
<endif>
<if(extensionMembers)>

<extensionMembers; separator="\n\n">
<endif>
}
>>

```

```

AltLabelStructDecl(struct,attrs, getters,dispatchMethods) ::= <<
class <struct.name> extends <struct.parentRule; format="cap">Context
{
<PropertiesDecl(struct)>
public function __construct(<struct.parentRule; format="cap">Context $context)
{
    parent::__construct($context);

    $this->copyFrom($context);
}
<if(getters)>

    <getters:{g | <g>}; separator="\n\n">
<endif>
<if(dispatchMethods)>

    <dispatchMethods; separator="\n\n">
<endif>
}
>>

ListenerDispatchMethod(method) ::= <<
public function <if(method.isEnter)>enter<else>exit<endif>Rule(ParseTreeListener $listener) : void
{
    if ($listener instanceof <parser.grammarName>Listener) {
        $listener-><if(method.isEnter)>enter<else>exit<endif><struct.derivedFromName; format="cap">($this);
    }
}
>>

VisitorDispatchMethod(method) ::= <<
public function accept(ParseTreeVisitor $visitor)
{
    if ($visitor instanceof <parser.grammarName>Visitor) {
        return $visitor->visit<struct.derivedFromName; format="cap">($this);
    }

    return $visitor->visitChildren($this);
}
>>

/** If we don't know location of label def x, use this template */
labelref(x) ::= "<if(!x.isLocal)>$localContext-><endif><x.name>"

/** For any action chunk, what is correctly-typed context struct ptr? */
ctx(actionChunk) ::= "$localContext"

// used for left-recursive rules

```

```

recRuleAltPredicate(ruleName,opPrec) ::= "\$this->precpred(\$this->ctx, <opPrec>)"

recRuleSetReturnAction(src,name) ::= "\$<name> = \$<src-><name>;"

recRuleSetStopToken()          ::= "\$this->ctx->stop = \$this->input->LT(-1);"

recRuleAltStartAction(ruleName, ctxName, label, isListLabel) ::= <<
$localContext = new Context\\<ctxName>Context($parentContext, $parentState);
<if(label)>
<if(isListLabel)>
$localContext-><label>[] = $previousContext;
<else>
$localContext-><label> = $previousContext;
<endif>
<endif>

$this->pushNewRecursionContext($localContext, $startState, self::RULE_<ruleName>);
>>

recRuleLabeledAltStartAction(ruleName, currentAltLabel, label, isListLabel) ::= <<
$localContext = new Context\\<currentAltLabel; format="cap">Context(new Context\\<ruleName;
format="cap">Context($parentContext, $parentState));
<if(label)>
<if(isListLabel)>
$localContext-><label>[] = $previousContext;
<else>
$localContext-><label> = $previousContext;
<endif>
<endif>

$this->pushNewRecursionContext($localContext, $startState, self::RULE_<ruleName>);
>>

recRuleReplaceContext(ctxName) ::= <<
$localContext = new Context\\<ctxName>Context($localContext);
$this->ctx = $localContext;
$previousContext = $localContext;
>>

recRuleSetPrevCtx() ::= <<
if ($this->getParseListeners() !== null) {
    $this->triggerExitRuleEvent();
}

$previousContext = $localContext;
>>

```

```

LexerFile(lexerFile, lexer, namedActions) ::= <<
<fileHeader(lexerFile.grammarFileName, lexerFile.ANTLRVersion)>
<lexer>
>>

```

```

Lexer(lexer, atn, actionFuncs, sempredFuncs, superClass) ::= <<
namespace<if(lexerFile.genPackage)> <lexerFile.genPackage><endif> {
<if(namedActions.header)><namedActions.header><endif>
use Antlr\\Antlr4\\Runtime\\Atn\\ATNDeserializer;
use Antlr\\Antlr4\\Runtime\\Atn\\LexerATNSimulator;
use Antlr\\Antlr4\\Runtime\\Lexer;
use Antlr\\Antlr4\\Runtime\\CharStream;
use Antlr\\Antlr4\\Runtime\\PredictionContexts\\PredictionContextCache;
use Antlr\\Antlr4\\Runtime\\RuleContext;
use Antlr\\Antlr4\\Runtime\\Atn\\ATN;
use Antlr\\Antlr4\\Runtime\\Dfa\\DFA;
use Antlr\\Antlr4\\Runtime\\Vocabulary;
use Antlr\\Antlr4\\Runtime\\RuntimeMetaData;
use Antlr\\Antlr4\\Runtime\\VocabularyImpl;
<if(namedActions.definitions)><namedActions.definitions><endif>

```

```

final class <lexer.name> extends <superClass; null="Lexer">

```

```

{
<if(lexer.tokens)>
public const <lexer.tokens:{k | <k> = <lexer.tokens.(k)>}; separator=", ", wrap, anchor>;
<endif>

```

```

<if(lexer.channels)>
public const <lexer.channels:{c | <c> = <lexer.channels.(c)>}; separator=", ", wrap, anchor>;
<endif>

```

```

<if(rest(lexer.modes))>
public const <rest(lexer.modes):{m | <m>=<i>}; separator=", ", wrap, anchor>;
<endif>

```

```

/**

```

```

* @var array<string>

```

```

*/

```

```

public const CHANNEL_NAMES = [
'DEFAULT_TOKEN_CHANNEL', 'HIDDEN'<if (lexer.channels)>, <lexer.channels:{c | '<c>'}; separator=", ",
wrap, anchor><endif>
];

```

```

/**

```

```

* @var array<string>

```

```

*/

```

```

public const MODE_NAMES = [
<lexer.modes:{m | '<m>'}; separator=", ", wrap, anchor>

```

```

];

/**
 * @var array<string>
 */
public const RULE_NAMES = [
<lexer.ruleNames:{r | '<r>'}; separator=", ", wrap, anchor>
];

<vocabulary(lexer.literalNames, lexer.symbolicNames)>

<atn>
protected static $atn;
protected static $decisionToDFA;
protected static $sharedContextCache;
<if(namedActions.members)>

<namedActions.members>
<endif>

public function __construct(CharStream $input)
{
    parent::__construct($input);

    self::initialize();

    $this->interp = new LexerATNSimulator($this, self::$atn, self::$decisionToDFA, self::$sharedContextCache);
}

private static function initialize() : void
{
    if (self::$atn !== null) {
        return;
    }

    RuntimeMetaData::checkVersion('<lexerFile.ANTLRVersion>', RuntimeMetaData::VERSION);

    $atn = (new ATNDeserializer()->deserialize(self::SERIALIZED_ATN);

    $decisionToDFA = [];
    for ($i = 0, $count = $atn->getNumberOfDecisions(); $i < $count; $i++) {
        $decisionToDFA[] = new DFA($atn->getDecisionState($i), $i);
    }

    self::$atn = $atn;
    self::$decisionToDFA = $decisionToDFA;
    self::$sharedContextCache = new PredictionContextCache();
}

```

```

public static function vocabulary() : Vocabulary
{
    static $vocabulary;

    return $vocabulary = $vocabulary ?? new VocabularyImpl(self::LITERAL_NAMES, self::SYMBOLIC_NAMES);
}

public function getGrammarFileName() : string
{
    return '<lexer.grammarFileName>';
}

public function getRuleNames() : array
{
    return self::RULE_NAMES;
}

public function getSerializedATN() : string
{
    return self::SERIALIZED_ATN;
}

/**
 * @return array<string>
 */
public function getChannelNames() : array
{
    return self::CHANNEL_NAMES;
}

/**
 * @return array<string>
 */
public function getModeNames() : array
{
    return self::MODE_NAMES;
}

public function getATN() : ATN
{
    return self::$atn;
}

public function getVocabulary() : Vocabulary
{
    return self::vocabulary();
}

```

```

    <dumpActions(lexer, "", actionFuncs, sempredFuncs)>
    }
    }
    >>

SerializedATN(model) ::= <<
<if(rest(model.segments))>
/**
 * @var string
 */
private const SERIALIZED_ATN =
    <model.segments:{segment| "<segment; wrap={ " .<n>"}>"; separator=" .\n">;
<else>
/**
 * @var string
 */
private const SERIALIZED_ATN =
    "<model.serialized; wrap={ " .<n>  "}>";
<endif>
>>

/**
 * Using a type to init value map, try to init a type; if not in table
 * must be an object, default value is `null`.
 */
initValue(typeName) ::= <<
<phpTypeInitMap.(typeName)>
>>

codeFileExtension() ::= ".php"

```

Found in path(s):

```

* /opt/cola/permits/1473460068_1668481512.5494933/0/antlr4-4-8-1-
jar/org/antlr/v4/tool/templates/codegen/PHP/PHP.stg

```

No license file was found, but licenses were detected in source scan.

```

/*
 * [The "BSD license"]
 * Copyright (c) 2015 Dan McLaughlin, Mike Lischke
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright

```

* notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * 3. The name of the author may not be used to endorse or promote products
 * derived from this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
 * IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
 * IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
 * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
 * DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
 * THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
 * THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */

```
import "Files.stg" // All file specific stuff.
```

```
cppTypeInitMap ::= [
  "int": "0",
  "long": "0",
  "float": "0.0f",
  "double": "0.0",
  "bool": "false",
  "short": "0",
  "char": "0",
  default: "nullptr" // anything other than a primitive type is an object
]
```

```
LexerHeader(lexer, atn, actionFuncs, sempredFuncs, superClass = {antlr4::Lexer}) ::= <<
<namedActions.context>
```

```
class <file.exportMacro> <lexer.name> : public <superClass> {
public:
<if (lexer.tokens)>
enum {
  <lexer.tokens: {k | <k> = <lexer.tokens.(k)>}; separator=" ", wrap, anchor>
};
<endif>

<if (lexer.channels)>
enum {
  <lexer.channels: {k | <k> = <lexer.channels.(k)>}; separator=" ", wrap, anchor>
};
<endif>

<if (rest(lexer.modes))>
```

```

enum {
    <rest(lexer.modes): {m | <m> = <i>}; separator=", ", wrap, anchor>
};
<endif>

<lexer.name>(antlr4::CharStream *input);
~<lexer.name>();

<namedActions.members>
virtual std::string getGrammarFileName() const override;
virtual const std::vector<std::string>& getRuleNames() const override;

virtual const std::vector<std::string>& getChannelNames() const override;
virtual const std::vector<std::string>& getModeNames() const override;
virtual const std::vector<std::string>& getTokenNames() const override; // deprecated, use vocabulary instead
virtual antlr4::dfa::Vocabulary& getVocabulary() const override;

virtual const std::vector<uint16_t> getSerializedATN() const override;
virtual const antlr4::atn::ATN& getATN() const override;

<if (actionFuncs)>
virtual void action(antlr4::RuleContext *context, size_t ruleIndex, size_t actionIndex) override;
<endif>
<if (sempredFuncs)>
virtual bool sempred(antlr4::RuleContext *_localctx, size_t ruleIndex, size_t predicateIndex) override;
<endif>

private:
static std::vector<antlr4::dfa::DFA> _decisionToDFA;
static antlr4::atn::PredictionContextCache _sharedContextCache;
static std::vector<std::string> _ruleNames;
static std::vector<std::string> _tokenNames;
static std::vector<std::string> _channelNames;
static std::vector<std::string> _modeNames;

static std::vector<std::string> _literalNames;
static std::vector<std::string> _symbolicNames;
static antlr4::dfa::Vocabulary _vocabulary;
<atn>

<namedActions.declarations>

// Individual action functions triggered by action() above.
<actionFuncs.values; separator="\n">

// Individual semantic predicate functions triggered by sempred() above.
<sempredFuncs.values; separator="\n">

```

```

struct Initializer {
    Initializer();
};
static Initializer _init;
};
>>

Lexer(lexer, atn, actionFuncs, sempredFuncs, superClass = {Lexer}) ::= <<
<lexer.name>::<lexer.name>(CharStream *input) : <superClass>(input) {
    _interpreter = new atn::LexerATNSimulator(this, _atn, _decisionToDFA, _sharedContextCache);
}

<lexer.name>::~~<lexer.name>() {
    delete _interpreter;
}

std::string <lexer.name>::getGrammarFileName() const {
    return "<lexer.grammarFileName>";
}

const std::vector<std::string>& <lexer.name>::getRuleNames() const {
    return _ruleNames;
}

const std::vector<std::string>& <lexer.name>::getChannelNames() const {
    return _channelNames;
}

const std::vector<std::string>& <lexer.name>::getModeNames() const {
    return _modeNames;
}

const std::vector<std::string>& <lexer.name>::getTokenNames() const {
    return _tokenNames;
}

dfa::Vocabulary& <lexer.name>::getVocabulary() const {
    return _vocabulary;
}

const std::vector<uint16_t> <lexer.name>::getSerializedATN() const {
    return _serializedATN;
}

const atn::ATN& <lexer.name>::getATN() const {
    return _atn;
}

```

```

<namedActions.definitions>

<if (actionFuncs)>
void <lexer.name>::action(RuleContext *context, size_t ruleIndex, size_t actionIndex) {
    switch (ruleIndex) {
        <lexer.actionFuncs.values: {f | case <f.ruleIndex>: <f.name>Action(dynamic_cast<<f.ctxType> *)>(context),
actionIndex); break;}; separator="\n">

        default:
            break;
    }
}
<endif>

<if (sempredFuncs)>
bool <lexer.name>::sempred(RuleContext *context, size_t ruleIndex, size_t predicateIndex) {
    switch (ruleIndex) {
        <lexer.sempredFuncs.values: {f | case <f.ruleIndex>: return <f.name>Sempred(dynamic_cast<<f.ctxType>
*>(context), predicateIndex);}; separator="\n">

        default:
            break;
    }
    return true;
}
<endif>

<actionFuncs.values; separator="\n">

<sempredFuncs.values; separator="\n">

// Static vars and initialization.
std::vector<dfa::DFA> <lexer.name>::_decisionToDFA;
atn::PredictionContextCache <lexer.name>::_sharedContextCache;

// We own the ATN which in turn owns the ATN states.
atn::ATN <lexer.name>::_atn;
std::vector<uint16_t> <lexer.name>::_serializedATN;

std::vector<std::string> <lexer.name>::_ruleNames = {
    <lexer.ruleNames: {r | u8"<r>"}; separator = ", ", wrap, anchor>
};

std::vector<std::string> <lexer.name>::_channelNames = {
    "DEFAULT_TOKEN_CHANNEL", "HIDDEN"<if (lexer.channels)>, <lexer.channels: {c | u8"<c>"}; separator =
", ", wrap, anchor><endif>
};

```

```

std::vector<std::string> <lexer.name>::_modeNames = {
    <lexer.modes: {m | u8"<m>"}; separator = ", ", wrap, anchor>
};

std::vector<std::string> <lexer.name>::_literalNames = {
    <lexer.literalNames: {t | u8<t>}; null = "\\\"", separator = ", ", wrap, anchor>
};

std::vector<std::string> <lexer.name>::_symbolicNames = {
    <lexer.symbolicNames: {t | u8<t>}; null = "\\\"", separator = ", ", wrap, anchor>
};

dfa::Vocabulary <lexer.name>::_vocabulary(_literalNames, _symbolicNames);

std::vector<std::string> <lexer.name>::_tokenNames;

<lexer.name>::Initializer::Initializer() {
    // This code could be in a static initializer lambda, but VS doesn't allow access to private class members from there.
    for (size_t i = 0; i < _symbolicNames.size(); ++i) {
        std::string name = _vocabulary.getLiteralName(i);
        if (name.empty()) {
            name = _vocabulary.getSymbolicName(i);
        }

        if (name.empty()) {
            _tokenNames.push_back("\\<INVALID>");
        } else {
            _tokenNames.push_back(name);
        }
    }
}

<atn>
}

<lexer.name>::Initializer <lexer.name>::_init;
>>

RuleActionFunctionHeader(r, actions) ::= <<
void <r.name>Action(<r.ctxType> *context, size_t actionIndex);
>>

RuleActionFunction(r, actions) ::= <<
void <r.factory.grammar.name>::<r.name>Action(<r.ctxType> *context, size_t actionIndex) {
    switch (actionIndex) {
        <actions: {index | case <index>: <actions.(index)> break;}; separator="\n">

    default:
        break;
}
}

```

```
}  
}  
  
>>
```

```
RuleSempredFunctionHeader(r, actions) ::= <<  
bool <r.name>Sempred(<r.ctxType> *_localctx, size_t predicateIndex);  
>>
```

```
RuleSempredFunction(r, actions) ::= <<  
<! Called for both lexer and parser. But only one of them is actually available. Testing for the parser directly  
generates a warning, however. So do the check via the factory instead. !>  
bool <if (r.factory.g.lexer)><lexer.name><else><parser.name><endif>::<r.name>Sempred(<r.ctxType> *_localctx,  
size_t predicateIndex) {  
switch (predicateIndex) {  
    <actions: {index | case <index>: return <actions.(index)>}; separator=";\n">;  
  
default:  
    break;  
}  
return true;  
}  
  
>>
```

```
//-----
```

```
ParserHeader(parser, funcs, atn, sempredFuncs, superClass = {antlr4::Parser}) ::= <<  
<namedActions.context>
```

```
class <file.exportMacro> <parser.name> : public <superClass> {  
public:  
<if (parser.tokens)>  
enum {  
    <parser.tokens: {k | <k> = <parser.tokens.(k)>}; separator=", ", wrap, anchor>  
};  
<endif>  
  
<if (parser.rules)>  
enum {  
    <parser.rules: {r | Rule<r.name; format="cap"> = <r.index>}; separator=", ", wrap, anchor>  
};  
<endif>  
  
<parser.name>(antlr4::TokenStream *input);  
~<parser.name>();  
  
virtual std::string getGrammarFileName() const override;
```

```

virtual const antlr4::atn::ATN& getATN() const override { return _atn; };
virtual const std::vector<std::string>& getTokenNames() const override { return _tokenNames; }; // deprecated:
use vocabulary instead.
virtual const std::vector<std::string>& getRuleNames() const override;
virtual antlr4::dfa::Vocabulary& getVocabulary() const override;

<namedActions.members>

<parser.funcs: {f | class <f.name; format = "cap">Context;}; separator = "\n"> <! Forward declare context classes.
!>

<funcs; separator = "\n">

<if (sempredFuncs)>
virtual bool sempred(antlr4::RuleContext *_localctx, size_t ruleIndex, size_t predicateIndex) override;
<sempredFuncs.values; separator = "\n">
<endif>

private:
static std::vector<antlr4::dfa::DFA> _decisionToDFA;
static antlr4::atn::PredictionContextCache _sharedContextCache;
static std::vector<std::string> _ruleNames;
static std::vector<std::string> _tokenNames;

static std::vector<std::string> _literalNames;
static std::vector<std::string> _symbolicNames;
static antlr4::dfa::Vocabulary _vocabulary;
<atn>

<namedActions.declarations>

struct Initializer {
    Initializer();
};
static Initializer _init;
};
>>

Parser(parser, funcs, atn, sempredFuncs, superClass = {Parser}) ::= <<
using namespace antlr4;

<parser.name>::<parser.name>(TokenStream *input) : <superClass>(input) {
    _interpreter = new atn::ParserATNSimulator(this, _atn, _decisionToDFA, _sharedContextCache);
}

<parser.name>::~~<parser.name>() {
    delete _interpreter;
}

```

```

std::string <parser.name>::getGrammarFileName() const {
    return "<parser.grammarFileName>";
}

const std::vector<std::string>& <parser.name>::getRuleNames() const {
    return _ruleNames;
}

dfa::Vocabulary& <parser.name>::getVocabulary() const {
    return _vocabulary;
}

<namedActions.definitions>

<funcs; separator = "\n\n">

<if (sempredFuncs)>
bool <parser.name>::sempred(RuleContext *context, size_t ruleIndex, size_t predicateIndex) {
    switch (ruleIndex) {
        <parser.sempredFuncs.values: { f |
        case <f.ruleIndex>: return <f.name>Sempred(dynamic_cast<<f.ctxType> *(context), predicateIndex);};
        separator="\n">

    default:
        break;
    }
    return true;
}

<sempredFuncs.values; separator="\n"><endif>

// Static vars and initialization.
std::vector<dfa::DFA> <parser.name>::_decisionToDFA;
atn::PredictionContextCache <parser.name>::_sharedContextCache;

// We own the ATN which in turn owns the ATN states.
atn::ATN <parser.name>::_atn;
std::vector<uint16_t> <parser.name>::_serializedATN;

std::vector<std::string> <parser.name>::_ruleNames = {
    <parser.ruleNames: { r | "<r">}; separator = ", ", wrap, anchor>
};

std::vector<std::string> <parser.name>::_literalNames = {
    <parser.literalNames: { t | <t>}; null = "\\\"", separator = ", ", wrap, anchor>
};

```

```

std::vector<std::string> <parser.name>::_symbolicNames = {
    <parser.symbolicNames: {t | <t>}; null = "\\\"", separator = ", ", wrap, anchor>
};

dfa::Vocabulary <parser.name>::_vocabulary(_literalNames, _symbolicNames);

std::vector<std::string> <parser.name>::_tokenNames;

<parser.name>::Initializer::Initializer() {
    for (size_t i = 0; i < _symbolicNames.size(); ++i) {
        std::string name = _vocabulary.getLiteralName(i);
        if (name.empty()) {
            name = _vocabulary.getSymbolicName(i);
        }

        if (name.empty()) {
            _tokenNames.push_back("\\<INVALID>");
        } else {
            _tokenNames.push_back(name);
        }
    }
}

<atn>
}

<parser.name>::Initializer <parser.name>::_init;
>>

SerializedATNHeader(model) ::= <<
static antlr4::atn::ATN _atn;
static std::vector<uint16_t> _serializedATN;
>>

// Constructs the serialized ATN and writes init code for static member vars.
SerializedATN(model) ::= <<
<if (rest(model.segments))>
<model.segments: {segment | static uint16_t serializedATNSegment<i0>[] = {
    <segment; wrap={<\n> }>
}}}; separator="\n">

<model.segments: {segment | _serializedATN.insert(_serializedATN.end(), serializedATNSegment<i0>,
    serializedATNSegment<i0> + sizeof(serializedATNSegment<i0>) / sizeof(serializedATNSegment<i0>[0]));
}>

<else>
<! only one segment, can be inlined !>
_serializedATN = {
    <model.serialized; wrap = {<\n>}>
}

```

```

};
<endif>

atn::ATNDeserializer deserializer;
_atn = deserializer.deserialize(_serializedATN);

size_t count = _atn.getNumberOfDecisions();
_decisionToDFA.reserve(count);
for (size_t i = 0; i < count; i++) { <! Rework class ATN to allow standard iterations. !>
    _decisionToDFA.emplace_back(_atn.getDecisionState(i), i);
}
>>

RuleFunctionHeader(currentRule, args, code, locals, ruleCtx, altLabelCtxs, namedActions, finallyAction,
postamble, exceptions) ::= <<
<ruleCtx>
<! TODO: untested !><if (altLabelCtxs)><altLabelCtxs: {1 | <altLabelCtxs.(1)>}; separator="\n"><endif>
<currentRule.ctxType>* <currentRule.name><(args; separator=", ">);

>>

RuleFunction(currentRule, args, code, locals, ruleCtx, altLabelCtxs, namedActions, finallyAction, postamble,
exceptions) ::= <<
<ruleCtx>
<! TODO: untested !><altLabelCtxs: {1 | <altLabelCtxs.(1)>}; separator = "\n">
<parser.name>::<currentRule.ctxType>* <parser.name>::<currentRule.name><(args; separator=", "> {
    <currentRule.ctxType> * _localctx = _tracker.createInstance<<<currentRule.ctxType>>>(_ctx,
    getState()<currentRule.args: {a | , <a.name>}>);
    enterRule(_localctx, <currentRule.startState>, <parser.name>::Rule<currentRule.name; format = "cap">);
    <namedActions.init>
    <locals; separator = "\n">

    auto onExit = finally([=] {
        <finallyAction>
        exitRule();
    });
    try {
        <! TODO: untested !><if (currentRule.hasLookaheadBlock)>
            size_t alt;
            <endif>
            <code>
        <! TODO: untested !> <postamble; separator = "\n">
            <namedActions.after>
        }
        <if (exceptions)>
            <exceptions; separator="\n">
        <else>
            catch (RecognitionException &e) {

```

```

    _errHandler->reportError(this, e);
    _localctx->exception = std::current_exception();
    _errHandler->recover(this, _localctx->exception);
}
<endif>

```

```

return _localctx;
}
>>

```

LeftRecursiveRuleFunctionHeader(currentRule, args, code, locals, ruleCtx, altLabelCtxs, namedActions, finallyAction, postamble) ::= <<

```

<ruleCtx>
<! TODO: untested !><altLabelCtxs: {1 | <altLabelCtxs.(1)>} ; separator="\n">
<currentRule.ctxType>* <currentRule.name><(currentRule.args; separator = ", ">);
<currentRule.ctxType>* <currentRule.name><(int precedence<currentRule.args: {a | , <a>}>>);
>>

```

LeftRecursiveRuleFunction(currentRule, args, code, locals, ruleCtx, altLabelCtxs, namedActions, finallyAction, postamble) ::= <<

```

<ruleCtx>
<altLabelCtxs: {1 | <altLabelCtxs.(1)>} ; separator="\n">

<parser.name>::<currentRule.ctxType>* <parser.name>::<currentRule.name><(currentRule.args; separator=", ">) {
<! TODO: currentRule.args untested !> return <currentRule.name><(0<currentRule.args: {a | , <a.name>}>>);
}

```

```

<parser.name>::<currentRule.ctxType>* <parser.name>::<currentRule.name><(int precedence<currentRule.args:{a | , <a>}>>) {
    ParserRuleContext *parentContext = _ctx;
    size_t parentState = getState();
    <parser.name>::<currentRule.ctxType> *_localctx = _tracker.createInstance\<<currentRule.ctxType>\>(_ctx,
parentState<currentRule.args: {a | , <a.name>}>>);
    <parser.name>::<currentRule.ctxType> *previousContext = _localctx;
    (void)previousContext; // Silence compiler, in case the context is not used by generated code.
    size_t startState = <currentRule.startState>;
    enterRecursionRule(_localctx, <currentRule.startState>, <parser.name>::Rule<currentRule.name; format = "cap">,
precedence);
}

```

```

    <namedActions.init>
<! TODO: untested !> <locals; separator = "\n">

auto onExit = finally([=] {
    <if (finallyAction)><finallyAction><endif>
    unrollRecursionContexts(parentContext);
});
try {
    <if (currentRule.hasLookaheadBlock)>size_t alt;<endif>

```

```

<code>
<! TODO: untested !><postamble; separator = "\n">
  <namedActions.after>
}
catch (RecognitionException &e) {
  _errHandler->reportError(this, e);
  _localctx->exception = std::current_exception();
  _errHandler->recover(this, _localctx->exception);
}
return _localctx;
}
>>

```

```

StructDeclHeader(struct, ctorAttrs, attrs, getters, dispatchMethods, interfaces, extensionMembers) ::= <<
class <file.exportMacro> <struct.name> : public <if
(contextSuperClass)><contextSuperClass><else>antlr4::ParserRuleContext<endif><if(interfaces)>, <interfaces;
separator=", "><endif> {
public:
  <attrs: {a | <a>}; separator="\n">
  <if (ctorAttrs)><struct.name>(antlr4::ParserRuleContext *parent, size_t invokingState);<endif>
  <struct.name>(antlr4::ParserRuleContext *parent, size_t invokingState<ctorAttrs: {a | , <a>}>);
  <if (struct.provideCopyFrom)> <! don't need copy unless we have subclasses !>
  <struct.name>() = default;
  void copyFrom(<struct.name> *context);
  using antlr4::ParserRuleContext::copyFrom;
<endif>

  virtual size_t getRuleIndex() const override;
  <getters: {g | <g>}; separator = "\n">

  <dispatchMethods; separator = "\n">
  <! TODO: untested !> <extensionMembers; separator = "\n">
};

>>

```

```

StructDecl(struct, ctorAttrs, attrs, getters, dispatchMethods, interfaces, extensionMembers) ::= <<
//----- <struct.name> -----

<if (ctorAttrs)>
<parser.name>::<struct.name>::<struct.name>(ParserRuleContext *parent, size_t invokingState)
: <if (contextSuperClass)><contextSuperClass><else>ParserRuleContext<endif>(parent, invokingState) {
}
<endif>

<parser.name>::<struct.name>::<struct.name>(ParserRuleContext *parent, size_t invokingState<ctorAttrs: {a | ,
<a>}>)
: <if (contextSuperClass)><contextSuperClass><else>ParserRuleContext<endif>(parent, invokingState) {

```

```

<struct.ctorAttrs: {a | this-><a.name> = <a.name>;}; separator="\n">
}

<getters: {g | <g>}; separator = "\n">

size_t <parser.name>::<struct.name>::getRuleIndex() const {
return <parser.name>::Rule<struct.derivedFromName; format = "cap">;
}

<if (struct.provideCopyFrom)>
void <parser.name>::<struct.name>::copyFrom(<struct.name> *ctx) {
<if (contextSuperClass)><contextSuperClass><else>ParserRuleContext<endif>::copyFrom(ctx);
<struct.attrs: {a | this-><a.name> = ctx-><a.name>;}; separator = "\n">
}
<endif>
<dispatchMethods; separator = "\n\n">
<! TODO: untested !><extensionMembers; separator = "\n\n">

>>

AltLabelStructDeclHeader(struct, attrs, getters, dispatchMethods) ::= <<
class <file.exportMacro> <struct.name> : public <currentRule.name; format = "cap">Context {
public:
<struct.name>(<currentRule.name; format = "cap">Context *ctx);

<if (attrs)><attrs: {a | <a>}; separator = "\n"><endif>
<getters: {g | <g>}; separator = "\n">
<dispatchMethods; separator = "\n">
};

>>

AltLabelStructDecl(struct, attrs, getters, dispatchMethods) ::= <<
//----- <struct.name> -----

<! TODO: untested !><if (attrs)><attrs: {a | <a>}; separator = "\n"><endif>
<getters: {g | <g>}; separator = "\n">
<parser.name>::<struct.name>::<struct.name>(<currentRule.name; format = "cap">Context *ctx) { copyFrom(ctx);
}

<dispatchMethods; separator="\n">
>>

CodeBlockForOuterMostAltHeader(currentOuterMostAltCodeBlock, locals, preamble, ops) ::= "<! Required to
exist, but unused. !>"
CodeBlockForOuterMostAlt(currentOuterMostAltCodeBlock, locals, preamble, ops) ::= <<
<if (currentOuterMostAltCodeBlock.altLabel)>

```

```

_localctx = dynamic_cast<<currentRule.ctxType>
*>(_tracker.createInstance<<parser.name>::<currentOuterMostAltCodeBlock.altLabel; format =
"cap">Context>(_localctx));
<endif>
enterOuterAlt(_localctx, <currentOuterMostAltCodeBlock.alt.altNum>);
<CodeBlockForAlt(currentAltCodeBlock = currentOuterMostAltCodeBlock, ...)>
>>

CodeBlockForAltHeader(currentAltCodeBlock, locals, preamble, ops) ::= "<! Required to exist, but unused. !>"
CodeBlockForAlt(currentAltCodeBlock, locals, preamble, ops) ::= <<
<! TODO: untested !><locals; separator = "\n">
<! TODO: untested !><preamble; separator = "\n">
<ops; separator = "\n">
>>

LL1AltBlockHeader(choice, preamble, alts, error) ::= "<! Required to exist, but unused. !>"
LL1AltBlock(choice, preamble, alts, error) ::= <<
setState(<choice.stateNumber>);
_errHandler->sync(this);
<! TODO: untested !><if (choice.label)>LL1AltBlock(choice, preamble, alts, error) <labelref(choice.label)> =
_input->LT(1);<endif>
<preamble; separator="\n">
switch (_input->LA(1)) {
<choice.altLook, alts: {look, alt | <cases(ttypes = look)> {
<alt>
break;
\}
}; separator = "\n">
default:
<error>
}
>>

LL1OptionalBlockHeader(choice, alts, error) ::= "<! Required but unused. !>"
LL1OptionalBlock(choice, alts, error) ::= <<
setState(<choice.stateNumber>);
_errHandler->sync(this);
switch (_input->LA(1)) {
<choice.altLook, alts: {look, alt | <cases(ttypes = look)> {
<alt>
break;
\}
}; separator="\n">
default:
break;
}
>>

```

```

LL1OptionalBlockSingleAltHeader(choice, expr, alts, preamble, error, followExpr) ::= "<! Required but unused. !>"
LL1OptionalBlockSingleAlt(choice, expr, alts, preamble, error, followExpr) ::= <<
setState(<choice.stateNumber>);
_errHandler->sync(this);

```

```

<preamble; separator = "\n">
if (<expr>) {
  <alts; separator = "\n">
}
>>

```

```

LL1StarBlockSingleAltHeader(choice, loopExpr, alts, preamble, iteration) ::= "<! Required but unused. !>"

```

```

LL1StarBlockSingleAlt(choice, loopExpr, alts, preamble, iteration) ::= <<
setState(<choice.stateNumber>);
_errHandler->sync(this);
<preamble; separator="\n">
while (<loopExpr>) {
  <alts; separator="\n">
  setState(<choice.loopBackStateNumber>);
  _errHandler->sync(this);
  <iteration>
}
>>

```

```

LL1PlusBlockSingleAltHeader(choice, loopExpr, alts, preamble, iteration) ::= "<! Required but unused. !>"

```

```

LL1PlusBlockSingleAlt(choice, loopExpr, alts, preamble, iteration) ::= <<
setState(<choice.blockStartStateNumber>); <! alt block decision !>
_errHandler->sync(this);
<preamble; separator="\n">
do {
  <alts; separator="\n">
  setState(<choice.stateNumber>); <! loopback/exit decision !>
  _errHandler->sync(this);
  <iteration>
} while (<loopExpr>);
>>

```

```

// LL(*) stuff

```

```

AltBlockHeader(choice, preamble, alts, error) ::= "<! Unused but must be present. !>"

```

```

AltBlock(choice, preamble, alts, error) ::= <<
setState(<choice.stateNumber>);
_errHandler->sync(this);
<! TODO: untested !><if (choice.label)><labelref(choice.label)> = _input->LT(1);<endif>
<! TODO: untested !><preamble; separator = "\n">
switch (getInterpreter<atn::ParserATNSimulator>()->adaptivePredict(_input, <choice.decision>, _ctx)) {
<alts: {alt | case <i>: {
  <alt>

```

```

break;
\}
}; separator="\n">
}
>>

```

OptionalBlockHeader(choice, alts, error) ::= "<! Unused but must be present. !>"

OptionalBlock(choice, alts, error) ::= <<

setState(<choice.stateNumber>);

_errHandler->sync(this);

switch (getInterpreter<atn::ParserATNSimulator>()->adaptivePredict(_input, <choice.decision>, _ctx)) {

<alts: {alt | case <i><if (!choice.ast.greedy)> + 1<endif>: {

<alt>

break;

\}

}; separator = "\n">

}

>>

StarBlockHeader(choice, alts, sync, iteration) ::= "<! Unused but must be present. !>"

StarBlock(choice, alts, sync, iteration) ::= <<

setState(<choice.stateNumber>);

_errHandler->sync(this);

alt = getInterpreter<atn::ParserATNSimulator>()->adaptivePredict(_input, <choice.decision>, _ctx);

while (alt != <choice.exitAlt> && alt != atn::ATN::INVALID_ALT_NUMBER) {

if (alt == 1<if(!choice.ast.greedy)> + 1<endif>) {

<iteration>

<alts> <! should only be one !>

}

setState(<choice.loopBackStateNumber>);

_errHandler->sync(this);

alt = getInterpreter<atn::ParserATNSimulator>()->adaptivePredict(_input, <choice.decision>, _ctx);

}

>>

PlusBlockHeader(choice, alts, error) ::= "<! Required to exist, but unused. !>"

PlusBlock(choice, alts, error) ::= <<

setState(<choice.blockStartStateNumber>); <! alt block decision !>

_errHandler->sync(this);

alt = 1<if(!choice.ast.greedy)> + 1<endif>;

do {

switch (alt) {

<alts: {alt | case <i><if (!choice.ast.greedy)> + 1<endif>: {

<alt>

break;

\}

}; separator="\n">

```

default:
  <error>
}
setState(<choice.loopBackStateNumber>); <! loopback/exit decision !>
_errHandler->sync(this);
alt = getInterpreter\<atn::ParserATNSimulator>()->adaptivePredict(_input, <choice.decision>, _ctx);
} while (alt != <choice.exitAlt> && alt != atn::ATN::INVALID_ALT_NUMBER);
>>

Sync(s) ::= "Sync(s) sync(<s.expecting.name>);"

ThrowNoViableAltHeader(t) ::= "<! Unused but must be present. !>"
ThrowNoViableAlt(t) ::= "throw NoViableAltException(this);"

TestSetInlineHeader(s) ::= "<! Required but unused. !>"
TestSetInline(s) ::= <<
<s.bitsets: {bits | <if (rest(rest(bits.ttypes)))><bitsetBitfieldComparison(s, bits)><else><bitsetInlineComparison(s,
bits)><endif>} ; separator=" || ">
>>

// Java language spec 15.19 - shift operators mask operands rather than overflow to 0... need range test
testShiftInRange(shiftAmount) ::= <<
((<shiftAmount> & ~ 0x3fULL) == 0)
>>

// produces smaller bytecode only when bits.ttypes contains more than two items
bitsetBitfieldComparison(s, bits) ::= <<
(<testShiftInRange({ <offsetShift(s.varName, bits.shift)>})> &&
((1ULL \<< <offsetShift(s.varName, bits.shift)>) & (<bits.ttypes: {ttype | (1ULL \<< <offsetShift(ttype, bits.shift,
true)>)} ; separator = "\n | ">)) != 0)
>>

isZero ::= [
"0": true,
default: false
]

offsetShift(shiftAmount, offset, prefix = false) ::= <%
<if (!isZero.(offset))><if (prefix)><parser.name>::<endif><shiftAmount> - <offset><else><if
(prefix)><parser.name>::<endif><shiftAmount><endif>
%>

// produces more efficient bytecode when bits.ttypes contains at most two items
bitsetInlineComparison(s, bits) ::= <%
<bits.ttypes: {ttype | <s.varName> == <parser.name>::<ttype>} ; separator = "\n\n| ">
%>

cases(ttypes) ::= <<

```

```

<ttypes: { t | case <parser.name>::<t>; separator="\n">
>>

InvokeRuleHeader(r, argExprsChunks) ::= "InvokeRuleHeader"
InvokeRule(r, argExprsChunks) ::= <<
  setState(<r.stateNumber>);
  <if(r.labels)><r.labels: {1 | <labelref(1)> =
  }><endif><r.name><(if(r.ast.options.p)><r.ast.options.p><if(argExprsChunks)><endif><endif><argExprsChunks>
  );
>>

MatchTokenHeader(m) ::= "<! Required but unused. !>"
MatchToken(m) ::= <<
  setState(<m.stateNumber>);
  <if (m.labels)><m.labels: {1 | <labelref(1)> = }><endif>match(<parser.name>::<m.name>);
>>

MatchSetHeader(m, expr, capture) ::= "<! Required but unused. !>"
MatchSet(m, expr, capture) ::= "<CommonSetStuff(m, expr, capture, false)>"

MatchNotSetHeader(m, expr, capture) ::= "<! Required but unused. !>"
MatchNotSet(m, expr, capture) ::= "<CommonSetStuff(m, expr, capture, true)>"

CommonSetStuff(m, expr, capture, invert) ::= <<
  setState(<m.stateNumber>);
  <if (m.labels)><m.labels: {1 | <labelref(1)> = }>_input->LT(1);<endif>
  <capture>
  if (<if (invert)><m.varName> == 0 || <m.varName> == Token::EOF || <else>!<endif><(expr)>) {
    <if (m.labels)><m.labels: {1 | <labelref(1)> = }><endif>_errHandler->recoverInline(this);
  }
  else {
    _errHandler->reportMatch(this);
    consume();
  }
>>

WildcardHeader(w) ::= "<! Required but unused. !>"
Wildcard(w) ::= <<
  setState(<w.stateNumber>);
  <if (w.labels)><w.labels: {1 | <labelref(1)> = }><endif>matchWildcard();
>>

// ACTION STUFF

ActionHeader(a, foo, chunks) ::= "<chunks>"
Action(a, foo, chunks) ::= "<chunks>"

ArgAction(a, chunks) ::= "ArgAction(a, chunks) <chunks>"

```

```

SemPredHeader(p, chunks, failChunks) ::= "<! Required but unused. !>"
SemPred(p, chunks, failChunks) ::= <<
setState(<p.stateNumber>);

if (!(<chunks>)) throw FailedPredicateException(this, <p.predicate><if (failChunks)>, <failChunks><elseif
(p.msg)>, <p.msg><endif>);
>>

ExceptionClauseHeader(e, catchArg, catchAction) ::= "<! Required but unused. !>"
ExceptionClause(e, catchArg, catchAction) ::= <<
catch (<catchArg>) {
    <catchAction>
}
>>

// Lexer actions are not associated with model objects.

LexerSkipCommand() ::= "skip();"
LexerMoreCommand() ::= "more();"
LexerPopModeCommand() ::= "popMode();"

LexerTypeCommand(arg, grammar) ::= "type = <grammar.name>::<arg>);"
LexerChannelCommand(arg, grammar) ::= "channel = <arg>);"
LexerModeCommand(arg, grammar) ::= "mode = <grammar.name>Mode::<arg>);"
LexerPushModeCommand(arg, grammar) ::= "pushMode(<grammar.name>Mode::<arg>);"

ActionTextHeader(t) ::= "<t.text>"
ActionText(t) ::= "<t.text>"

ActionTemplateHeader(t) ::= "<! Required but unused. !>"
ActionTemplate(t) ::= "<t.st>"

ArgRefHeader(t) ::= "<! Required but unused. !>"
ArgRef(a) ::= "_localctx-><a.name>"

LocalRefHeader(t) ::= "<! Required but unused. !>"
LocalRef(a) ::= "_localctx-><a.name>"

RetValRefHeader(t) ::= "<! Required but unused. !>"
RetValRef(a) ::= "_localctx-><a.name>"

QRetValRefHeader(t) ::= "<! Required but unused. !>"
QRetValRef(a) ::= "<ctx(a)>-><a.dict>-><a.name>"
/** How to translate $tokenLabel */

TokenRefHeader(t) ::= "<! Required but unused. !>"
TokenRef(t) ::= "<ctx(t)>-><t.name>"

```

```

LabelRefHeader(t) ::= "<! Required but unused. !>"
LabelRef(t) ::= "<ctx(t)-><t.name>"

ListLabelRefHeader(t) ::= "<! Required but unused. !>"
ListLabelRef(t) ::= "<ctx(t)-><ListLabelName(t.name)>"

SetAttrHeader(t) ::= "<! Required but unused. !>"
SetAttr(s,rhsChunks) ::= "<ctx(s)-><s.name> = <rhsChunks>;"

InputSymbolType() ::= "<file.InputSymbolType; null = {Token}> *"

TokenPropertyRef_textHeader(t) ::= "<! Required but unused. !>"
TokenPropertyRef_text(t) ::= "<<(<ctx(t)-><t.label> != nullptr ? <ctx(t)-><t.label>->getText() : "")>>"

TokenPropertyRef_typeHeader(t) ::= "<! Required but unused. !>"
TokenPropertyRef_type(t) ::= "<(<ctx(t)-><t.label> != nullptr ? <ctx(t)-><t.label>->getType() : 0)"

TokenPropertyRef_lineHeader(t) ::= "<! Required but unused. !>"
TokenPropertyRef_line(t) ::= "<(<ctx(t)-><t.label> != nullptr ? <ctx(t)-><t.label>->getLine() : 0)"

TokenPropertyRef_posHeader(t) ::= "<! Required but unused. !>"
TokenPropertyRef_pos(t) ::= "<(<ctx(t)-><t.label> != nullptr ? <ctx(t)-><t.label>->getCharPositionInLine() : 0)"

TokenPropertyRef_channelHeader(t) ::= "<! Required but unused. !>"
TokenPropertyRef_channel(t) ::= "<(<ctx(t)-><t.label> != nullptr ? <ctx(t)-><t.label>->getChannel() : 0)"

TokenPropertyRef_indexHeader(t) ::= "<! Required but unused. !>"
TokenPropertyRef_index(t) ::= "<(<ctx(t)-><t.label> != nullptr ? <ctx(t)-><t.label>->getTokenIndex() : 0)"

TokenPropertyRef_intHeader(t) ::= "<! Required but unused. !>"
TokenPropertyRef_int(t) ::= "<(<ctx(t)-><t.label> != nullptr ? std::stoi(<ctx(t)-><t.label>->getText()) : 0)"

RulePropertyRef_startHeader(r) ::= "<! Required but unused. !>"
RulePropertyRef_start(r) ::= "<(<ctx(r)-><r.label> != nullptr ? (<ctx(r)-><r.label>->start) : nullptr)"

RulePropertyRef_stopHeader(r) ::= "<! Required but unused. !>"
RulePropertyRef_stop(r) ::= "<(<ctx(r)-><r.label> != nullptr ? (<ctx(r)-><r.label>->stop) : nullptr)"

RulePropertyRef_textHeader(r) ::= "<! Required but unused. !>"
RulePropertyRef_text(r) ::= "<(<ctx(r)-><r.label> != nullptr ? _input->getText(<ctx(r)-><r.label>->start,
<ctx(r)-><r.label>->stop) : nullptr)"

RulePropertyRef_ctxHeader(r) ::= "<! Required but unused. !>"
RulePropertyRef_ctx(r) ::= "<ctx(r)-><r.label>"

ThisRulePropertyRef_start(r) ::= "ThisRulePropertyRef_start(r) _localctx->start"
ThisRulePropertyRef_stop(r) ::= "ThisRulePropertyRef_stop(r) _localctx->stop"

```

```

ThisRulePropertyRef_textHeader(r) ::= "<! Required but unused. !>"
ThisRulePropertyRef_text(r) ::= "_input->getText(_localctx->start, _input->LT(-1))"

ThisRulePropertyRef_ctxHeader(r) ::= "<! Required but unused. !>"
ThisRulePropertyRef_ctx(r) ::= "_localctx"

ThisRulePropertyRef_parserHeader(r) ::= "<! Required but unused. !>"
ThisRulePropertyRef_parser(r) ::= "this"

NonLocalAttrRef(s) ::= "NonLocalAttrRef(s) ((<s.ruleName;
format=\"cap\">Context)getInvokingContext(<s.ruleIndex>)).<s.name>"
SetNonLocalAttr(s, rhsChunks) ::=
"SetNonLocalAttr(s, rhsChunks) ((<s.ruleName;
format=\"cap\">Context)getInvokingContext(<s.ruleIndex>)).<s.name> = <rhsChunks>;"

AddToLabelListHeader(a) ::= "<! Required but unused. !>"
AddToLabelList(a) ::= <<
<ctx(a.label)>-><a.listName>.push_back(<labelref(a.label)>);
>>

TokenLabelType() ::= "<file.TokenLabelType; null = {Token}> *"

TokenDeclHeader(t) ::= "antlr4::<TokenLabelType()><t.name> = nullptr;"
TokenDecl(t) ::= "<! Variable Declaration !>"

TokenTypeDeclHeader(t) ::= "<! Local Variable !>"
TokenTypeDecl(t) ::= "size_t <t.name> = 0;"

TokenListDeclHeader(t) ::= "std::vector<antlr4::Token *> <t.name>;"
TokenListDecl(t) ::= "<! Variable Declaration !>"

RuleContextDeclHeader(r) ::= "<parser.name>::<r.ctxName> *<r.name> = nullptr;"
RuleContextDecl(r) ::= "<! Variable Declaration !>"

RuleContextListDeclHeader(rdecl) ::= "std::vector<<rdecl.ctxName> *> <rdecl.name>;"
RuleContextListDecl(rdecl) ::= "<! Variable Declaration !>"

ContextTokenGetterDeclHeader(t) ::= "antlr4::tree::TerminalNode *<t.name>();"
ContextTokenGetterDecl(t) ::= <<
tree::TerminalNode* <parser.name>::<t.ctx.name>::<t.name>() {
    return getToken(<parser.name>::<t.name>, 0);
}
>>

ContextTokenListGetterDeclHeader(t) ::= "std::vector<antlr4::tree::TerminalNode *> <t.name>();"
ContextTokenListGetterDecl(t) ::= <<

```

```

std::vector<tree::TerminalNode *> <parser.name>::<t.ctx.name>::<t.name>() {
    return getTokens(<parser.name>::<t.name>);
}

>>

ContextTokenListIndexedGetterDeclHeader(t) ::= "antlr4::tree::TerminalNode* <t.name>(size_t i);"
ContextTokenListIndexedGetterDecl(t) ::= <<
tree::TerminalNode* <parser.name>::<t.ctx.name>::<t.name>(size_t i) {
    return getToken(<parser.name>::<t.name>, i);
}

>>

ContextRuleGetterDeclHeader(r) ::= "<r.ctxName> *<r.name>()";
ContextRuleGetterDecl(r) ::= <<
<! Note: ctxName is the name of the context to return, while ctx is the owning context. !>
<parser.name>::<r.ctxName>* <parser.name>::<r.ctx.name>::<r.name>() {
    return getRuleContext<<parser.name>::<r.ctxName>>(0);
}

>>

ContextRuleListGetterDeclHeader(r) ::= "std::vector<<r.ctxName> *> <r.name>()";
ContextRuleListGetterDecl(r) ::= <<
std::vector<<parser.name>::<r.ctxName> *> <parser.name>::<r.ctx.name>::<r.name>() {
    return getRuleContexts<<parser.name>::<r.ctxName>>(0);
}

>>

ContextRuleListIndexedGetterDeclHeader(r) ::= "<r.ctxName>* <r.name>(size_t i);"
ContextRuleListIndexedGetterDecl(r) ::= <<
<parser.name>::<r.ctxName>* <parser.name>::<r.ctx.name>::<r.name>(size_t i) {
    return getRuleContext<<parser.name>::<r.ctxName>>(i);
}

>>

LexerRuleContext() ::= "antlr4::RuleContext"

// The rule context name is the rule followed by a suffix; e.g. r becomes rContext.
RuleContextNameSuffix() ::= "Context"

ImplicitTokenLabel(tokenName) ::= <<
<tokenName; format = "lower">Token

>>

```

```

ImplicitRuleLabel(ruleName) ::= "<ruleName>Context"
ImplicitSetLabel(id) ::= "_tset<id>"
ListLabelName(label) ::= "<label>"

CaptureNextToken(d) ::= "CaptureNextToken(d) <d.varName> = _input->LT(1);"

CaptureNextTokenTypeHeader(d) ::= "<! Required but unused. !>"
CaptureNextTokenType(d) ::= "<d.varName> = _input->LA(1);"

ListenerDispatchMethodHeader(method) ::= <<
virtual void <if (method.isEnter)>enter<else>exit<endif>Rule(antlr4::tree::ParseTreeListener *listener) override;
>>
ListenerDispatchMethod(method) ::= <<
void <parser.name>::<struct.name>::<if (method.isEnter)>enter<else>exit<endif>Rule(tree::ParseTreeListener
*listener) {
    auto parserListener = dynamic_cast<<parser.grammarName>Listener *>(listener);
    if (parserListener != nullptr)
        parserListener-><if (method.isEnter)>enter<else>exit<endif><struct.derivedFromName; format="cap">(this);
}
>>

VisitorDispatchMethodHeader(method) ::= <<

virtual antlr4::Any accept(antlr4::tree::ParseTreeVisitor *visitor) override;
>>
VisitorDispatchMethod(method) ::= <<

antlr4::Any <parser.name>::<struct.name>::accept(tree::ParseTreeVisitor *visitor) {
    if (auto parserVisitor = dynamic_cast<<parser.grammarName>Visitor*>(visitor))
        return parserVisitor->visit<struct.derivedFromName; format="cap">(this);
    else
        return visitor->visitChildren(this);
}
>>

AttributeDeclHeader(d) ::= "<d.type> <d.name><if (d.initValue)> = <d.initValue><endif>"
AttributeDecl(d) ::= "<d.type> <d.name>"

/** If we don't know location of label def x, use this template */
labelref(x) ::= "<if (!x.isLocal)>dynamic_cast<<x.ctx.name> *>(_localctx)-><endif><x.name>"

/** For any action chunk, what is correctly-typed context struct ptr? */
ctx(actionChunk) ::= "dynamic_cast<<actionChunk.ctx.name> *>(_localctx)"

// used for left-recursive rules
recRuleAltPredicate(ruleName,opPrec) ::= "precpred(_ctx, <opPrec>)"
recRuleSetReturnAction(src,name) ::= "recRuleSetReturnAction(src,name) $<name>=$<src>.<name>;"
recRuleSetStopToken() ::= "_ctx->stop = _input->LT(-1);"

```

```

recRuleAltStartAction(ruleName, ctxName, label) ::= <<
_localctx = _tracker.createInstance<<ctxName>>Context>(parentContext, parentState);
<if (label)>_localctx-><label> = previousContext;<endif>
pushNewRecursionContext(_localctx, startState, Rule<ruleName; format = "cap">);
>>

// Separate context variable to avoid frequent pointer type casts.
recRuleLabeledAltStartAction(ruleName, currentAltLabel, label, isListLabel) ::= <<
auto newContext = _tracker.createInstance<<currentAltLabel; format =
"cap">>Context>(_tracker.createInstance<<ruleName; format="cap">>Context>(parentContext, parentState));
_localctx = newContext;
<if(label)>
<if(isListLabel)>
newContext-><label>.push_back(previousContext);
<else>
newContext-><label> = previousContext;
<endif>
<endif>
pushNewRecursionContext(newContext, startState, Rule<ruleName; format = "cap">);
>>

recRuleReplaceContext(ctxName) ::= <<
_localctx = _tracker.createInstance<<ctxName>>Context>(_localctx);
_ctx = _localctx;
previousContext = _localctx;
>>

recRuleSetPrevCtx() ::= <<
if (!_parseListeners.empty())
triggerExitRuleEvent();
previousContext = _localctx;
>>

/** Using a type to init value map, try to init a type; if not in table
 * must be an object, default value is "null".
 */
initValue(typeName) ::= <<
<CppTypeInitMap.(typeName)>
>>

Found in path(s):
* /opt/cola/permits/1473460068_1668481512.5494933/0/antlr4-4-8-1-
jar/org/antlr/v4/tool/templates/codegen/Cpp/Cpp.stg
No license file was found, but licenses were detected in source scan.

/*
[The "BSD licence"]

```

Copyright (c) 2006 Kay Roepke

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

/*

This file contains the actual layout of the messages emitted by ANTLR.

The text itself is coming out of the languages/*stg files, according to the chosen locale.

This file contains the format that mimicks GCC output.

*/

location(file, line, column) ::= "<file>:<line>:<column>:"

message(id, text) ::= "<text> [error <id>]"

report(location, message, type) ::= "<location> <type>: <message>"

wantsSingleLineMessage() ::= "true"

Found in path(s):

* /opt/cola/permits/1473460068_1668481512.5494933/0/antlr4-4-8-1-jar/org/antlr/v4/tool/templates/messages/formats/gnu.stg

No license file was found, but licenses were detected in source scan.

/*

* [The "BSD license"]

* Copyright (c) 2012-2016 Terence Parr

* Copyright (c) 2012-2016 Sam Harwell

```

* Copyright (c) 2014 Eric Vergnaud
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products
* derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

/** ANTLR tool checks output templates are compatible with tool code generation.
* For now, a simple string match used on x.y of x.y.z scheme.
* Must match Tool.VERSION during load to templates.
*
* REQUIRED.
*/

pythonTypeInitMap ::= [
  "bool":"False",
  "int":"0",
  "float":"0.0",
  "str": "",
  default:"None" // anything other than a primitive type is an object
]

// args must be <object-model-object>, <fields-resulting-in-STs>

ParserFile(file, parser, namedActions, contextSuperClass) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
# encoding: utf-8
from __future__ import print_function

```

```
from antlr4 import *
from io import StringIO
import sys
```

```
<namedActions.header>
<parser>
```

```
>>
```

```
ListenerFile(file, header, namedActions) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
from antlr4 import *
<header>
```

```
# This class defines a complete listener for a parse tree produced by <file.parserName>.
class <file.grammarName>Listener(ParseTreeListener):
```

```
    <file.listenerNames: {lname |
# Enter a parse tree produced by <file.parserName>#<lname>.
def enter<lname; format="cap">(self, ctx):
    pass

# Exit a parse tree produced by <file.parserName>#<lname>.
def exit<lname; format="cap">(self, ctx):
    pass

}; separator="\n">
```

```
>>
```

```
VisitorFile(file, header, namedActions) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
from antlr4 import *
<header>
```

```
# This class defines a complete generic visitor for a parse tree produced by <file.parserName>.
```

```
class <file.grammarName>Visitor(ParseTreeVisitor):
```

```
    <file.visitorNames: {lname |
# Visit a parse tree produced by <file.parserName>#<lname>.
def visit<lname; format="cap">(self, ctx):
    return self.visitChildren(ctx)

}; separator="\n">
```

```
>>
```

```

fileHeader(grammarFileName, ANTLRVersion) ::= <<
# Generated from <grammarFileName> by ANTLR <ANTLRVersion>
>>

Parser(parser, funcs, atn, sempredFuncs, superClass) ::= <<
<Parser_(ctor="parser_ctor", ...)>
>>

Parser_(parser, funcs, atn, sempredFuncs, ctor, superClass) ::= <<
<if(superClass)>
if __name__ is not None and "." in __name__:
    from .<superClass> import <superClass>
else:
    from <superClass> import <superClass>

<endif>
<atn>

class <parser.name> ( <if(superClass)><superClass><else>Parser<endif> ):

    grammarFileName = "<parser.grammarFileName>"

    atn = ATNDeserializer().deserialize(serializedATN())

    decisionsToDFA = [ DFA(ds, i) for i, ds in enumerate(atn.decisionToState) ]

    sharedContextCache = PredictionContextCache()

    literalNames = [ <parser.literalNames:{t | u<t>}; null="u\"<INVALID>\"", separator=", ", wrap, anchor> ]

    symbolicNames = [ <parser.symbolicNames:{t | u<t>}; null="u\"<INVALID>\"", separator=", ", wrap, anchor> ]

    <parser.rules:{r | RULE_<r.name> = <r.index>}; separator="\n", wrap, anchor>

    ruleNames = [ <parser.ruleNames:{r | u"<r>"}; separator=", ", wrap, anchor> ]

    EOF = <TokenLabelType().EOF
    <if(parser.tokens)>
    <parser.tokens:{k | <k>=<parser.tokens.(k)>}; separator="\n", wrap, anchor>
    <endif>

    <parser:(ctor)()>

    <namedActions.members>

    <funcs; separator="\n">

```

```

<if(sempredFuncs)>
  def sempred(self, localctx, ruleIndex, predIndex):
    if self._predicates == None:
      self._predicates = dict()
<parser.sempredFuncs.values: {f |
  self._predicates[<f.ruleIndex>] = self.<f.name>_sempred}; separator="\n
  " >
  pred = self._predicates.get(ruleIndex, None)
  if pred is None:
    raise Exception("No predicate with index:" + str(ruleIndex))
  else:
    return pred(localctx, predIndex)

  <sempredFuncs.values; separator="\n">
<endif>

```

```
>>
```

```

dumpActions(recog, argFuncs, actionFuncs, sempredFuncs) ::= <<
<if(actionFuncs)>
def action(self, localctx, ruleIndex, actionIndex):
  if self._actions is None:
    actions = dict()
<recog.actionFuncs.values: {f|
  actions[<f.ruleIndex>] = self.<f.name>_action }; separator="\n">
  self._actions = actions
  action = self._actions.get(ruleIndex, None)
  if action is not None:
    action(localctx, actionIndex)
  else:
    raise Exception("No registered action for:" + str(ruleIndex))

<actionFuncs.values; separator="\n">

<endif>
<if(sempredFuncs)>
def sempred(self, localctx, ruleIndex, predIndex):
  if self._predicates is None:
    preds = dict()
<recog.sempredFuncs.values: {f|
  preds[<f.ruleIndex>] = self.<f.name>_sempred}; separator="\n">
  self._predicates = preds
  pred = self._predicates.get(ruleIndex, None)
  if pred is not None:
    return pred(localctx, predIndex)

```

```

else:
    raise Exception("No registered predicate for:" + str(ruleIndex))

<sempredFuncs.values; separator="\n">
<endif>
>>

parser_ctor(p) ::= <<
def __init__(self, input, output=sys.stdout):
    super(<parser.name>, self).__init__(input, output=output)
    self.checkVersion("<file.ANTLRVersion>")
    self._interp = ParserATNSimulator(self, self.atn, self.decisionsToDFA, self.sharedContextCache)
    self._predicates = None

>>

/* This generates a private method since the actionIndex is generated, making an
* overriding implementation impossible to maintain.
*/
RuleActionFunction(r, actions) ::= <<

def <r.name>_action(self, localctx , actionIndex):
<actions: {index|
<if(first(actions))>
    if actionIndex == <index>:
        <actions.(index)>
<elseif(rest(actions))>
    elif actionIndex == <index>:
        <actions.(index)>
<endif> }; separator="\n">
>>

/* This generates a private method since the predIndex is generated, making an
* overriding implementation impossible to maintain.
*/
RuleSempredFunction(r, actions) ::= <<
def <r.name>_sempred(self, localctx, predIndex):
    <actions: {index|
<if(first(actions))>
    if predIndex == <index>:
        return <actions.(index)>
<elseif(rest(actions))>
    elif predIndex == <index>:
        return <actions.(index)>
<endif> }; separator="\n">

>>

```

```
RuleFunction(currentRule,args,code,locals,ruleCtx,altLabelCtxs,namedActions,finallyAction,postamble,exceptions)
```

```
::= <<
```

```
<ruleCtx>
```

```
<altLabelCtxs:{l | <altLabelCtxs.(l)>}; separator="\n">
```

```
def <currentRule.name>(self<currentRule.args:{a | , <a.name>}>):
```

```
  localctx = <parser.name>.<currentRule.ctxType>(self, self._ctx, self.state<currentRule.args:{a | , <a.name>}>)
```

```
  self.enterRule(localctx, <currentRule.startState>, self.RULE_<currentRule.name>)
```

```
  <namedActions.init>
```

```
  <locals; separator="\n">
```

```
  try:
```

```
    <code>
```

```
    <postamble; separator="\n">
```

```
    <namedActions.after>
```

```
  <if(exceptions)>
```

```
  <exceptions; separator="\n">
```

```
  <else>
```

```
  except RecognitionException as re:
```

```
    localctx.exception = re
```

```
    self._errHandler.reportError(self, re)
```

```
    self._errHandler.recover(self, re)
```

```
  <endif>
```

```
  finally:
```

```
    <finallyAction>
```

```
    self.exitRule()
```

```
  return localctx
```

```
>>
```

```
LeftRecursiveRuleFunction(currentRule,args,code,locals,ruleCtx,altLabelCtxs,
```

```
namedActions,finallyAction,postamble) ::=
```

```
<<
```

```
<ruleCtx>
```

```
<altLabelCtxs:{l | <altLabelCtxs.(l)>}; separator="\n">
```

```
def <currentRule.name>(self, _p=0<if(currentRule.args)>, <args:{a | , <a>}><endif>):
```

```
  _parentctx = self._ctx
```

```
  _parentState = self.state
```

```
  localctx = <parser.name>.<currentRule.ctxType>(self, self._ctx, _parentState<args:{a | , <a.name>}>)
```

```
  _prevctx = localctx
```

```
  _startState = <currentRule.startState>
```

```
  self.enterRecursionRule(localctx, <currentRule.startState>, self.RULE_<currentRule.name>, _p)
```

```
  <namedActions.init>
```

```
  <locals; separator="\n">
```

```

try:
    <code>
    <postamble; separator="\n">
    <namedActions.after>
except RecognitionException as re:
    localctx.exception = re
    self._errHandler.reportError(self, re)
    self._errHandler.recover(self, re)
finally:
    <finallyAction>
    self.unrollRecursionContexts(_parentctx)
return localctx

>>

CodeBlockForOuterMostAlt(currentOuterMostAltCodeBlock, locals, preamble, ops) ::= <<
<if(currentOuterMostAltCodeBlock.altLabel)>localctx = <parser.name>.<currentOuterMostAltCodeBlock.altLabel;
format="cap">Context(self, localctx)<endif>
self.enterOuterAlt(localctx, <currentOuterMostAltCodeBlock.alt.altNum>)
<CodeBlockForAlt(currentAltCodeBlock=currentOuterMostAltCodeBlock, ...)>
>>

CodeBlockForAlt(currentAltCodeBlock, locals, preamble, ops) ::= <<
<locals; separator="\n">
<preamble; separator="\n">
<ops; separator="\n">
>>

LL1AltBlock(choice, preamble, alts, error) ::= <<
self.state = <choice.stateNumber>
self._errHandler.sync(self)
<if(choice.label)><labelref(choice.label)> = _input.LT(1)<endif>
<preamble; separator="\n">
token = self._input.LA(1)
<choice.altLook,alts:{look,alt| <cases(ttypes=look)>
    <alt>
    pass}; separator="\n1">
else:
    <error>

>>

LL1OptionalBlock(choice, alts, error) ::= <<
self.state = <choice.stateNumber>
self._errHandler.sync(self)
token = self._input.LA(1)
<choice.altLook,alts:{look,alt| <cases(ttypes=look)>
    <alt>

```

```

    pass }; separator="\n">
else:
    pass
>>

```

```

LL1OptionalBlockSingleAlt(choice, expr, alts, preamble, error, followExpr) ::= <<
self.state = <choice.stateNumber>
self._errHandler.sync(self)
<preamble; separator="\n">
if <expr>:
    <alts; separator="\n">

<!else if ( !(<followExpr> ) <error>!>
>>

```

```

LL1StarBlockSingleAlt(choice, loopExpr, alts, preamble, iteration) ::= <<
self.state = <choice.stateNumber>
self._errHandler.sync(self)
<preamble; separator="\n">
while <loopExpr>:
    <alts; separator="\n">
    self.state = <choice.loopBackStateNumber>
    self._errHandler.sync(self)
    <iteration>

>>

```

```

LL1PlusBlockSingleAlt(choice, loopExpr, alts, preamble, iteration) ::= <<
self.state = <choice.blockStartStateNumber> <! alt block decision !>
self._errHandler.sync(self)
<preamble; separator="\n">
while True:
    <alts; separator="\n">
    self.state = <choice.stateNumber> <! loopback/exit decision !>
    self._errHandler.sync(self)
    <iteration>
    if not (<loopExpr>):
        break

>>

```

```

// LL(*) stuff

```

```

AltBlock(choice, preamble, alts, error) ::= <<
self.state = <choice.stateNumber>
self._errHandler.sync(self)
<if(choice.label)><labelref(choice.label)> = _input.LT(1)<endif>

```

```

<preamble; separator="\n">
la_ = self._interp.adaptivePredict(self._input,<choice.decision>,self._ctx)
<alts:{ alt |
if la_ == <i>:
    <alt>
    pass
}; separator="\nel">

>>

```

```

OptionalBlock(choice, alts, error) ::= <<
self.state = <choice.stateNumber>
self._errHandler.sync(self)
la_ = self._interp.adaptivePredict(self._input,<choice.decision>,self._ctx)
<alts:{ alt |
if la_ == <i><if(!choice.ast.greedy)>+1<endif>:
    <alt>
}; separator="\nel">

>>

```

```

StarBlock(choice, alts, sync, iteration) ::= <<
self.state = <choice.stateNumber>
self._errHandler.sync(self)
_alt = self._interp.adaptivePredict(self._input,<choice.decision>,self._ctx)
while _alt! =<choice.exitAlt> and _alt!=ATN.INVALID_ALT_NUMBER:
    if _alt==1<if(!choice.ast.greedy)>+1<endif>:
        <iteration>
        <alts> <! should only be one !>
        self.state = <choice.loopBackStateNumber>
        self._errHandler.sync(self)
        _alt = self._interp.adaptivePredict(self._input,<choice.decision>,self._ctx)

>>

```

```

PlusBlock(choice, alts, error) ::= <<
self.state = <choice.blockStartStateNumber> <! alt block decision !>
self._errHandler.sync(self)
_alt = 1<if(!choice.ast.greedy)>+1<endif>
while _alt! =<choice.exitAlt> and _alt!=ATN.INVALID_ALT_NUMBER:
    <alts:{ alt|
if _alt == <i><if(!choice.ast.greedy)>+1<endif>:
    <alt>
}; separator="\nel">
    else:
        <error>
        self.state = <choice.loopBackStateNumber> <! loopback/exit decision !>
        self._errHandler.sync(self)

```

```

    _alt = self._interp.adaptivePredict(self._input,<choice.decision>,self._ctx)

>>

Sync(s) ::= "sync(<s.expecting.name>)"

ThrowNoViableAlt(t) ::= "raise NoViableAltException(self)"

TestSetInline(s) ::= <<
<s.bitsets:{bits | <if(rest(rest(bits.ttypes)))><bitsetBitfieldComparison(s, bits)><else><bitsetInlineComparison(s,
bits)><endif>} ; separator=" or ">
>>

// Java language spec 15.19 - shift operators mask operands rather than overflow to 0... need range test
testShiftInRange(shiftAmount) ::= <<
((<shiftAmount>) & ~0x3f) == 0
>>

// produces smaller bytecode only when bits.ttypes contains more than two items
bitsetBitfieldComparison(s, bits) ::= <%
(<testShiftInRange({<offsetShiftVar(s.varName, bits.shift)>})> and ((1 \<< <offsetShiftVar(s.varName,
bits.shift)>) & (<bits.ttypes:{ttype | (1 \<< <offsetShiftType(ttype, bits.shift)>)}); separator=" | ">)) != 0)
%>

isZero ::= [
"0":true,
default:false
]

offsetShiftVar(shiftAmount, offset) ::= <%
<if(!isZero.(offset))><shiftAmount> - <offset><else><shiftAmount><endif>
%>

offsetShiftType(shiftAmount, offset) ::= <%
<if(!isZero.(offset))><parser.name>.<shiftAmount> - <offset><else><parser.name>.<shiftAmount><endif>
%>

// produces more efficient bytecode when bits.ttypes contains at most two items
bitsetInlineComparison(s, bits) ::= <%
<bits.ttypes:{ttype | <s.varName>==<parser.name>.<ttype>} ; separator=" or ">
%>

cases(ttypes) ::= <<
if token in [<ttypes:{t | <parser.name>.<t>} ; separator=", ">]:
>>

InvokeRule(r, argExprsChunks) ::= <<
self.state = <r.stateNumber>

```

```

<if(r.labels)><r.labels:{1 | <labelref(1)> =
}><endif>self.<r.name>( <if(r.ast.options.p)><r.ast.options.p><if(argExprsChunks)><endif><endif><argExprsChu
nks>)
>>

```

```

MatchToken(m) ::= <<
self.state = <m.stateNumber>
<if(m.labels)><m.labels:{1 | <labelref(1)> = }><endif>self.match(<parser.name>.<m.name>)
>>

```

```

MatchSet(m, expr, capture) ::= "<CommonSetStuff(m, expr, capture, false)>"

```

```

MatchNotSet(m, expr, capture) ::= "<CommonSetStuff(m, expr, capture, true)>"

```

```

CommonSetStuff(m, expr, capture, invert) ::= <<
self.state = <m.stateNumber>
<if(m.labels)><m.labels:{1 | <labelref(1)> = }>self._input.LT(1)<endif>
<capture>
<if(invert)>if <m.varName> \<= 0 or <expr><else>if not(<expr>)<endif>:
<if(m.labels)><m.labels:{1 | <labelref(1)> = }><else> <endif>self._errHandler.recoverInline(self)
else:
    self._errHandler.reportMatch(self)
    self.consume()
>>

```

```

Wildcard(w) ::= <<
self.state = <w.stateNumber>
<if(w.labels)><w.labels:{1 | <labelref(1)> = }><endif>self.matchWildcard()
>>

```

```

// ACTION STUFF

```

```

Action(a, foo, chunks) ::= "<chunks>"

```

```

ArgAction(a, chunks) ::= "<chunks>"

```

```

SemPred(p, chunks, failChunks) ::= <<
self.state = <p.stateNumber>
if not <chunks>:
    from antlr4.error.Errors import FailedPredicateException
    raise FailedPredicateException(self, <p.predicate><if(failChunks)>, <failChunks><elseif(p.msg)>,
<p.msg><endif>)
>>

```

```

ExceptionClause(e, catchArg, catchAction) ::= <<
catch (<catchArg>) {
    <catchAction>
}

```

>>

// lexer actions are not associated with model objects

LexerSkipCommand() ::= "skip()"

LexerMoreCommand() ::= "more()"

LexerPopModeCommand() ::= "popMode()"

LexerTypeCommand(arg, grammar) ::= "_type = <arg>"

LexerChannelCommand(arg, grammar) ::= "_channel = <arg>"

LexerModeCommand(arg, grammar) ::= "_mode = <arg>"

LexerPushModeCommand(arg, grammar) ::= "pushMode(<arg>)"

ActionText(t) ::= "<t.text>"

ActionTemplate(t) ::= "<t.st>"

ArgRef(a) ::= "localctx.<a.name>"

LocalRef(a) ::= "localctx.<a.name>"

RetValRef(a) ::= "localctx.<a.name>"

QRetValRef(a) ::= "<ctx(a)>.<a.dict>.<a.name>"

/** How to translate \$tokenLabel */

TokenRef(t) ::= "<ctx(t)>.<t.name>"

LabelRef(t) ::= "<ctx(t)>.<t.name>"

ListLabelRef(t) ::= "<ctx(t)>.<ListLabelName(t.name)>"

SetAttr(s,rhsChunks) ::= "<ctx(s)>.<s.name> = <rhsChunks>"

TokenLabelType() ::= "<file.TokenLabelType; null={Token}>"

InputSymbolType() ::= "<file.InputSymbolType; null={Token}>"

TokenPropertyRef_text(t) ::= "(None if <ctx(t)>.<t.label> is None else <ctx(t)>.<t.label>.text)"

TokenPropertyRef_type(t) ::= "(0 if <ctx(t)>.<t.label> is None else <ctx(t)>.<t.label>.type)"

TokenPropertyRef_line(t) ::= "(0 if <ctx(t)>.<t.label> is None else <ctx(t)>.<t.label>.line)"

TokenPropertyRef_pos(t) ::= "(0 if <ctx(t)>.<t.label> is None else <ctx(t)>.<t.label>.column)"

TokenPropertyRef_channel(t) ::= "(0 if (<ctx(t)>.<t.label> is None else <ctx(t)>.<t.label>.channel)"

TokenPropertyRef_index(t) ::= "(0 if <ctx(t)>.<t.label> is None else <ctx(t)>.<t.label>.tokenIndex)"

TokenPropertyRef_int(t) ::= "(0 if <ctx(t)>.<t.label> is None else int(<ctx(t)>.<t.label>.text))"

RulePropertyRef_start(r) ::= "(None if <ctx(r)>.<r.label> is None else <ctx(r)>.<r.label>.start)"

RulePropertyRef_stop(r) ::= "(None if <ctx(r)>.<r.label> is None else <ctx(r)>.<r.label>.stop)"

RulePropertyRef_text(r) ::= "(None if <ctx(r)>.<r.label> is None else
self._input.getText(<ctx(r)>.<r.label>.start,<ctx(r)>.<r.label>.stop))"

RulePropertyRef_ctx(r) ::= "<ctx(r)>.<r.label>"

RulePropertyRef_parser(r) ::= "self"

ThisRulePropertyRef_start(r) ::= "localctx.start"

ThisRulePropertyRef_stop(r) ::= "localctx.stop"

ThisRulePropertyRef_text(r) ::= "self._input.getText(localctx.start, self._input.LT(-1))"

ThisRulePropertyRef_ctx(r) ::= "localctx"

ThisRulePropertyRef_parser(r) ::= "self"

```

NonLocalAttrRef(s) ::= "getInvokingContext(<s.ruleIndex>).<s.name>"
SetNonLocalAttr(s, rhsChunks) ::= "getInvokingContext(<s.ruleIndex>).<s.name> = <rhsChunks>"

AddToLabelList(a) ::= "<ctx(a.label)>.<a.listName>.append(<labelref(a.label)>)"

TokenDecl(t) ::= "self.<t.name> = None # <TokenLabelType()>"
TokenTypeDecl(t) ::= "self.<t.name> = 0 # <TokenLabelType()> type"
TokenListDecl(t) ::= "self.<t.name> = list() # of <TokenLabelType()>s"
RuleContextDecl(r) ::= "self.<r.name> = None # <r.ctxName>"
RuleContextListDecl(rdecl) ::= "self.<rdecl.name> = list() # of <rdecl.ctxName>s"

ContextTokenGetterDecl(t) ::= <<
def <t.name>(self):
    return self.getToken(<parser.name>.<t.name>, 0)
>>

// should never be called
ContextTokenListGetterDecl(t) ::= <<
def <t.name>_list(self):
    return self.getTokens(<parser.name>.<t.name>)
>>

ContextTokenListIndexedGetterDecl(t) ::= <<
def <t.name>(self, i=None):
    if i is None:
        return self.getTokens(<parser.name>.<t.name>)
    else:
        return self.getToken(<parser.name>.<t.name>, i)
>>

ContextRuleGetterDecl(r) ::= <<
def <r.name>(self):
    return self.getTypedRuleContext(<parser.name>.<r.ctxName>,0)
>>

// should never be called
ContextRuleListGetterDecl(r) ::= <<
def <r.name>_list(self):
    return self.getTypedRuleContexts(<parser.name>.<r.ctxName>)
>>

ContextRuleListIndexedGetterDecl(r) ::= <<
def <r.name>(self, i=None):
    if i is None:
        return self.getTypedRuleContexts(<parser.name>.<r.ctxName>)

```

```

else:
    return self.getTypedRuleContext(<parser.name>.<r.ctxName>,i)

>>

LexerRuleContext() ::= "RuleContext"

/** The rule context name is the rule followed by a suffix; e.g.,
 * r becomes rContext.
 */
RuleContextNameSuffix() ::= "Context"

ImplicitTokenLabel(tokenName) ::= "_<tokenName>"
ImplicitRuleLabel(ruleName) ::= "_<ruleName>"
ImplicitSetLabel(id) ::= "_tset<id>"
ListLabelName(label) ::= "<label>"

CaptureNextToken(d) ::= "<d.varName> = self._input.LT(1)"
CaptureNextTokenType(d) ::= "<d.varName> = self._input.LA(1)"

StructDecl(struct,ctorAttrs,attrs,getters,dispatchMethods,interfaces,extensionMembers) ::= <<
class <struct.name>(<if(contextSuperClass)><contextSuperClass><else>ParserRuleContext<endif>):

    def __init__(self, parser, parent=None, invokingState=-1<struct.ctorAttrs:{a | , <a.name>=None}>):
        super(<parser.name>.<struct.name>, self).__init__(parent, invokingState)
        self.parser = parser
        <attrs:{a | <a>}; separator="\n">
        <struct.ctorAttrs:{a | self.<a.name> = <a.name>}; separator="\n">

        <getters:{g | <g>}; separator="\n\n">

    def getRuleIndex(self):
        return <parser.name>.RULE_<struct.derivedFromName>

<if(struct.provideCopyFrom)> <! don't need copy unless we have subclasses !>
    def copyFrom(self, ctx):
        super(<parser.name>.<struct.name>, self).copyFrom(ctx)
        <struct.attrs:{a | self.<a.name> = ctx.<a.name>}; separator="\n">

<endif>
    <dispatchMethods; separator="\n">
    <extensionMembers; separator="\n">

>>

AltLabelStructDecl(struct,attrs,getters,dispatchMethods) ::= <<
class <struct.name>(<currentRule.name; format="cap">Context):

```

```

def __init__(self, parser, ctx): # actually a <parser.name>.<currentRule.name; format="cap">Context)
    super(<parser.name>.<struct.name>, self).__init__(parser)
    <attrs:{a | <a>}; separator="\n">
    self.copyFrom(ctx)

<getters:{g | <g>}; separator="\n">

<dispatchMethods; separator="\n">

>>

ListenerDispatchMethod(method) ::= <<
def <if(method.isEnter)>enter<else>exit<endif>Rule(self, listener):
    if hasattr(listener, "<if(method.isEnter)>enter<else>exit<endif><<struct.derivedFromName; format="cap">"):
        listener.<if(method.isEnter)>enter<else>exit<endif><<struct.derivedFromName; format="cap">(self)

>>

VisitorDispatchMethod(method) ::= <<
def accept(self, visitor):
    if hasattr(visitor, "visit<struct.derivedFromName; format="cap">"):
        return visitor.visit<struct.derivedFromName; format="cap">(self)
    else:
        return visitor.visitChildren(self)

>>

AttributeDecl(d) ::= "self.<d.name> = <if(d.initValue)><d.initValue><else>None<endif>"

/** If we don't know location of label def x, use this template */
labelref(x) ::= "<if(!x.isLocal)>localctx.<endif><x.name>"

/** For any action chunk, what is correctly-typed context struct ptr? */
ctx(actionChunk) ::= "localctx"

// used for left-recursive rules
recRuleAltPredicate(ruleName,opPrec) ::= "self.precpred(self._ctx, <opPrec>)"
recRuleSetReturnAction(src,name) ::= "$<name>=<src>.<name>"
recRuleSetStopToken() ::= "self._ctx.stop = self._input.LT(-1)"

recRuleAltStartAction(ruleName, ctxName, label) ::= <<
localctx = <parser.name>.<ctxName>Context(self, _parentctx, _parentState)
<if(label)>localctx.<label> = _prevctx<endif>
self.pushNewRecursionContext(localctx, _startState, self.RULE_<ruleName>)
>>

recRuleLabeledAltStartAction(ruleName, currentAltLabel, label, isListLabel) ::= <<
localctx = <parser.name>.<currentAltLabel; format="cap">Context(self, <parser.name>.<ruleName>

```

```

format="cap">Context(self, _parentctx, _parentState))
<if(label)>
<if(isListLabel)>
localctx.<label>.append(_prevctx)
<else>
localctx.<label> = _prevctx
<endif>
<endif>
self.pushNewRecursionContext(localctx, _startState, self.RULE_<ruleName>)
>>

```

```

recRuleReplaceContext(ctxName) ::= <<
localctx = <parser.name>.<ctxName>Context(self, localctx)
self._ctx = localctx
_prevctx = localctx
>>

```

```

recRuleSetPrevCtx() ::= <<
if self._parseListeners is not None:
    self.triggerExitRuleEvent()
_prevctx = localctx
>>

```

```

LexerFile(lexerFile, lexer, namedActions) ::= <<
<fileHeader(lexerFile.grammarFileName, lexerFile.ANTLRVersion)>
# encoding: utf-8
from __future__ import print_function
from antlr4 import *
from io import StringIO
import sys

<namedActions.header>

<lexer>
>>

```

```

Lexer(lexer, atn, actionFuncs, sempredFuncs, superClass) ::= <<
<if(superClass)>
if __name__ is not None and "." in __name__:
    from .<superClass> import <superClass>
else:
    from <superClass> import <superClass>

<endif>

<atn>

```

```

class <lexer.name>(<if(superClass)><superClass><else>Lexer<endif>):

    atn = ATNDeserializer().deserialize(serializedATN())

    decisionsToDFA = [ DFA(ds, i) for i, ds in enumerate(atn.decisionToState) ]

    <if(lexer.channels)>
        <lexer.channels:{c| <c> = <lexer.channels.(c)>}; separator="\n">

    <endif>
    <if(rest(lexer.modes)>
        <rest(lexer.modes):{m| <m> = <i>}; separator="\n">

    <endif>
    <lexer.tokens:{k | <k> = <lexer.tokens.(k)>}; separator="\n", wrap, anchor>

    channelNames = [ u"DEFAULT_TOKEN_CHANNEL", u"HIDDEN"<if (lexer.channels)>, <lexer.channels:{c|
u"<c>"}; separator=", ", wrap, anchor><endif> ]

    modeNames = [ <lexer.modes:{m| u"<m>"}; separator=", ", wrap, anchor> ]

    literalNames = [ u"\<INVALID>",
        <lexer.literalNames:{t | u<t>}; separator=", ", wrap, anchor> ]

    symbolicNames = [ u"\<INVALID>",
        <lexer.symbolicNames:{t | u<t>}; separator=", ", wrap, anchor> ]

    ruleNames = [ <lexer.ruleNames:{r | u"<r>"}; separator=", ", wrap, anchor> ]

    grammarFileName = u"<lexer.grammarFileName>"

    def __init__(self, input=None, output=sys.stdout):
        super(<lexer.name>, self).__init__(input, output=output)
        self.checkVersion("<lexerFile.ANTLRVersion>")
        self._interp = LexerATNSimulator(self, self.atn, self.decisionsToDFA, PredictionContextCache())
        self._actions = None
        self._predicates = None

    <namedActions.members>

    <dumpActions(lexer, "", actionFuncs, sempredFuncs)>

>>

SerializedATN(model) ::= <<
<! only one segment, can be inlined !>

```

```
def serializedATN():
    with StringIO() as buf:
        buf.write(u"<model.serialized; wrap={ }<\n>    buf.write(u" }>")
        return buf.getvalue()
```

```
>>
```

```
/** Using a type to init value map, try to init a type; if not in table
 * must be an object, default value is "null".
 */
initValue(typeName) ::= <<
<pythonTypeInitMap.(typeName)>
>>
```

```
codeFileExtension() ::= ".py"
```

Found in path(s):

```
* /opt/cola/permits/1473460068_1668481512.5494933/0/antlr4-4-8-1-
jar/org/antlr/v4/tool/templates/codegen/Python2/Python2.stg
```

No license file was found, but licenses were detected in source scan.

```
/*
 * [The "BSD license"]
 * Copyright (c) 2012-2016 Terence Parr
 * Copyright (c) 2012-2016 Sam Harwell
 * Copyright (c) 2014 Eric Vergnaud
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * 3. The name of the author may not be used to endorse or promote products
 * derived from this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
 * IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
 * IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
 * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
 * DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
 * THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
```

```
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

```
*/
```

```
/** ANTLR tool checks output templates are compatible with tool code generation.
```

```
* For now, a simple string match used on x.y of x.y.z scheme.
```

```
* Must match Tool.VERSION during load to templates.
```

```
*
```

```
* REQUIRED.
```

```
*/
```

```
javascriptTypeInitMap ::= [
```

```
  "bool": "false",
```

```
  "int": "0",
```

```
  "float": "0.0",
```

```
  "str": "",
```

```
  default: "{}" // anything other than a primitive type is an object
```

```
]
```

```
// args must be <object-model-object>, <fields-resulting-in-STs>
```

```
ParserFile(file, parser, namedActions, contextSuperClass) ::= <<
```

```
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
```

```
var antlr4 = require('antlr4/index');
```

```
<if(file.genListener)>
```

```
var <file.grammarName>Listener = require('./<file.grammarName>Listener').<file.grammarName>Listener;
```

```
<endif>
```

```
<if(file.genVisitor)>
```

```
var <file.grammarName>Visitor = require('./<file.grammarName>Visitor').<file.grammarName>Visitor;
```

```
<endif>
```

```
<namedActions.header>
```

```
<parser>
```

```
>>
```

```
ListenerFile(file, header, namedActions) ::= <<
```

```
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
```

```
var antlr4 = require('antlr4/index');
```

```
// This class defines a complete listener for a parse tree produced by <file.parserName>.
```

```
function <file.grammarName>Listener() {
```

```
  antlr4.tree.ParseTreeListener.call(this);
```

```
  return this;
```

```
}
```

```
<file.grammarName>Listener.prototype = Object.create(antlr4.tree.ParseTreeListener.prototype);
```

```
<file.grammarName>Listener.prototype.constructor = <file.grammarName>Listener;
```

```

<file.listenerNames:{ lname |
// Enter a parse tree produced by <file.parserName>#<lname>.
<file.grammarName>Listener.prototype.enter<lname; format="cap"> = function(ctx) {
\};

// Exit a parse tree produced by <file.parserName>#<lname>.
<file.grammarName>Listener.prototype.exit<lname; format="cap"> = function(ctx) {
\};

}; separator="\n">

exports.<file.grammarName>Listener = <file.grammarName>Listener;
>>

VisitorFile(file, header, namedActions) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
var antlr4 = require('antlr4/index');

// This class defines a complete generic visitor for a parse tree produced by <file.parserName>.

function <file.grammarName>Visitor() {
  antlr4.tree.ParseTreeVisitor.call(this);
  return this;
}

<file.grammarName>Visitor.prototype = Object.create(antlr4.tree.ParseTreeVisitor.prototype);
<file.grammarName>Visitor.prototype.constructor = <file.grammarName>Visitor;

<file.visitorNames:{ lname |
// Visit a parse tree produced by <file.parserName>#<lname>.
<file.grammarName>Visitor.prototype.visit<lname; format="cap"> = function(ctx) {
  return this.visitChildren(ctx);
\};

}; separator="\n">

exports.<file.grammarName>Visitor = <file.grammarName>Visitor;
>>

fileHeader(grammarFileName, ANTLRVersion) ::= <<
// Generated from <grammarFileName; format="java-escape"> by ANTLR <ANTLRVersion>
// jshint ignore: start
>>

Parser(parser, funcs, atn, sempredFuncs, superClass) ::= <<
<if(superClass)>

```

```

var <superClass> = require('./<superClass>').<superClass>;
<endif>

var grammarFileName = "<parser.grammarFileName; format='java-escape'>";

<atn>

var atn = new antlr4.atn.ATNDeserializer().deserialize(serializedATN);

var decisionsToDFA = atn.decisionToState.map( function(ds, index) { return new antlr4.dfa.DFA(ds, index); });

var sharedContextCache = new antlr4.PredictionContextCache();

var literalNames = [ <parser.literalNames:{t | <t>}; null="null", separator=", ", wrap, anchor > ];

var symbolicNames = [ <parser.symbolicNames:{t | <t>}; null="null", separator=", ", wrap, anchor > ];

var ruleNames = [ <parser.ruleNames:{r | "<r>"}; separator=", ", wrap, anchor > ];

function <parser.name> (input) {
  <superClass; null="antlr4.Parser">.call(this, input);
  this._interp = new antlr4.atn.ParserATNSimulator(this, atn, decisionsToDFA, sharedContextCache);
  this.ruleNames = ruleNames;
  this.literalNames = literalNames;
  this.symbolicNames = symbolicNames;
  <namedActions.members>
  return this;
}

<parser.name>.prototype = Object.create(<superClass; null="antlr4.Parser">.prototype);
<parser.name>.prototype.constructor = <parser.name>;

Object.defineProperty(<parser.name>.prototype, "atn", {
  get : function() {
    return atn;
  }
});

<parser.name>.EOF = antlr4.Token.EOF;
<if(parser.tokens)>
<parser.tokens:{k | <parser.name>.<k> = <parser.tokens.(k)>}; separator="\n", wrap, anchor >
<endif>

<parser.rules:{r | <parser.name>.RULE_<r.name> = <r.index>}; separator="\n", wrap, anchor >

<funcs; separator="\n">

<if(sempredFuncs)>

```

```

<parser.name>.prototype.sempred = function(localctx, ruleIndex, predIndex) {
  switch(ruleIndex) {
    <parser.sempredFuncs.values:{f | case <f.ruleIndex>:
      return this.<f.name>_sempred(localctx, predIndex);}; separator="\n">
    default:
      throw "No predicate with index:" + ruleIndex;
  }
};

```

```

<sempredFuncs.values; separator="\n">
<endif>

```

```

exports.<parser.name> = <parser.name>;

```

```

>>

```

```

dumpActions(recog, argFuncs, actionFuncs, sempredFuncs) ::= <<
<if(actionFuncs)>
<lexer.name>.prototype.action = function(localctx, ruleIndex, actionIndex) {
  switch (ruleIndex) {
    <recog.actionFuncs.values:{f|
  case <f.ruleIndex>:
    this.<f.name>_action(localctx, actionIndex);
    break;}; separator="\n">
    default:
      throw "No registered action for:" + ruleIndex;
  }
};

```

```

<actionFuncs.values; separator="\n">
<endif>
<if(sempredFuncs)>
<lexer.name>.prototype.sempred = function(localctx, ruleIndex, predIndex) {
  switch (ruleIndex) {
    <recog.sempredFuncs.values:{f| case <f.ruleIndex>:
    return this.<f.name>_sempred(localctx, predIndex);}; separator="\n">
    default:
      throw "No registered predicate for:" + ruleIndex;
  }
};

```

```

<sempredFuncs.values; separator="\n">
<endif>
>>

```

```

/* This generates a private method since the actionIndex is generated, making an
* overriding implementation impossible to maintain.

```

```

*/
RuleActionFunction(r, actions) ::= <<

<lexer.name>.prototype.<r.name>_action = function(localctx , actionIndex) {
  switch (actionIndex) {
    <actions:{index|
case <index>:
  <actions.(index)>
  break;}; separator="\n">
  default:
    throw "No registered action for:" + actionIndex;
  }
};
>>

/* This generates a private method since the predIndex is generated, making an
* overriding implementation impossible to maintain.
*/
RuleSempredFunction(r, actions) ::= <<
<if (r.factory.g.lexer)><lexer.name><else><parser.name><endif>.prototype.<r.name>_sempred = function(localctx,
predIndex) {
  switch(predIndex) {
    <actions:{index| case <index>:
return <actions.(index)>;}; separator="\n">
  default:
    throw "No predicate with index:" + predIndex;
  }
};
>>

RuleFunction(currentRule,args,code,locals,ruleCtx,altLabelCtxs,namedActions,finallyAction,postamble,exceptions)
::= <<

<ruleCtx>

<altLabelCtxs:{l | <altLabelCtxs.(l)>}; separator="\n">

<! Define fields of this parser to export the context classes !>
<parser.name>.<currentRule.ctxType> = <currentRule.ctxType>;

<parser.name>.prototype.<currentRule.name> = function(<currentRule.args:{a | <a.name>}; separator=", ">) {

  var localctx = new <currentRule.ctxType>(this, this._ctx, this.state<currentRule.args:{a | , <a.name>}>);
  this.enterRule(localctx, <currentRule.startState>, <parser.name>.RULE_<currentRule.name>);
  <namedActions.init>
  <locals; separator="\n">
  try {

```

```

    <code>
    <postamble; separator="\n">
    <namedActions.after>
<if(exceptions)>
<exceptions; separator="\n">
<else>
} catch (re) {
if(re instanceof antlr4.error.RecognitionException) {
    localctx.exception = re;
    this._errHandler.reportError(this, re);
    this._errHandler.recover(this, re);
} else {
    throw re;
}
}<endif> finally {
    <finallyAction>
    this.exitRule();
}
return localctx;
};
>>

```

```

LeftRecursiveRuleFunction(currentRule,args,code,locals,ruleCtx,altLabelCtxs,
namedActions,finallyAction,postamble) ::=
<<

```

```

<ruleCtx>
<altLabelCtxs:{l | <altLabelCtxs.(l)>}; separator="\n">

<parser.name>.prototype.<currentRule.name> = function(_p<if(currentRule.args)>, <args:{a | , <a>}><endif>) {
if(_p===undefined) {
    _p = 0;
}
var _parentctx = this._ctx;
var _parentState = this.state;
var localctx = new <currentRule.ctxType>(this, this._ctx, _parentState<args:{a | , <a.name>}>);
var _prevctx = localctx;
var _startState = <currentRule.startState>;
this.enterRecursionRule(localctx, <currentRule.startState>, <parser.name>.RULE_<currentRule.name>, _p);
<namedActions.init>
<locals; separator="\n">
try {
    <code>
    <postamble; separator="\n">
    <namedActions.after>
} catch( error) {
    if(error instanceof antlr4.error.RecognitionException) {

```

```

        localctx.exception = error;
        this._errHandler.reportError(this, error);
        this._errHandler.recover(this, error);
    } else {
        throw error;
    }
} finally {
    <finallyAction>
    this.unrollRecursionContexts(_parentctx)
}
return localctx;
};

>>

```

```

CodeBlockForOuterMostAlt(currentOuterMostAltCodeBlock, locals, preamble, ops) ::= <<
<if(currentOuterMostAltCodeBlock.altLabel)>localctx = new <currentOuterMostAltCodeBlock.altLabel;
format="cap">Context(this, localctx);<endif>
this.enterOuterAlt(localctx, <currentOuterMostAltCodeBlock.alt.altNum>);
<CodeBlockForAlt(currentAltCodeBlock=currentOuterMostAltCodeBlock, ...)>
>>

```

```

CodeBlockForAlt(currentAltCodeBlock, locals, preamble, ops) ::= <<
<locals; separator="\n">
<preamble; separator="\n">
<ops; separator="\n">
>>

```

```

LL1AltBlock(choice, preamble, alts, error) ::= <<
this.state = <choice.stateNumber>;
this._errHandler.sync(this);
<if(choice.label)><labelref(choice.label)> = this._input.LT(1);<endif>
<preamble; separator="\n">
switch(this._input.LA(1)) {
<choice.altLook,alts:{look,alt| <cases(ttypes=look)>
    <alt>
    break;}; separator="\n">
default:
    <error>
}
>>

```

```

LL1OptionalBlock(choice, alts, error) ::= <<
this.state = <choice.stateNumber>;
this._errHandler.sync(this);
switch (this._input.LA(1)) {
<choice.altLook,alts:{look,alt| <cases(ttypes=look)>
    <alt>

```

```

break;}; separator="\n">
default:
break;
}
>>

```

```

LL1OptionalBlockSingleAlt(choice, expr, alts, preamble, error, followExpr) ::= <<
this.state = <choice.stateNumber>;
this._errHandler.sync(this);
<preamble; separator="\n">
if(<expr>) {
    <alts; separator="\n">
}
<!else if ( !(<followExpr> ) <error>!>
>>

```

```

LL1StarBlockSingleAlt(choice, loopExpr, alts, preamble, iteration) ::= <<
this.state = <choice.stateNumber>;
this._errHandler.sync(this);
<preamble; separator="\n">
while(<loopExpr>) {
    <alts; separator="\n">
    this.state = <choice.loopBackStateNumber>;
    this._errHandler.sync(this);
    <iteration>
}
>>

```

```

LL1PlusBlockSingleAlt(choice, loopExpr, alts, preamble, iteration) ::= <<
this.state = <choice.blockStartStateNumber>; <! alt block decision !>
this._errHandler.sync(this);
<preamble; separator="\n">
do {
    <alts; separator="\n">
    this.state = <choice.stateNumber>; <! loopback/exit decision !>
    this._errHandler.sync(this);
    <iteration>
} while(<loopExpr>);
>>

```

```

// LL(*) stuff

```

```

AltBlock(choice, preamble, alts, error) ::= <<
this.state = <choice.stateNumber>;
this._errHandler.sync(this);
<if(choice.label)><labelref(choice.label)> = _input.LT(1)<endif>
<preamble; separator="\n">
var la_ = this._interp.adaptivePredict(this._input,<choice.decision>,this._ctx);

```

```

switch(la_) {
<alts:{alt |
case <i>:
  <alt>
  break;
}; separator="\n">
}
>>

```

```

OptionalBlock(choice, alts, error) ::= <<
this.state = <choice.stateNumber>;
this._errHandler.sync(this);
var la_ = this._interp.adaptivePredict(this._input,<choice.decision>,this._ctx);
<alts:{alt |
if(la_===<i><if(!choice.ast.greedy)>+1<endif>) {
  <alt>
}; separator="\n} else ">
}
>>

```

```

StarBlock(choice, alts, sync, iteration) ::= <<
this.state = <choice.stateNumber>;
this._errHandler.sync(this);
var _alt = this._interp.adaptivePredict(this._input,<choice.decision>,this._ctx)
while(_alt!=<choice.exitAlt> && _alt!=antlr4.atn.ATN.INVALID_ALT_NUMBER) {
  if(_alt===1<if(!choice.ast.greedy)>+1<endif>) {
    <iteration>
    <alts> <! should only be one !>
  }
  this.state = <choice.loopBackStateNumber>;
  this._errHandler.sync(this);
  _alt = this._interp.adaptivePredict(this._input,<choice.decision>,this._ctx);
}
>>

```

```

PlusBlock(choice, alts, error) ::= <<
this.state = <choice.blockStartStateNumber>; <! alt block decision !>
this._errHandler.sync(this);
var _alt = 1<if(!choice.ast.greedy)>+1<endif>;
do {
  switch (_alt) {
  <alts:{alt|
case <i><if(!choice.ast.greedy)>+1<endif>:
  <alt>
  break;}; separator="\n">
  default:
  <error>

```

```

}
this.state = <choice.loopBackStateNumber>; <! loopback/exit decision !>
this._errHandler.sync(this);
_alt = this._interp.adaptivePredict(this._input,<choice.decision>, this._ctx);
} while ( _alt!=<choice.exitAlt> && _alt!=antlr4.atn.ATN.INVALID_ALT_NUMBER );
>>

Sync(s) ::= "sync(<s.expecting.name>)"

ThrowNoViableAlt(t) ::= "throw new antlr4.error.NoViableAltException(this);"

TestSetInline(s) ::= <<
<s.bitsets:{bits | <if(rest(rest(bits.ttypes)))><bitsetBitfieldComparison(s, bits)><else><bitsetInlineComparison(s,
bits)><endif>}; separator=" || ">
>>

// Javascript language spec - shift operators are 32 bits long max
testShiftInRange(shiftAmount) ::= <<
((<shiftAmount>) & ~0x1f) == 0
>>

// produces smaller bytecode only when bits.ttypes contains more than two items
bitsetBitfieldComparison(s, bits) ::= <%
(<testShiftInRange({<offsetShiftVar(s.varName, bits.shift)>}) && ((1 \<< <offsetShiftVar(s.varName,
bits.shift)>) & (<bits.ttypes:{ttype | (1 \<< <offsetShiftType(ttype, bits.shift)>)}); separator=" | ">)) != 0
%>

isZero ::= [
"0":true,
default:false
]

offsetShiftVar(shiftAmount, offset) ::= <%
<if(!isZero.(offset))><shiftAmount> - <offset><else><shiftAmount><endif>
%>

offsetShiftType(shiftAmount, offset) ::= <%
<if(!isZero.(offset))><parser.name>.<shiftAmount> - <offset><else><parser.name>.<shiftAmount><endif>
%>

// produces more efficient bytecode when bits.ttypes contains at most two items
bitsetInlineComparison(s, bits) ::= <%
<bits.ttypes:{ttype | <s.varName>===<parser.name>.<ttype>}; separator=" || ">
%>

cases(ttypes) ::= <<
<ttypes:{t | case <parser.name>.<t>:}; separator="\n">
>>

```

```

InvokeRule(r, argExprsChunks) ::= <<
this.state = <r.stateNumber>;
<if(r.labels)><r.labels:{1 | <labelref(1)> =
}><endif>this.<r.name><(if(r.ast.options.p)><r.ast.options.p><if(argExprsChunks)><endif><endif><argExprsChu
nks>);
>>

```

```

MatchToken(m) ::= <<
this.state = <m.stateNumber>;
<if(m.labels)><m.labels:{1 | <labelref(1)> = }><endif>this.match(<parser.name>.<m.name>);
>>

```

```

MatchSet(m, expr, capture) ::= "<CommonSetStuff(m, expr, capture, false)>"

```

```

MatchNotSet(m, expr, capture) ::= "<CommonSetStuff(m, expr, capture, true)>"

```

```

CommonSetStuff(m, expr, capture, invert) ::= <<
this.state = <m.stateNumber>;
<if(m.labels)><m.labels:{1 | <labelref(1)> = }>this._input.LT(1);<endif>
<capture>
<if(invert)>if(<m.varName>\<=0 || <expr><else>if(!(<expr>))<endif> {
  <if(m.labels)><m.labels:{1 | <labelref(1)> = }><endif>this._errHandler.recoverInline(this);
}
else {
  this._errHandler.reportMatch(this);
  this.consume();
}
>>

```

```

Wildcard(w) ::= <<
this.state = <w.stateNumber>;
<if(w.labels)><w.labels:{1 | <labelref(1)> = }><endif>this.matchWildcard();
>>

```

```

// ACTION STUFF

```

```

Action(a, foo, chunks) ::= "<chunks>"

```

```

ArgAction(a, chunks) ::= "<chunks>"

```

```

SemPred(p, chunks, failChunks) ::= <<
this.state = <p.stateNumber>;
if (!( <chunks>)) {
  throw new antlr4.error.FailedPredicateException(this, <p.predicate><if(failChunks)>,
<failChunks><elseif(p.msg)>, <p.msg><endif>);
}
>>

```

```

ExceptionClause(e, catchArg, catchAction) ::= <<
catch (<catchArg>) {
  <catchAction>
}
>>

// lexer actions are not associated with model objects

LexerSkipCommand() ::= "this.skip()"
LexerMoreCommand() ::= "this.more()"
LexerPopModeCommand() ::= "this.popMode()"
LexerTypeCommand(arg, grammar) ::= "this._type = <arg>"
LexerChannelCommand(arg, grammar) ::= "this._channel = <arg>"
LexerModeCommand(arg, grammar) ::= "this._mode = <arg>"
LexerPushModeCommand(arg, grammar) ::= "this.pushMode(<arg>)"

ActionText(t) ::= "<t.text>"
ActionTemplate(t) ::= "<t.st>"
ArgRef(a) ::= "localctx.<a.name>"
LocalRef(a) ::= "localctx.<a.name>"
RetValRef(a) ::= "localctx.<a.name>"
QRetValRef(a) ::= "<ctx(a)>.<a.dict>.<a.name>"
/** How to translate $tokenLabel */
TokenRef(t) ::= "<ctx(t)>.<t.name>"
LabelRef(t) ::= "<ctx(t)>.<t.name>"
ListLabelRef(t) ::= "<ctx(t)>.<ListLabelName(t.name)>"
SetAttr(s,rhsChunks) ::= "<ctx(s)>.<s.name> = <rhsChunks>"

TokenLabelType() ::= "<file.TokenLabelType; null={Token}>"
InputSymbolType() ::= "<file.InputSymbolType; null={Token}>"

TokenPropertyRef_text(t) ::= "<ctx(t)>.<t.label>===null ? null : <ctx(t)>.<t.label>.text)"
TokenPropertyRef_type(t) ::= "<ctx(t)>.<t.label> === null ? 0 : <ctx(t)>.<t.label>.type)"
TokenPropertyRef_line(t) ::= "<ctx(t)>.<t.label> === null ? 0 : <ctx(t)>.<t.label>.line)"
TokenPropertyRef_pos(t) ::= "<ctx(t)>.<t.label> === null ? 0 : <ctx(t)>.<t.label>.column)"
TokenPropertyRef_channel(t) ::= "<ctx(t)>.<t.label> === null ? 0 : <ctx(t)>.<t.label>.channel)"
TokenPropertyRef_index(t) ::= "<ctx(t)>.<t.label> === null ? 0 : <ctx(t)>.<t.label>.tokenIndex)"
TokenPropertyRef_int(t) ::= "<ctx(t)>.<t.label> === null ? 0 : parseInt(<ctx(t)>.<t.label>.text)"

RulePropertyRef_start(r) ::= "<ctx(r)>.<r.label>===null ? null : <ctx(r)>.<r.label>.start)"
RulePropertyRef_stop(r) ::= "<ctx(r)>.<r.label>===null ? null : <ctx(r)>.<r.label>.stop)"
RulePropertyRef_text(r) ::= "<ctx(r)>.<r.label>===null ? null : this._input.getText(new
antlr4.Interval(<ctx(r)>.<r.label>.start,<ctx(r)>.<r.label>.stop)))"
RulePropertyRef_ctx(r) ::= "<ctx(r)>.<r.label>"
RulePropertyRef_parser(r) ::= "this"

ThisRulePropertyRef_start(r) ::= "localctx.start"

```

```

ThisRulePropertyRef_stop(r) ::= "localctx.stop"
ThisRulePropertyRef_text(r) ::= "this._input.getText(new antlr4.Interval(localctx.start, this._input.LT(-1)))"
ThisRulePropertyRef_ctx(r) ::= "localctx"
ThisRulePropertyRef_parser(r) ::= "this"

```

```

NonLocalAttrRef(s) ::= "getInvokingContext(<s.ruleIndex>).<s.name>"
SetNonLocalAttr(s, rhsChunks) ::= "getInvokingContext(<s.ruleIndex>).<s.name> = <rhsChunks>"

```

```

AddToLabelList(a) ::= "<ctx(a.label)>.<a.listName>.push(<labelref(a.label)>);"

```

```

TokenDecl(t) ::= "this.<t.name> = null; // <TokenLabelType()>"
TokenTypeDecl(t) ::= "var <t.name> = 0; // <TokenLabelType()> type"
TokenListDecl(t) ::= "this.<t.name> = []; // of <TokenLabelType()>s"
RuleContextDecl(r) ::= "this.<r.name> = null; // <r.ctxName>"
RuleContextListDecl(rdecl) ::= "this.<rdecl.name> = []; // of <rdecl.ctxName>s"

```

```

ContextTokenGetterDecl(t) ::= <<
<t.name> = function() {
    return this.getToken(<parser.name>.<t.name>, 0);
};
>>

```

```

// should never be called
ContextTokenListGetterDecl(t) ::= <<
def <t.name>_list(self):
    return self.getTokens(<parser.name>.<t.name>)
>>

```

```

ContextTokenListIndexedGetterDecl(t) ::= <<
<t.name> = function(i) {
    if(i===undefined) {
        i = null;
    }
    if(i===null) {
        return this.getTokens(<parser.name>.<t.name>);
    } else {
        return this.getToken(<parser.name>.<t.name>, i);
    }
};
>>

```

```

ContextRuleGetterDecl(r) ::= <<
<r.name> = function() {
    return this.getTypedRuleContext(<r.ctxName>,0);
};
>>

```

```

// should never be called
ContextRuleListGetterDecl(r) ::= <<
def <r.name>_list(self):
    return self.getTypedRuleContexts(<parser.name>.<r.ctxName>)

>>

ContextRuleListIndexedGetterDecl(r) ::= <<
<r.name> = function(i) {
    if(i===undefined) {
        i = null;
    }
    if(i===null) {
        return this.getTypedRuleContexts(<r.ctxName>);
    } else {
        return this.getTypedRuleContext(<r.ctxName>,i);
    }
};
>>

LexerRuleContext() ::= "RuleContext"

/** The rule context name is the rule followed by a suffix; e.g.,
 * r becomes rContext.
 */
RuleContextNameSuffix() ::= "Context"

ImplicitTokenLabel(tokenName) ::= "_<tokenName>"
ImplicitRuleLabel(ruleName) ::= "_<ruleName>"
ImplicitSetLabel(id) ::= "_tset<id>"
ListLabelName(label) ::= "<label>"

CaptureNextToken(d) ::= "<d.varName> = self._input.LT(1)"
CaptureNextTokenType(d) ::= "<d.varName> = this._input.LA(1);"

StructDecl(struct,ctorAttrs,attrs, getters,dispatchMethods,interfaces,extensionMembers) ::= <<
function <struct.name>(parser, parent, invokingState<struct.ctorAttrs:{a | , <a.name>}> ) {
    if(parent===undefined) {
        parent = null;
    }
    if(invokingState===undefined || invokingState===null) {
        invokingState = -1;
    }
    <if(contextSuperClass><contextSuperClass><else>antlr4.ParserRuleContext<endif>.call(this, parent,
invokingState);
    this.parser = parser;
    this.ruleIndex = <parser.name>.RULE_<struct.derivedFromName>;
    <attrs:{a | <a>} separator="\n">

```

```

    <struct.ctorAttrs:{ a | this.<a.name> = <a.name> || null; }; separator="\n">
    return this;
}

<struct.name>.prototype =
Object.create(<if(contextSuperClass)><contextSuperClass><else>antlr4.ParserRuleContext<endif>.prototype);
<struct.name>.prototype.constructor = <struct.name>;

<getters:{ g | <struct.name>.prototype.<g> }; separator="\n\n">

<if(struct.provideCopyFrom)> <! don't need copy unless we have subclasses !>
<struct.name>.prototype.copyFrom = function(ctx) {
<if(contextSuperClass)><contextSuperClass><else>antlr4.ParserRuleContext<endif>.prototype.copyFrom.call(this,
ctx);
    <struct.attrs:{ a | this.<a.name> = ctx.<a.name>; }; separator="\n">
};
<endif>
<dispatchMethods; separator="\n">
<extensionMembers; separator="\n">

>>

AltLabelStructDecl(struct,attrs,getters,dispatchMethods) ::= <<
function <struct.name>(parser, ctx) {
    <currentRule.name; format="cap">Context.call(this, parser);
    <attrs:{ a | <a>; }; separator="\n">
    <currentRule.name; format="cap">Context.prototype.copyFrom.call(this, ctx);
    return this;
}

<struct.name>.prototype = Object.create(<currentRule.name; format="cap">Context.prototype);
<struct.name>.prototype.constructor = <struct.name>;

<! Define fields of this parser to export this struct/context class !>
<parser.name>.<struct.name> = <struct.name>;

<getters:{ g | <struct.name>.prototype.<g> }; separator="\n\n">
<dispatchMethods; separator="\n">

>>

ListenerDispatchMethod(method) ::= <<
<struct.name>.prototype.<if(method.isEnter)>enter<else>exit<endif>Rule = function(listener) {
    if(listener instanceof <parser.grammarName>Listener ) {
        listener.<if(method.isEnter)>enter<else>exit<endif><struct.derivedFromName; format="cap">(this);
    }
};

```

>>

```
VisitorDispatchMethod(method) ::= <<
<struct.name>.prototype.accept = function(visitor) {
  if ( visitor instanceof <parser.grammarName>Visitor ) {
    return visitor.visit<struct.derivedFromName; format="cap">(this);
  } else {
    return visitor.visitChildren(this);
  }
};
```

>>

```
AttributeDecl(d) ::= "this.<d.name> = <if(d.initValue)><d.initValue><else>null<endif>"
```

```
/** If we don't know location of label def x, use this template */
```

```
labelref(x) ::= "<if(!x.isLocal)>localctx.<endif><x.name>"
```

```
/** For any action chunk, what is correctly-typed context struct ptr? */
```

```
ctx(actionChunk) ::= "localctx"
```

```
// used for left-recursive rules
```

```
recRuleAltPredicate(ruleName,opPrec) ::= "this.precpred(this._ctx, <opPrec>)"
```

```
recRuleSetReturnAction(src,name) ::= "$<name>=$<src>.<name>"
```

```
recRuleSetStopToken() ::= "this._ctx.stop = this._input.LT(-1);"
```

```
recRuleAltStartAction(ruleName, ctxName, label) ::= <<
```

```
localctx = new <ctxName>Context(this, _parentctx, _parentState);
```

```
<if(label)>localctx.<label> = _prevctx;<endif>
```

```
this.pushNewRecursionContext(localctx, _startState, <parser.name>.RULE_<ruleName>);
```

```
>>
```

```
recRuleLabeledAltStartAction(ruleName, currentAltLabel, label, isListLabel) ::= <<
```

```
localctx = new <currentAltLabel; format="cap">Context(this, new <ruleName; format="cap">Context(this,
_parentctx, _parentState));
```

```
<if(label)>
```

```
<if(isListLabel)>
```

```
localctx.<label>.push(_prevctx);
```

```
<else>
```

```
localctx.<label> = _prevctx;
```

```
<endif>
```

```
<endif>
```

```
this.pushNewRecursionContext(localctx, _startState, <parser.name>.RULE_<ruleName>);
```

```
>>
```

```
recRuleReplaceContext(ctxName) ::= <<
```

```
localctx = new <ctxName>Context(this, localctx);
```

```
this._ctx = localctx;
```

```
_prevctx = localctx;
```

```
>>
```

```
recRuleSetPrevCtx() ::= <<
```

```
if(this._parseListeners!==null) {
```

```
    this.triggerExitRuleEvent();
```

```
}
```

```
_prevctx = localctx;
```

```
>>
```

```
LexerFile(lexerFile, lexer, namedActions) ::= <<
```

```
<fileHeader(lexerFile.grammarFileName, lexerFile.ANTLRVersion)>
```

```
var antlr4 = require('antlr4/index');
```

```
<namedActions.header>
```

```
<lexer>
```

```
>>
```

```
Lexer(lexer, atn, actionFuncs, sempredFuncs, superClass) ::= <<
```

```
<if(superClass)>
```

```
var <superClass> = require('./<superClass>').<superClass>;
```

```
<endif>
```

```
<atn>
```

```
var atn = new antlr4.atn.ATNDeserializer().deserialize(serializedATN);
```

```
var decisionsToDFA = atn.decisionToState.map( function(ds, index) { return new antlr4.dfa.DFA(ds, index); });
```

```
function <lexer.name>(input) {
```

```
    <if(superClass)><superClass><else>antlr4.Lexer<endif>.call(this, input);
```

```
    this._interp = new antlr4.atn.LexerATNSimulator(this, atn, decisionsToDFA, new  
    antlr4.PredictionContextCache());
```

```
    <namedActions.members>
```

```
    return this;
```

```
}
```

```
<lexer.name>.prototype = Object.create(<if(superClass)><superClass><else>antlr4.Lexer<endif>.prototype);
```

```
<lexer.name>.prototype.constructor = <lexer.name>;
```

```
Object.defineProperty(<lexer.name>.prototype, "atn", {
```

```
    get : function() {
```

```
        return atn;
```

```
    }
```

```
});
```

```

<lexer.name>.EOF = antlr4.Token.EOF;
<lexer.tokens:{k | <lexer.name>.<k> = <lexer.tokens.(k)>; separator="\n", wrap, anchor>

<if(lexer.channels)>
<lexer.channels:{c | <lexer.name>.<c> = <lexer.channels.(c)>; separator="\n">

<endif>
<if(rest(lexer.modes))>
<rest(lexer.modes):{m | <lexer.name>.<m> = <i>; separator="\n">

<endif>
<lexer.name>.prototype.channelNames = [ "DEFAULT_TOKEN_CHANNEL", "HIDDEN"<if (lexer.channels)>,
<lexer.channels:{c | "<c>"}; separator=", ", wrap, anchor><endif> ];

<lexer.name>.prototype.modeNames = [ <lexer.modes:{m | "<m>"}; separator=", ", wrap, anchor> ];

<lexer.name>.prototype.literalNames = [ <lexer.literalNames:{t | <t>}; null="null", separator=", ", wrap, anchor> ];

<lexer.name>.prototype.symbolicNames = [ <lexer.symbolicNames:{t | <t>}; null="null", separator=", ", wrap,
anchor> ];

<lexer.name>.prototype.ruleNames = [ <lexer.ruleNames:{r | "<r>"}; separator=", ", wrap, anchor> ];

<lexer.name>.prototype.grammarFileName = "<lexer.grammarFileName>";

<dumpActions(lexer, "", actionFuncs, sempredFuncs)>

exports.<lexer.name> = <lexer.name>;

>>

SerializedATN(model) ::= <<
<! only one segment, can be inlined !>

var serializedATN = ["<model.serialized; wrap={",<\n> " }>"].join("");

>>

/** Using a type to init value map, try to init a type; if not in table
* must be an object, default value is "null".
*/
initValue(typeName) ::= <<
<javacriptTypeInitMap.(typeName)>
>>

codeFileExtension() ::= ".js"

```

Found in path(s):

```
* /opt/cola/permits/1473460068_1668481512.5494933/0/antlr4-4-8-1-  
jar/org/antlr/v4/tool/templates/codegen/JavaScript/JavaScript.stg  
No license file was found, but licenses were detected in source scan.
```

```
/*  
* [The "BSD license"]  
* Copyright (c) 2016, Mike Lischke  
* All rights reserved.  
*  
* Redistribution and use in source and binary forms, with or without  
* modification, are permitted provided that the following conditions  
* are met:  
*  
* 1. Redistributions of source code must retain the above copyright  
* notice, this list of conditions and the following disclaimer.  
* 2. Redistributions in binary form must reproduce the above copyright  
* notice, this list of conditions and the following disclaimer in the  
* documentation and/or other materials provided with the distribution.  
* 3. The name of the author may not be used to endorse or promote products  
* derived from this software without specific prior written permission.  
*  
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR  
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES  
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.  
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,  
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT  
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,  
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY  
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT  
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF  
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.  
*/
```

Found in path(s):

```
* /opt/cola/permits/1473460068_1668481512.5494933/0/antlr4-4-8-1-  
jar/org/antlr/v4/tool/templates/codegen/Cpp/Files.stg  
No license file was found, but licenses were detected in source scan.
```

```
/*  
* [The "BSD license"]  
* Copyright (c) 2012-2016 Terence Parr  
* Copyright (c) 2012-2016 Sam Harwell  
* All rights reserved.  
*  
* Redistribution and use in source and binary forms, with or without  
* modification, are permitted provided that the following conditions  
* are met:
```

*
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * 3. The name of the author may not be used to endorse or promote products
 * derived from this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
 * IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
 * IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
 * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
 * DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
 * THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
 * THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */

```
atn(startState, states, edges, rankdir, decisionRanks, useBox) ::= <<
digraph ATN {
rankdir=LR;
<decisionRanks; separator="\n">
<states; separator="\n">
<edges; separator="\n">
}
>>
```

```
dfa(name, startState, states, edges, rankdir, decisionRanks, useBox) ::= <<
digraph <name> {
<if(rankdir)>rankdir=<rankdir>;<endif>
<decisionRanks; separator="\n">
<states; separator="\n">
<edges; separator="\n">
}
>>
```

```
decision-rank(states) ::= <<
{rank=same; rankdir=TB; <states:{s | s<s>}; separator=" "; ">}
>>
```

```
edge(src,target,label,arrowhead,transitionIndex) ::= <<
<src><if(transitionIndex)>p<transitionIndex><endif> -> <target> [fontsize=11, fontname="Courier", arrowsize=.7,
label = "<label>"<if(arrowhead)>, arrowhead = <arrowhead><endif>];
>>
```

```

action-edge(src,target,label,arrowhead,transitionIndex) ::= <<
<src><if(transitionIndex)>p<transitionIndex><endif> -> <target> [fontsize=11, fontname="Courier", arrowsize=.7,
label = "<label>"<if(arrowhead)>, arrowhead = <arrowhead><endif>];
>>

```

```

epsilon-edge(src,label,target,arrowhead,transitionIndex,loopback=false) ::= <<
<src><if(transitionIndex)>p<transitionIndex><endif> -> <target> [fontname="Times-Italic",
label="&epsilon;"<if(loopback)>, style="dashed"<endif>];
>>

```

```

state(state, label, name, transitions) ::= <%
<name>[fontsize=11,
label="
<! rest(transition) tests for decision states: these nodes have a non-empty set of transitions after the first one. !>
<if(rest(transitions))>
{
<! Label on the left side of the record node. !>
<label>
|
<! Named ports in order on right side of record node, no display text. !>
{<transitions:{t|\<p>i0>>}; separator="|">}
<else>
<label>
<endif>
"
<if(rest(transitions))>
, shape=record, fixedsize=false
<else>
, shape=circle, fixedsize=true, width=.55
<endif>
, peripheries=1];
%>

```

```

stopstate(name,label,actionIndex,useBox) ::= <<
<name>[fontsize=11, label="<label><if(actionIndex)>,\naction:<actionIndex><endif>",
<if(useBox)>shape=polygon,sides=4,peripheries=2,fixedsize=false<else>shape=doublecircle, fixedsize=true,
width=.6<endif>];
>>

```

Found in path(s):

```

* /opt/cola/permits/1473460068_1668481512.5494933/0/antlr4-4-8-1-jar/org/antlr/v4/tool/templates/dot/graphs.stg
No license file was found, but licenses were detected in source scan.

```

```

/*
* [The "BSD license"]
* Copyright (c) 2012-2016 Terence Parr
* Copyright (c) 2012-2016 Sam Harwell
* Copyright (c) 2015 Janyou

```

* All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * 3. The name of the author may not be used to endorse or promote products
 * derived from this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
 * IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
 * IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
 * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
 * DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
 * THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
 * THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */

```
SwiftTypeInitMap ::= [
  "Int": "0",
  "Int64": "0",
  "Float": "0.0",
  "Double": "0.0",
  "Bool": "false",
  default: "nil" // anything other than a primitive type is an object
]
SwiftTypeMap ::= [
  "int": "Int",
  "float": "Float",
  "long": "Int64",
  "double": "Double",
  "bool": "Bool",
  "boolean": "Bool",
  default : key
]
// args must be <object-model-object>, <fields-resulting-in-STs>

accessLevelOpenOK(obj) ::= "<obj.accessLevel; null=\\"open\\">"
accessLevelNotOpen(obj) ::= "<obj.accessLevel; null=\\"public\\">"
```

```

ParserFile(file, parser, namedActions, contextSuperClass) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
<if(file.genPackage)>
<!package <file.genPackage>;!>
<endif>
<namedActions.header>
import Antlr4

<parser>
>>

```

```

ListenerFile(file, header, namedActions) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
<if(file.genPackage)>
<!package <file.genPackage>;!>
<endif>
<header>
import Antlr4

```

```

/**
 * This interface defines a complete listener for a parse tree produced by
 * { @link <file.parserName> }.
 */
<accessLevelNotOpen(file)> protocol <file.grammarName>Listener: ParseTreeListener {
  <file.listenerNames>: {Iname |
/**
<if(file.listenerLabelRuleNames.(Iname))>
 * Enter a parse tree produced by the { @code <Iname>\}
 * labeled alternative in { @link <file.parserName>#<file.listenerLabelRuleNames.(Iname)>\}.
<else>
 * Enter a parse tree produced by { @link <file.parserName>#<Iname>\}.
<endif>
- Parameters:
  - ctx: the parse tree
*/
func enter<Iname; format="cap">(_ ctx: <file.parserName>.<Iname; format="cap">Context)
/**
<if(file.listenerLabelRuleNames.(Iname))>
 * Exit a parse tree produced by the { @code <Iname>\}
 * labeled alternative in { @link <file.parserName>#<file.listenerLabelRuleNames.(Iname)>\}.
<else>
 * Exit a parse tree produced by { @link <file.parserName>#<Iname>\}.
<endif>
- Parameters:
  - ctx: the parse tree
*/
func exit<Iname; format="cap">(_ ctx: <file.parserName>.<Iname; format="cap">Context); separator="\n">
}

```

>>

```
BaseListenerFile(file, header, namedActions) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
<if(file.genPackage)>
<!package <file.genPackage>;!>
<endif>
<header>

import Antlr4

/**
 * This class provides an empty implementation of { @link <file.grammarName>Listener },
 * which can be extended to create a listener which only needs to handle a subset
 * of the available methods.
 */
<accessLevelOpenOK(file)> class <file.grammarName>BaseListener: <file.grammarName>Listener {
    <accessLevelNotOpen(file)> init() { \}
    <file.listenerNames:{lname |

/**
 * { @inheritDoc\}
 *
 * \<p>The default implementation does nothing.\</p>
 */
<accessLevelOpenOK(file)> func enter<lname; format="cap">(_ ctx: <file.parserName>.<lname;
format="cap">Context) { \}

/**
 * { @inheritDoc\}
 *
 * \<p>The default implementation does nothing.\</p>
 */
<accessLevelOpenOK(file)> func exit<lname; format="cap">(_ ctx: <file.parserName>.<lname;
format="cap">Context) { \} }; separator="\n">

/**
 * { @inheritDoc\}
 *
 * \<p>The default implementation does nothing.\</p>
 */
<accessLevelOpenOK(file)> func enterEveryRule(_ ctx: ParserRuleContext) { }

/**
 * { @inheritDoc\}
 *
 * \<p>The default implementation does nothing.\</p>
 */
<accessLevelOpenOK(file)> func exitEveryRule(_ ctx: ParserRuleContext) { }
```

```

/**
 * {@inheritDoc\}
 *
 * \<p>The default implementation does nothing.\</p>
 */
<accessLevelOpenOK(file)> func visitTerminal(_ node: TerminalNode) { }
/**
 * {@inheritDoc\}
 *
 * \<p>The default implementation does nothing.\</p>
 */
<accessLevelOpenOK(file)> func visitErrorNode(_ node: ErrorNode) { }
}
>>

VisitorFile(file, header, namedActions) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
<if(file.genPackage)>
<!package <file.genPackage>;!>
<endif>
<header>
import Antlr4

/**
 * This interface defines a complete generic visitor for a parse tree produced
 * by {@link <file.parserName>}.
 *
 * @param \<T> The return type of the visit operation. Use {@link Void} for
 * operations with no return type.
 */
<accessLevelOpenOK(file)> class <file.grammarName>Visitor\<T>: ParseTreeVisitor\<T> {
  <file.visitorNames>{ Iname |
/**
<if(file.visitorLabelRuleNames.(Iname))>
 * Visit a parse tree produced by the {@code <Iname>}
 * labeled alternative in {@link <file.parserName>#<file.visitorLabelRuleNames.(Iname)>}.
<else>
 * Visit a parse tree produced by {@link <file.parserName>#<Iname>}.
<endif>
- Parameters:
- ctx: the parse tree
- returns: the visitor result
*/
<accessLevelOpenOK(file)> func visit<Iname>(format="cap")(_ ctx: <file.parserName>.<Iname>;
format="cap">Context) -> T {
  fatalError("#function + " must be overridden")
}
}; separator="\n">

```

```
}  
>>
```

```
BaseVisitorFile(file, header, namedActions) ::= <<  
<fileHeader(file.grammarFileName, file.ANTLRVersion)>  
<if(file.genPackage)>  
<!package <file.genPackage>;!>  
<endif>  
<header>  
import Antlr4  
  
/**  
 * This class provides an empty implementation of { @link <file.grammarName>Visitor},  
 * which can be extended to create a visitor which only needs to handle a subset  
 * of the available methods.  
 *  
 * @param \<T> The return type of the visit operation. Use { @link Void} for  
 * operations with no return type.  
 */  
<accessLevelOpenOK(file)> class <file.grammarName>BaseVisitor\<T>: AbstractParseTreeVisitor\<T> {  
  <file.visitorNames:{lname |  
  /**  
  * { @inheritDoc\}  
  *  
  * \<p>The default implementation returns the result of calling  
  * { @link #visitChildren\} on { @code ctx\}.\</p>  
  */  
  <accessLevelOpenOK(file)> func visit<lname; format="cap">(_ ctx: <file.parserName>.<lname;  
  format="cap">Context) -> T? { return visitChildren(ctx) \}}; separator="\n">  
  }  
>>
```

```
fileHeader(grammarFileName, ANTLRVersion) ::= <<  
// Generated from <grammarFileName; format="java-escape"> by ANTLR <ANTLRVersion>  
>>
```

```
Parser(parser, funcs, atn, sempredFuncs, superClass) ::= <<  
<Parser_(ctor="parser_ctor", ...)>  
>>
```

```
Parser_(parser, funcs, atn, sempredFuncs, ctor, superClass) ::= <<  
</*!@SuppressWarnings({"all", "warnings", "unchecked", "unused", "cast"})!>  
<accessLevelOpenOK(parser)> class <parser.name>: <superClass; null="Parser"> {
```

```
  internal static var _decisionToDFA: [DFA] = {  
    var decisionToDFA = [DFA]()  
    let length = <parser.name>._ATN.getNumberOfDecisions()  
    for i in 0..\<length {
```

```

    <!-- decisionToDFA[i] = DFA(<parser.name>._ATN.getDecisionState(i)!, i);!>
    decisionToDFA.append(DFA(<parser.name>._ATN.getDecisionState(i)!, i))
  }
  return decisionToDFA
}()

```

```

internal static let _sharedContextCache = PredictionContextCache()

```

```

<if(parser.tokens)>
<accessLevelNotOpen(parser)>
enum Tokens: Int {
  case EOF = -1, <parser.tokens:{k | <k> = <parser.tokens.(k)>}; separator=", ", wrap, anchor>
}
<endif>

```

```

<accessLevelNotOpen(parser)>
static let <parser.rules:{r | RULE_<r.name> = <r.index>}; separator=", ", wrap, anchor>

```

```

<accessLevelNotOpen(parser)>
static let ruleNames: [String] = [
  <parser.ruleNames:{r | "<r>"}; separator=", ", wrap, anchor>
]

```

```

<vocabulary(parser.literalNames, parser.symbolicNames,
  accessLevelNotOpen(parser))>

```

```

override <accessLevelOpenOK(parser)>
func getGrammarFileName() -> String { return "<parser.grammarFileName; format="java-escape">" }

```

```

override <accessLevelOpenOK(parser)>
func getRuleNames() -> [String] { return <parser.name>.ruleNames }

```

```

override <accessLevelOpenOK(parser)>
func getSerializedATN() -> String { return <parser.name>._serializedATN }

```

```

override <accessLevelOpenOK(parser)>
func getATN() -> ATN { return <parser.name>._ATN }

```

```

<namedActions.members>
<parser:(ctor)()>
<funcs; separator="\n">

```

```

<if(sempredFuncs)>
override <accessLevelOpenOK(parser)>
func sempred(_ _localctx: RuleContext?, _ ruleIndex: Int, _ predIndex: Int)throws -> Bool {
  switch (ruleIndex) {
    <parser.sempredFuncs.values:{f}
  case <f.ruleIndex>:

```

```

return try <f.name>_sempred(_localctx?.castdown(<f.ctxType>.self), predIndex)); separator="\n">
    default: return true
}
<!return true;!>
}
<sempredFuncs.values; separator="\n">
<endif>

<atn>

<accessLevelNotOpen(parser)>
static let _serializedATN = <parser.name>ATN().jsonString

<accessLevelNotOpen(parser)>
static let _ATN = ATNDeserializer().deserializeFromJson(_serializedATN)
}
>>

vocabulary(literalNames, symbolicNames, accessLevel) ::= <<
private static let _LITERAL_NAMES: [String?] = [
<literalNames:{t | <t>}; null="nil", separator=", ", wrap, anchor>
]
private static let _SYMBOLIC_NAMES: [String?] = [
<symbolicNames:{t | <t>}; null="nil", separator=", ", wrap, anchor>
]
<accessLevel>
static let VOCABULARY = Vocabulary(_LITERAL_NAMES, _SYMBOLIC_NAMES)
>>

dumpActions(recog, argFuncs, actionFuncs, sempredFuncs) ::= <<
<if(actionFuncs)>
override <accessLevelOpenOK(parser)>
func action(_ _localctx:RuleContext?, _ ruleIndex: Int, _ actionIndex: Int) throws {
switch (ruleIndex) {
<recog.actionFuncs.values:{f}
case <f.ruleIndex>:
<f.name>_action((_localctx as <f.ctxType>?), actionIndex)
}; separator="\n">
default: break
}
}
<actionFuncs.values; separator="\n">
<endif>
<if(sempredFuncs)>
override <accessLevelOpenOK(parser)>
func sempred(_ _localctx:RuleContext?, _ ruleIndex: Int, _ predIndex: Int) throws -> Bool {
switch (ruleIndex) {
<recog.sempredFuncs.values:{f}

```

```

case <f.ruleIndex>:
  return try <f.name>_sempred(_localctx?.castdown(<f.ctxType>.self), predIndex)); separator="\n">
  default: return true
}
<!return true;!>
}
<sempredFuncs.values; separator="\n">
<endif>
>>

```

```

parser_ctor(p) ::= <<

```

```

override <accessLevelOpenOK(parser)>
func getVocabulary() -> Vocabulary {
  return <p.name>.VOCABULARY
}

```

```

override <accessLevelNotOpen(parser)>
init(_ input:TokenStream) throws {
  RuntimeMetaData.checkVersion("4.8", RuntimeMetaData.VERSION)
  try super.init(input)
  _interp = ParserATNSimulator(self,<p.name>._ATN,<p.name>._decisionToDFA,
<parser.name>._sharedContextCache)
}

```

```

>>

```

```

/* This generates a private method since the actionIndex is generated, making an
* overriding implementation impossible to maintain.
*/

```

```

RuleActionFunction(r, actions) ::= <<
private func <r.name>_action(_ localctx: <r.ctxType>?, _ actionIndex: Int) {
  switch (actionIndex) {
  <actions:{index|
  case <index>:
  <actions.(index)>
  }; separator="\n">
  default: break
  }
}
}
>>

```

```

/* This generates a private method since the predIndex is generated, making an
* overriding implementation impossible to maintain.
*/

```

```

RuleSempredFunction(r, actions) ::= <<
private func <r.name>_sempred(_ localctx: <r.ctxType>!, _ predIndex: Int) throws -> Bool {
  switch (predIndex) {

```

```

<actions:{index|
  case <index>:return <actions.(index)>; separator="\n">
  default: return true
}
<!return true;!>
}
>>

```

```

RuleFunction(currentRule,args,code,locals,ruleCtx,altLabelCtxs,namedActions,finallyAction,postamble,exceptions)
::= <<
<ruleCtx>
<altLabelCtxs:{l | <altLabelCtxs.(l)>; separator="\n">
@discardableResult
<if(currentRule.modifiers)><currentRule.modifiers:{f | <f> }><else> <accessLevelOpenOK(parser)> func
<endif><currentRule.name><(if(first(args))>_<endif><args; separator=", _">) throws -> <currentRule.ctxType> {
  var _localctx: <currentRule.ctxType> = <currentRule.ctxType>(_ctx, getState()<currentRule.args:{a | ,
<a.name>}>>)
  try enterRule(_localctx, <currentRule.startState>, <parser.name>.RULE_<currentRule.name>)
  <namedActions.init>
  <locals; separator="\n">
  defer {
    <finallyAction>
    try! exitRule()
  }
  do {
<if(currentRule.hasLookaheadBlock)>
  var _alt:Int
<endif>
  <code>
  <postamble; separator="\n">
  <namedActions.after>
  }
<if(exceptions)>
  <exceptions; separator="\n">
<else>
  catch ANTLRException.recognition(let re) {
    _localctx.exception = re
    _errHandler.reportError(self, re)
    try _errHandler.recover(self, re)
  }
<endif>

  return _localctx
}
>>

```

```

LeftRecursiveRuleFunction(currentRule,args,code,locals,ruleCtx,altLabelCtxs,
  namedActions,finallyAction,postamble) ::=

```

<<

<ruleCtx>

<altLabelCtxs:{l | <altLabelCtxs.(l)>}; separator="\n">

<if(currentRule.modifiers)><currentRule.modifiers:{f | <f> }><else> <accessLevelNotOpen(parser)> final <endif>

func <currentRule.name>(<if(first(args))>_ <endif><args; separator=", _">) throws -> <currentRule.ctxType> {

return try <currentRule.name>(0<currentRule.args:{a | , <a.name>}>)

}

@discardableResult

private func <currentRule.name>(_ _p<args:{a | , <a>}>: Int) throws -> <currentRule.ctxType> {

let _parentctx: ParserRuleContext? = _ctx

var _parentState: Int = getState()

var _localctx: <currentRule.ctxType> = <currentRule.ctxType>(_ctx, _parentState<currentRule.args:{a | , <a.name>}>)

var _prevctx: <currentRule.ctxType> = _localctx

var _startState: Int = <currentRule.startState>

try enterRecursionRule(_localctx, <currentRule.startState>, <parser.name>.RULE_<currentRule.name>, _p)

<namedActions.init>

<locals; separator="\n">

defer {

<finallyAction>

try! unrollRecursionContexts(_parentctx)

}

do {

<if(currentRule.hasLookaheadBlock)>

var _alt: Int

<endif>

<code>

<postamble; separator="\n">

<namedActions.after>

}

catch ANTLRException.recognition(let re) {

_localctx.exception = re

_errHandler.reportError(self, re)

try _errHandler.recover(self, re)

}

return _localctx;

}

>>

CodeBlockForOuterMostAlt(currentOuterMostAltCodeBlock, locals, preamble, ops) ::= <<

<if(currentOuterMostAltCodeBlock.altLabel)>_localctx = <currentOuterMostAltCodeBlock.altLabel;

format="cap">Context(_localctx);<endif>

try enterOuterAlt(_localctx, <currentOuterMostAltCodeBlock.alt.altNum>)

<CodeBlockForAlt(currentAltCodeBlock=currentOuterMostAltCodeBlock, ...)>

>>

```

CodeBlockForAlt(currentAltCodeBlock, locals, preamble, ops) ::= <<
<!/{!>
<locals; separator="\n">
<preamble; separator="\n">
<ops; separator="\n">
<!/{!>
>>

```

```

LL1AltBlock(choice, preamble, alts, error) ::= <<
setState(<choice.stateNumber>)
try _errHandler.sync(self)
<if(choice.label)><labelref(choice.label)> = try _input.LT(1)<endif>
<preamble; separator="\n">
switch (<parser.name>.Tokens(rawValue: try _input.LA(1))!) {
<choice.altLook,alts:{look,alt | <cases(ttypes=look)>
<alt>
break }; separator="\n">
default:
<error>
}
>>

```

```

LL1OptionalBlock(choice, alts, error) ::= <<
setState(<choice.stateNumber>)
try _errHandler.sync(self)
switch (<parser.name>.Tokens(rawValue: try _input.LA(1))!) {
<choice.altLook,alts:{look,alt | <cases(ttypes=look)>
<alt>
break }; separator="\n">
default:
break
}
>>

```

```

LL1OptionalBlockSingleAlt(choice, expr, alts, preamble, error, followExpr) ::= <<
setState(<choice.stateNumber>)
try _errHandler.sync(self)
<preamble; separator="\n">
if (<expr>) {
<alts; separator="\n">
}
<!else if ( !(<followExpr> ) <error>!>
>>

```

```

LL1StarBlockSingleAlt(choice, loopExpr, alts, preamble, iteration) ::= <<
setState(<choice.stateNumber>)
try _errHandler.sync(self)

```

```

<preamble; separator="\n">
while (<loopExpr>) {
  <alts; separator="\n">
  setState(<choice.loopBackStateNumber>)
  try _errHandler.sync(self)
  <iteration>
}
>>

LL1PlusBlockSingleAlt(choice, loopExpr, alts, preamble, iteration) ::= <<
setState(<choice.blockStartStateNumber>) <! alt block decision !>
try _errHandler.sync(self)
<preamble; separator="\n">
repeat {
  <alts; separator="\n">
  setState(<choice.stateNumber>); <! loopback/exit decision !>
  try _errHandler.sync(self)
  <iteration>
} while (<loopExpr>)
>>

// LL(*) stuff

AltBlock(choice, preamble, alts, error) ::= <<
setState(<choice.stateNumber>)
try _errHandler.sync(self)
<if(choice.label)><labelref(choice.label)> = try _input.LT(1)<endif>
<preamble; separator="\n">
switch(try getInterpreter().adaptivePredict(_input,<choice.decision>,_ctx)) {
  <alts:{alt |
  case <i>:
    <alt>
    break }; separator="\n">
  default: break
}
>>

OptionalBlock(choice, alts, error) ::= <<
setState(<choice.stateNumber>)
try _errHandler.sync(self)
switch (try getInterpreter().adaptivePredict(_input,<choice.decision>,_ctx)) {
  <alts:{alt |
  case <i><if(!choice.ast.greedy)>+1<endif>:
    <alt>
    break }; separator="\n">
  default: break
}
>>

```

```

StarBlock(choice, alts, sync, iteration) ::= <<
setState(<choice.stateNumber>)
try _errHandler.sync(self)
_alt = try getInterpreter().adaptivePredict(_input,<choice.decision>,_ctx)
while (_alt != <choice.exitAlt> && _alt != ATN.INVALID_ALT_NUMBER) {
if ( _alt==1<if(!choice.ast.greedy)>+1<endif> ) {
<iteration>
<alts> <! should only be one !>
}
setState(<choice.loopBackStateNumber>)
try _errHandler.sync(self)
_alt = try getInterpreter().adaptivePredict(_input,<choice.decision>,_ctx)
}
>>

```

```

PlusBlock(choice, alts, error) ::= <<
setState(<choice.blockStartStateNumber>); <! alt block decision !>
try _errHandler.sync(self)
_alt = 1<if(!choice.ast.greedy)>+1<endif>;
repeat {
switch (_alt) {
<alts:{alt|
case <i><if(!choice.ast.greedy)>+1<endif>:
<alt>
break }; separator="\n">
default:
<error>
}
setState(<choice.loopBackStateNumber>); <! loopback/exit decision !>
try _errHandler.sync(self)
_alt = try getInterpreter().adaptivePredict(_input,<choice.decision>,_ctx)
} while (_alt != <choice.exitAlt> && _alt != ATN.INVALID_ALT_NUMBER)
>>

```

```

Sync(s) ::= "sync(<s.expecting.name>);"

```

```

ThrowNoViableAlt(t) ::= "throw ANTLRException.recognition(e: NoViableAltException(self))"

```

```

TestSetInline(s) ::= <<
<!<s.bitsets:{ bits | <if(rest(rest(bits.ttypes)))><bitsetBitfieldComparison(s, bits)><else><bitsetInlineComparison(s,
bits)><endif>}; separator=" || ">!>
//closure
{ () -> Bool in
<if(rest(s.bitsets)>var<else>let<endif> testSet: Bool = <first(s.bitsets):{ bits |
<if(rest(rest(bits.ttypes)))><bitsetBitfieldComparison(s, bits)><else><bitsetInlineComparison(s, bits)><endif>}>
<rest(s.bitsets):{ bits | testSet = testSet || <if(rest(rest(bits.ttypes)))><bitsetBitfieldComparison(s,
bits)><else><bitsetInlineComparison(s, bits)><endif>}; separator="\n">

```

```

    return testSet
  }()
  >>

// Java language spec 15.19 - shift operators mask operands rather than overflow to 0... need range test
testShiftInRange(shiftAmount) ::= <<
((<shiftAmount>) & ~0x3f) == 0
>>

// produces smaller bytecode only when bits.ttypes contains more than two items
bitsetBitfieldComparison(s, bits) ::= <<
<!(<testShiftInRange({<offsetShift(s.varName, bits.shift)>})> && ((1 \<< <offsetShift(s.varName, bits.shift)>) &
(<bits.ttypes:{ ttype | (1 \<< <offsetShift(ttype, bits.shift)>); separator=" | ">) != 0)!>
{ () -> Bool in
  <! let test: Bool = (<testShiftInRange({<offsetShift(s.varName, bits.shift)>})>)!>
  <!var temp: Int64 = Int64(<offsetShift(s.varName, bits.shift)>)!>
  <!temp = (temp \< 0) ? (64 + (temp % 64)) : (temp % 64)!>
  <!let test1: Int64 = (Int64(1) \<< temp)!>
  <!var test2: Int64 = (<first(bits.ttypes):{ ttype | Utils.bitLeftShift(<offsetShift(parserName(ttype), bits.shift)>)}>)!>
  <!<rest(bits.ttypes):{ ttype | test2 = test2 | Utils.bitLeftShift(<offsetShift(parserName(ttype), bits.shift)>)}>);
separator="\n">!>
  let testArray: [Int] = [<s.varName>, <bits.ttypes:{ ttype |<parserName(ttype)>}; separator=", ">]
  <!var test2: Int64 = Utils.testBitLeftShiftArray(testArray)!>
  return Utils.testBitLeftShiftArray(testArray, <bits.shift>)
}()
>>

isZero ::= [
"0": true,
default: false
]
parserName(ttype) ::= <%
<parser.name>.Tokens.<ttype>.rawValue
%>
offsetShift(shiftAmount, offset) ::= <%
<if(!isZero.(offset))><shiftAmount> - <offset><else><shiftAmount><endif>
%>

// produces more efficient bytecode when bits.ttypes contains at most two items
bitsetInlineComparison(s, bits) ::= <%
<bits.ttypes:{ ttype | <s.varName> == <parser.name>.Tokens.<ttype>.rawValue }; separator=" || ">
%>

cases(ttypes) ::= <<
<trunc(ttypes): { t | case .<t>:fallthrough } ; separator="\n">
<last(ttypes): { t | case .<t>: } ; separator="\n">
>>

```

```

InvokeRule(r, argExprsChunks) ::= <<
  setState(<r.stateNumber>)
  <if(r.labels)>
  try {
    let assignmentValue = try
  <r.name>(<if(r.ast.options.p)><r.ast.options.p><if(argExprsChunks)>,<endif><endif><argExprsChunks>)
  <r.labels:{1 | <labelref(l)> = assignmentValue } ; separator="\n">
    }()
  <else>try
  <r.name>(<if(r.ast.options.p)><r.ast.options.p><if(argExprsChunks)>,<endif><endif><argExprsChunks>)<endif>
  >>

```

```

MatchToken(m) ::= <<
  setState(<m.stateNumber>)
  <if(m.labels)>
  try {
    let assignmentValue = try match(<parser.name>.Tokens.<m.name>.rawValue)
  <m.labels:{1 | <labelref(l)> = assignmentValue } ; separator="\n">
    }()
  <else>try match(<parser.name>.Tokens.<m.name>.rawValue)<endif>
  >>

```

```

MatchSet(m, expr, capture) ::= "<CommonSetStuff(m, expr, capture, false)>"

```

```

MatchNotSet(m, expr, capture) ::= "<CommonSetStuff(m, expr, capture, true)>"

```

```

CommonSetStuff(m, expr, capture, invert) ::= <<
  setState(<m.stateNumber>)
  <if(m.labels)><m.labels:{1 | <labelref(l)> = }>try _input.LT(1)<endif>
  <capture>
  if (<if(invert)><m.varName> \<= 0 || <else>!<endif>(<expr>)) {
  <if(m.labels)><m.labels:{1 | <labelref(l)> = }><endif>try _errHandler.recoverInline(self)<if(m.labels)> as
  Token<endif>
  }
  else {
  _errHandler.reportMatch(self)
  try consume()
  }
  >>

```

```

Wildcard(w) ::= <<
  setState(<w.stateNumber>)
  <if(w.labels)><w.labels:{1 | <labelref(l)> = }><endif>try matchWildcard();
  >>

```

```

// ACTION STUFF

```

```

Action(a, foo, chunks) ::= "<chunks>"

```

```

ArgAction(a, chunks) ::= "<chunks>"

SemPred(p, chunks, failChunks) ::= <<
setState(<p.stateNumber>)
if (!(<chunks>)) {
    throw ANTLRException.recognition(e:FailedPredicateException(self, <p.predicate><if(failChunks)>,
<failChunks><elseif(p.msg)>, <p.msg><endif>))
}
>>

ExceptionClause(e, catchArg, catchAction) ::= <<
catch (<catchArg>) {
    <catchAction>
}
>>

// lexer actions are not associated with model objects

LexerSkipCommand() ::= "skip()"
LexerMoreCommand() ::= "more()"
LexerPopModeCommand() ::= "popMode()"

LexerTypeCommand(arg)    ::= "_type = <arg>"
LexerChannelCommand(arg) ::= "_channel = <arg>"
LexerModeCommand(arg)    ::= "_mode = <arg>"
LexerPushModeCommand(arg) ::= "pushMode(<arg>)"

ActionText(t) ::= "<t.text>"
ActionTemplate(t) ::= "<t.st>"
ArgRef(a) ::= "_localctx.<a.name>"
LocalRef(a) ::= "_localctx.<a.name>"
RetValRef(a) ::= "_localctx.<a.name>"
QRetValRef(a) ::= "<ctx(a)>.<a.dict>.<a.name>"
/** How to translate $tokenLabel */
TokenRef(t) ::= "<ctx(t)>.<t.name>"
LabelRef(t) ::= "<ctx(t)>.<t.name>"
ListLabelRef(t) ::= "<ctx(t)>.<ListLabelName(t.name)>"
SetAttr(s,rhsChunks) ::= "<ctx(s)>.<s.name> = <rhsChunks>"

TokenLabelType() ::= "<file.TokenLabelType; null={Token}>"
InputSymbolType() ::= "<file.InputSymbolType; null={Token}>"

TokenPropertyRef_text(t) ::= "(<ctx(t)>.<t.label> != nil ? <ctx(t)>.<t.label>!.getText()! : \"\")"
TokenPropertyRef_type(t) ::= "(<ctx(t)>.<t.label> != nil ? <ctx(t)>.<t.label>!.getType() : 0)"
TokenPropertyRef_line(t) ::= "(<ctx(t)>.<t.label> != nil ? <ctx(t)>.<t.label>!.getLine() : 0)"
TokenPropertyRef_pos(t) ::= "(<ctx(t)>.<t.label> != nil ? <ctx(t)>.<t.label>!.getCharPositionInLine() : 0)"
TokenPropertyRef_channel(t) ::= "(<ctx(t)>.<t.label> != nil ? <ctx(t)>.<t.label>!.getChannel() : 0)"

```

```

TokenPropertyRef_index(t) ::= "<ctx(t)>.<t.label> != nil ? <ctx(t)>.<t.label>!.getTokenIndex() : 0)"
TokenPropertyRef_int(t) ::= "<ctx(t)>.<t.label> != nil ? Int(<ctx(t)>.<t.label>!.getText()) : 0)"

RulePropertyRef_start(r) ::= "<ctx(r)>.<r.label> != nil ? (<ctx(r)>.<r.label>!.start?.description ?? \"\"): \"\)"
RulePropertyRef_stop(r) ::= "<ctx(r)>.<r.label> != nil ? (<ctx(r)>.<r.label>!.stop?.description ?? \"\"): \"\)"
RulePropertyRef_text(r) ::= "<ctx(r)>.<r.label> != nil ? try
_input.getText(<ctx(r)>.<r.label>!.start,<ctx(r)>.<r.label>!.stop) : \"\)"
RulePropertyRef_ctx(r) ::= "<ctx(r)>.<r.label>"
RulePropertyRef_parser(r) ::= "self"

ThisRulePropertyRef_start(r) ::= "_localctx.start"
ThisRulePropertyRef_stop(r) ::= "_localctx.stop"
ThisRulePropertyRef_text(r) ::= "(try _input.getText(_localctx.start, try _input.LT(-1)))"
ThisRulePropertyRef_ctx(r) ::= "_localctx"
ThisRulePropertyRef_parser(r) ::= "self"

NonLocalAttrRef(s) ::= "(<s.ruleName; format=\\\"cap\\\">Context)getInvokingContext(<s.ruleIndex>).<s.name>"
SetNonLocalAttr(s, rhsChunks) ::=
"(<s.ruleName; format=\\\"cap\\\">Context)getInvokingContext(<s.ruleIndex>).<s.name> = <rhsChunks>"

AddToLabelList(a) ::= "<ctx(a.label)>.<a.listName>.append(<labelref(a.label)>)"

TokenDecl(t) ::= "<t.name>: <SwiftTypeMap.(TokenLabelType())>!"
TokenTypeDecl(t) ::= "var <t.name>: Int = 0"
TokenListDecl(t) ::= "<t.name>: [Token] = [Token]()"
RuleContextDecl(r) ::= "<r.name>: <r.ctxName>!"
RuleContextListDecl(rdecl) ::= "<rdecl.name>: [<rdecl.ctxName>] = [<rdecl.ctxName>]()"

ContextTokenGetterDecl(t) ::= <<
<accessLevelOpenOK(parser)>
func <t.name>() -> TerminalNode? {
return getToken(<parser.name>.Tokens.<t.name>.rawValue, 0)
}
>>

ContextTokenListGetterDecl(t) ::= <<
<accessLevelOpenOK(parser)>
func <t.name>() -> [TerminalNode] {
return getTokens(<parser.name>.Tokens.<t.name>.rawValue)
}
>>

ContextTokenListIndexedGetterDecl(t) ::= <<
<accessLevelOpenOK(parser)>
func <t.name>(_ i:Int) -> TerminalNode? {
return getToken(<parser.name>.Tokens.<t.name>.rawValue, i)
}
>>

ContextRuleGetterDecl(r) ::= <<
<accessLevelOpenOK(parser)>

```

```

func <r.name>() -> <r.ctxName>? {
  return getRuleContext(<r.ctxName>.self, 0)
}
>>
ContextRuleListGetterDecl(r) ::= <<
  <accessLevelOpenOK(parser)>
  func <r.name>() -> [<r.ctxName>] {
    return getRuleContexts(<r.ctxName>.self)
  }
>>
ContextRuleListIndexedGetterDecl(r) ::= <<
  <accessLevelOpenOK(parser)>
  func <r.name>(_ i: Int) -> <r.ctxName>? {
    return getRuleContext(<r.ctxName>.self, i)
  }
>>

LexerRuleContext() ::= "RuleContext"

/** The rule context name is the rule followed by a suffix; e.g.,
 * r becomes rContext.
 */
RuleContextNameSuffix() ::= "Context"

ImplicitTokenLabel(tokenName) ::= "_<tokenName>"
ImplicitRuleLabel(ruleName) ::= "_<ruleName>"
ImplicitSetLabel(id) ::= "_tset<id>"
ListLabelName(label) ::= "<label>"

CaptureNextToken(d) ::= "<d.varName> = try _input.LT(1)"
CaptureNextTokenType(d) ::= "<d.varName> = try _input.LA(1)"

StructDecl(struct,ctorAttrs,attrs,getters,dispatchMethods,interfaces,extensionMembers,
  superClass={ParserRuleContext}) ::= <<

<accessLevelNotOpen(parser)> class <struct.name>:
<if(contextSuperClass)><contextSuperClass><else>ParserRuleContext<endif><if(interfaces)>, <interfaces>
separator=", "><endif> {
  <attrs:{a | <accessLevelOpenOK(parser)> var <a>}; separator="\n">
  <getters:{g | <g>}; separator="\n">
  <! <if(ctorAttrs)> <accessLevelNotOpen(parser)> init(_ parent: ParserRuleContext,_ invokingState: Int) {
super.init(parent, invokingState) }<endif> !>
  <if(ctorAttrs)>
  <accessLevelNotOpen(parser)> convenience init(_ parent: ParserRuleContext?, _ invokingState: Int<ctorAttrs:{a | ,
_ <a>}>)> {
  self.init(parent, invokingState)
  <struct.ctorAttrs:{a | self.<a.name> = <a.name>;}; separator="\n">
  }
}

```

```

<endif>

override <accessLevelOpenOK(parser)>
func getRuleIndex() -> Int {
  return <parser.name>.RULE_<struct.derivedFromName>
}
<if(struct.provideCopyFrom && struct.attrs)> <! don't need copy unless we have subclasses !>
<accessLevelOpenOK(parser)>
override func copyFrom(_ ctx_: ParserRuleContext) {
  super.copyFrom(ctx_)
  let ctx = ctx_ as! <struct.name>
  <struct.attrs:{ a | self.<a.name> = ctx.<a.name>; }; separator="\n">
}
<endif>
<dispatchMethods; separator="\n">
<extensionMembers; separator="\n">
}
>>

AltLabelStructDecl(struct,attrs, getters,dispatchMethods) ::= <<
<accessLevelNotOpen(parser)> class <struct.name>: <currentRule.name; format="cap">Context {
  <attrs:{ a | <accessLevelNotOpen(parser)> var <a> }; separator="\n">
  <getters:{ g | <g> }; separator="\n">

  <accessLevelNotOpen(parser)>
  init(_ ctx: <currentRule.name; format="cap">Context) {
    super.init()
    copyFrom(ctx)
  }
  <dispatchMethods; separator="\n">
}
>>

ListenerDispatchMethod(method) ::= <<
override <accessLevelOpenOK(parser)>
func <if(method.isEnter)>enter<else>exit<endif>Rule(_ listener: ParseTreeListener) {
  if let listener = listener as? <parser.grammarName>Listener {
    listener.<if(method.isEnter)>enter<else>exit<endif><struct.derivedFromName; format="cap">(self)
  }
}
>>

VisitorDispatchMethod(method) ::= <<
override <accessLevelOpenOK(parser)>
func accept<T>(_ visitor: ParseTreeVisitor<T>) -> T? {
  if let visitor = visitor as? <parser.grammarName>Visitor {
    return visitor.visit<struct.derivedFromName; format="cap">(self)
  }
}

```

```

else if let visitor = visitor as? <parser.grammarName>BaseVisitor {
    return visitor.visit<struct.derivedFromName; format="cap">(self)
}
else {
    return visitor.visitChildren(self)
}
}
}
>>

```

```

AttributeDecl(d) ::= "<d.name>: <SwiftTypeMap.(d.type)><if(d.initValue)> = <d.initValue><else>!<endif>"

```

```

/** If we don't know location of label def x, use this template (_localctx as! <x.ctx.name> */
labelref(x) ::= "<if(!x.isLocal)>_localctx.castdown(<x.ctx.name>.self).<endif><x.name>"

```

```

/** For any action chunk, what is correctly-typed context struct ptr? */
ctx(actionChunk) ::= "_localctx.castdown(<actionChunk.ctx.name>.self)"

```

```

// used for left-recursive rules

```

```

recRuleAltPredicate(ruleName,opPrec) ::= "precpred(_ctx, <opPrec>)"
recRuleSetReturnAction(src,name) ::= "$<name>=$<src>.<name>"
recRuleSetStopToken() ::= "_ctx!.stop = try _input.LT(-1)"

```

```

recRuleAltStartAction(ruleName, ctxName, label, isListLabel) ::= <<
_localctx = <ctxName>Context(_parentctx, _parentState);
<if(label)>
<if(isListLabel)>
_localctx.<label>.append(_prevctx)
<else>
_localctx.<label> = _prevctx
<endif>
<endif>
<if(label)>_localctx.<label> = _prevctx;<endif>
try pushNewRecursionContext(_localctx, _startState, <parser.name>.RULE_<ruleName>)
>>

```

```

recRuleLabeledAltStartAction(ruleName, currentAltLabel, label, isListLabel) ::= <<
_localctx = <currentAltLabel; format="cap">Context( <ruleName; format="cap">Context(_parentctx,
_parentState))
<if(label)>
<if(isListLabel)>
(_localctx as! <currentAltLabel; format="cap">Context).<label>.append(_prevctx)
<else>
(_localctx as! <currentAltLabel; format="cap">Context).<label> = _prevctx
<endif>
<endif>
try pushNewRecursionContext(_localctx, _startState, <parser.name>.RULE_<ruleName>)
>>

```

```

recRuleReplaceContext(ctxName) ::= <<
  _localctx = <ctxName>Context(_localctx)
  _ctx = _localctx
  _prevctx = _localctx
>>

```

```

recRuleSetPrevCtx() ::= <<
  if _parseListeners != nil {
    try triggerExitRuleEvent()
  }
  _prevctx = _localctx
>>

```

```

LexerFile(lexerFile, lexer, namedActions) ::= <<
  <fileHeader(lexerFile.grammarFileName, lexerFile.ANTLRVersion)>
  <if(lexerFile.genPackage)>
  <!package <lexerFile.genPackage>;!>
  <endif>
  <namedActions.header>
  import Antlr4

  <lexer>
>>

```

```

Lexer(lexer, atn, actionFuncs, sempredFuncs, superClass) ::= <<
  <accessLevelOpenOK(lexer)> class <lexer.name>: <superClass; null="Lexer"> {

  internal static var _decisionToDFA: [DFA] = {
    var decisionToDFA = [DFA]()
    let length = <lexer.name>._ATN.getNumberOfDecisions()
    for i in 0..<length {
      <! decisionToDFA[i] = DFA(<lexer.name>._ATN.getDecisionState(i)!, i);!>
      decisionToDFA.append(DFA(<lexer.name>._ATN.getDecisionState(i)!, i))
    }
    return decisionToDFA
  }()

```

```

  internal static let _sharedContextCache = PredictionContextCache()

```

```

  <accessLevelNotOpen(lexer)>
  static let <lexer.tokens>:{k | <k>=<lexer.tokens.(k)>}; separator=", ", wrap, anchor>

```

```

  <if(lexer.channels)>
  <accessLevelNotOpen(lexer)>
  static let <lexer.channels>:{k | <k>=<lexer.channels.(k)>}; separator=", ", wrap, anchor>
  <endif>

```

```

  <if(rest(lexer.modes))>
  <accessLevelNotOpen(lexer)>

```

```

static let <rest(lexer.modes):{m| <m>=<i>}; separator=", ", wrap, anchor>
<endif>
<accessLevelNotOpen(lexer)>
static let channelNames: [String] = [
  "DEFAULT_TOKEN_CHANNEL", "HIDDEN"<if (lexer.channels)>, <lexer.channels:{c| "<c>"}; separator=", ",
wrap, anchor><endif>
]

<accessLevelNotOpen(lexer)>
static let modeNames: [String] = [
  <lexer.modes:{m| "<m>"}; separator=", ", wrap, anchor>
]

<accessLevelNotOpen(lexer)>
static let ruleNames: [String] = [
  <lexer.ruleNames:{r| "<r>"}; separator=", ", wrap, anchor>
]

<vocabulary(lexer.literalNames, lexer.symbolicNames,
  accessLevelNotOpen(lexer))>

<namedActions.members>

override <accessLevelOpenOK(lexer)>
func getVocabulary() -> Vocabulary {
  return <lexer.name>.VOCABULARY
}

<accessLevelNotOpen(lexer)>
required init(_ input: CharStream) {
  RuntimeMetaData.checkVersion("<lexerFile.ANTLRVersion>", RuntimeMetaData.VERSION)
  super.init(input)
  _interp = LexerATNSimulator(self, <lexer.name>._ATN, <lexer.name>._decisionToDFA,
<lexer.name>._sharedContextCache)
}

override <accessLevelOpenOK(lexer)>
func getGrammarFileName() -> String { return "<lexer.grammarFileName>" }

override <accessLevelOpenOK(lexer)>
func getRuleNames() -> [String] { return <lexer.name>.ruleNames }

override <accessLevelOpenOK(lexer)>
func getSerializedATN() -> String { return <lexer.name>._serializedATN }

override <accessLevelOpenOK(lexer)>
func getChannelNames() -> [String] { return <lexer.name>.channelNames }

```

```

override <accessLevelOpenOK(lexer)>
func getModeNames() -> [String] { return <lexer.name>.modeNames }

override <accessLevelOpenOK(lexer)>
func getATN() -> ATN { return <lexer.name>._ATN }

<dumpActions(lexer, "", actionFuncs, sempredFuncs)>
<atn>

<accessLevelNotOpen(lexer)>
static let _serializedATN: String = <lexer.name>ATN().jsonString

<accessLevelNotOpen(lexer)>
static let _ATN: ATN = ATNDeserializer().deserializeFromJson(_serializedATN)
}
>>

/** Don't need to define anything. The tool generates a XParserATN.swift file (and same for lexer)
 * which is referenced from static field _serializedATN. This json string is passed to
 * deserializeFromJson(). Note this is not the "serialization as array of ints" that other targets
 * do. It is more or less the output of ATNPrinter which gets read back in.
 */
SerializedATN(model) ::= <<
>>

/** Using a type to init value map, try to init a type; if not in table
 * must be an object, default value is "null".
 */
initValue(typeName) ::= <<
<SwiftTypeInitMap.(typeName)>
>>

codeFileExtension() ::= ".swift"

Found in path(s):
* /opt/cola/permits/1473460068_1668481512.5494933/0/antlr4-4-8-1-
jar/org/antlr/v4/tool/templates/codegen/Swift/Swift.stg
No license file was found, but licenses were detected in source scan.

<!--
~ Copyright (c) 2012-2017 The ANTLR Project. All rights reserved.
~ Use of this file is governed by the BSD 3-clause license that
~ can be found in the LICENSE.txt file in the project root.
-->

Found in path(s):
* /opt/cola/permits/1473460068_1668481512.5494933/0/antlr4-4-8-1-jar/META-
INF/maven/org/antlr/antlr4/pom.xml

```

1.136 fabric8-::-kubernetes-model-::-batch

4.13.3

1.136.1 Available under license :

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0

Bnd-LastModified: 1619068547404

Build-Jdk-Spec: 1.8

Bundle-Description: Java client for Kubernetes and OpenShift

Bundle-DocURL: <http://redhat.com>

Bundle-License: <http://www.apache.org/licenses/LICENSE-2.0.txt>

Bundle-ManifestVersion: 2

Bundle-Name: Fabric8 :: Kubernetes Model :: Batch

Bundle-SymbolicName: io.fabric8.kubernetes-model-batch

Bundle-Vendor: Red Hat

Bundle-Version: 4.13.3

Created-By: Apache Maven Bundle Plugin

Export-Package: io.fabric8.kubernetes.api.model.batch;uses:="com.fasterxml.jackson.annotation,com.fasterxml.jackson.databind,com.fasterxml.jackson.databind.annotation,io.fabric8.kubernetes.api.builder,io.fabric8.kubernetes.api.model,io.fabric8.kubernetes.model.annotation";version="4.13.3"

Implementation-Title: Fabric8 :: Kubernetes Model :: Batch

Implementation-Vendor: Red Hat

Implementation-Version: 4.13.3

Import-Package: com.fasterxml.jackson.annotation;version="[2.11,3)",com.fasterxml.jackson.databind;version="[2.11,3)",com.fasterxml.jackson.databind.annotation;version="[2.11,3)",io.fabric8.kubernetes.api.builder;version="[4.13,5)",io.fabric8.kubernetes.api.model,io.fabric8.kubernetes.model.annotation;version="[4.13,5)"

Require-Capability: osgi.ee;filter:="(&(osgi.ee=JavaSE)(version=1.8))"

Specification-Title: Fabric8 :: Kubernetes Model :: Batch

Specification-Vendor: Red Hat

Specification-Version: 4.13

Tool: Bnd-5.1.1.202006162103

Found in path(s):

* /opt/cola/permits/1288519908_1647861688.22/0/kubernetes-model-batch-4-13-3-jar/META-INF/MANIFEST.MF

No license file was found, but licenses were detected in source scan.

*

* Copyright (C) 2015 Red Hat, Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

- * you may not use this file except in compliance with the License.
- * You may obtain a copy of the License at
- *
- * <http://www.apache.org/licenses/LICENSE-2.0>
- *
- * Unless required by applicable law or agreed to in writing, software
- * distributed under the License is distributed on an "AS IS" BASIS,
- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.
- *

Found in path(s):

* /opt/cola/permits/1288519908_1647861688.22/0/kubernetes-model-batch-4-13-3-jar/manifest.vm

No license file was found, but licenses were detected in source scan.

Copyright (C) 2015 Red Hat, Inc.

Licensed under the Apache License, Version 2.0 (the "License");
 you may not use this file except in compliance with the License.
 You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE>

2.0

Unless required by applicable law or agreed to in writing, software
 distributed under the License is distributed on an "AS IS" BASIS,
 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 See the License for the specific language governing permissions and
 limitations under the License.

Found in path(s):

* /opt/cola/permits/1288519908_1647861688.22/0/kubernetes-model-batch-4-13-3-jar/META-INF/maven/io.fabric8/kubernetes-model-batch/pom.xml

1.137 apache-groovy 3.0.13

1.137.1 Available under license :

Copyright (c) Nicolas Gallagher and Jonathan Neal

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work

(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses

granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]"

replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

ANTLR 2 License

Antlr2 is released in the public domain.
See licenses/antlr2-license.txt for details.

ANTLR 4 License

Antlr4 is released under a BSD 3-clause license.
See licenses/antlr4-license.txt for details.

ASM License

ASM uses a 3-clause BSD license. For details, see licenses/asm-license.txt.

Hamcrest License (needed when using optional JUnit dependency)

This product bundles the Hamcrest jar, which is available under a BSD license. For details, see licenses/hamcrest-license.txt.

JAXB License (for optional groovy-jaxb extensions)

This product bundles several JAXB related jars in binary form.

The "javax.activation" jar is available under the CDDL 1.0 license:
licenses/activation-license.txt

The jaxb-api, jaxb-core and jaxb-impl jars use the CDDL 1.1 license:
licenses/jaxb-license.txt

JLine2 License (optional dependency used with groovysh)

This product bundles the JLine2 jar, which is available under a
BSD License. For details, see licenses/jline2-license.txt.

javax.servlet.jsp-api License (for groovy-servlet module)

This product bundles the javax.servlet.jsp-api jar in binary form
which is available under the CDDL 1.1 license:
licenses/jsp-api-license.txt

JSR166y License (optionally used by the optional GParas dependency)

This product bundles the jsr166y jar (containing works from
the JSR-166 EG, Doug Lea, and Jason T. Greene) made available in
the public domain. For details, see licenses/jsr166y-license.txt.

JSR223 License

The following classes within this product:

org.codehaus.groovy.jsr223.GroovyCompiledScript
org.codehaus.groovy.jsr223.GroovyScriptEngineFactory
org.codehaus.groovy.jsr223.GroovyScriptEngineImpl

were derived from reference implementation files developed by Sun in
collaboration with the Groovy community. The reference implementation
has a BSD-style license. Details can be found in: licenses/jsr223-license.txt

JUnit Licenses (optional dependencies when using Groovy for testing)

This product bundles the JUnit 4 jar, which is available under the Eclipse Public License v1.0. For details, see licenses/junit4-license.txt.

This product bundles several JUnit 5 jars, which are available under the Eclipse Public License v2.0. For details, see licenses/junit5-license.txt.

normalize.css License

The stylesheet.css file (originally normalize.css) is used by the groovydoc and docgenerator components for groovy-jdk/gapi documentation. It is made available under a MIT License:
licenses/normalize-stylesheet-license.txt

javax.servlet.jsp-api License (for groovy-servlet module)

This product bundles the javax.servlet-api jar in binary form which is available under the CDDL 1.0 license:
licenses/servlet-api-license.txt

XStream License (optional dependency when serializing AST as XML)

This product bundles the XStream jar, which is available under a "3-clause BSD" license. For details, see licenses/xstream-license.txt.

This convenience zip embeds Groovy's src and doc zips.
See also src/LICENSE and doc/LICENSE files for additional license information.
Apache Commons Lang
Copyright 2001-2015 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

This product includes software from the Spring Framework,
under the Apache License 2.0 (see: `StringUtils.containsWhitespace()`)

javax.servlet.jsp-api License (for groovy-servlet module)

This product bundles the javax.servlet.jsp-api jar in binary form which is available under the CDDL 1.1 license:

licenses/jsp-api-license.txt

javax.servlet.jsp-api License (for groovy-servlet module)

This product bundles the javax.servlet-api jar in binary form which is available under the CDDL 1.0 license:

licenses/servlet-api-license.txt

ANTLR 2 License

We reserve no legal rights to the ANTLR--it is fully in the public domain. An individual or company may do whatever they wish with source code distributed with ANTLR or the code generated by ANTLR, including the incorporation of ANTLR, or its output, into commercial software.

We encourage users to develop software with ANTLR. However, we do ask that credit is given to us for developing ANTLR. By "credit", we mean that if you use ANTLR or incorporate any source code into one of your programs (commercial product, research project, or otherwise) that you acknowledge this fact somewhere in the documentation, research report, etc... If you like ANTLR and have developed a nice tool with the output, please mention that you developed it using ANTLR. In addition, we ask that the headers remain intact in our source code. As long as these guidelines are kept, we expect to continue enhancing this system and expect to make other tools available as they are completed.

In countries where the Public Domain status of the work may not be valid, the author grants a copyright licence to the general public to deal in the work without restriction and permission to sublicense derivatives under the terms of any (OSI approved) Open Source licence.

The Python parser generator code under antlr/actions/python/ is covered by the 3-clause BSD licence (this part is included in the binary JAR files); the run-time part under lib/python/ is covered by the GNU GPL, version 3 or later (this part is not included in the binary JAR files). See [1] for the full details.

<https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=750643#80%22>

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.1

1. Definitions.

1.1. "Contributor" means each individual or entity that creates or contributes to the creation of Modifications.

1.2. "Contributor Version" means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. "Covered Software" means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. "Executable" means the Covered Software in any form other than Source Code.

1.5. "Initial Developer" means the individual or entity that first makes Original Software available under this License.

1.6. "Larger Work" means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. "License" means this document.

1.8. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. "Modifications" means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. "Original Software" means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. "Source Code" means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than

fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in

combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the

Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Oracle is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section

4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as "Participant") alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. If You assert a patent infringement claim against Participant alleging that the Participant Software directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

6.4. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION

AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and "commercial computer software documentation" as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or

agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

AsciiDoc License

This product uses the style.css from asciidoctor.org within documentation. The file is available under the MIT License. For details, see licenses/asciidoc-style-license.txt.

jQuery License

The following file is used within documentation:

`src/spec/assets/css/jquery-2.1.1.min.js`

This file is made available under the following MIT license:
licenses/jquery-js-license.txt

JSR223 License

The following classes within this product:

`org.codehaus.groovy.jsr223.GroovyCompiledScript`
`org.codehaus.groovy.jsr223.GroovyScriptEngineFactory`
`org.codehaus.groovy.jsr223.GroovyScriptEngineImpl`

were derived from reference implementation files developed by Sun in collaboration with the Groovy community. The reference implementation has a BSD-style license. Details can be found in: licenses/jsr223-license.txt

normalize.css License

The stylesheet.css file (originally normalize.css) is used by the groovydoc and docgenerator components for groovy-jdk/gapi documentation.

It is made available under a MIT License:

licenses/normalize-stylesheet-license.txt

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

1. Definitions.

1.1. Contributor. means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version. means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software. means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable. means the Covered Software in any form other than Source Code.

1.5. Initial Developer. means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work. means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License. means this document.

1.8. Licensable. means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications. means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software. means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims. means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code. means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You. (or .Your.) means an individual or a legal entity exercising rights under, and complying with all of the

terms of, this License. For legal entities, .You. includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, .control. means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the

Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN .AS IS. BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as .Participant.) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software

against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a .commercial item., as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of .commercial computer software. (as that term is defined at 48 C.F.R. ? 252.227-7014(a)(1)) and .commercial computer software documentation. as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

normalize.css License

The stylesheet.css file (originally normalize.css) is used by the groovydoc and docgenerator components for groovy-jdk/gapi documentation.

It is made available under a MIT License:

licenses/normalize-stylesheet-license.txt

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.1

1. Definitions.

1.1. "Contributor" means each individual or entity that creates or contributes to the creation of Modifications.

1.2. "Contributor Version" means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. "Covered Software" means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. "Executable" means the Covered Software in any form other than Source Code.

1.5. "Initial Developer" means the individual or entity that first makes Original Software available under this License.

1.6. "Larger Work" means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. "License" means this document.

1.8. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or

subsequently acquired, any and all of the rights conveyed herein.

1.9. "Modifications" means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. "Original Software" means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. "Source Code" means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original

Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third

party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any

liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Oracle is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or

otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as "Participant") alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60

day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. If You assert a patent infringement claim against Participant alleging that the Participant Software directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

6.4. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and "commercial computer software documentation" as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer

software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

JSR223 License

The following classes within this product:

org.codehaus.groovy.jsr223.GroovyCompiledScript
org.codehaus.groovy.jsr223.GroovyScriptEngineFactory
org.codehaus.groovy.jsr223.GroovyScriptEngineImpl

were derived from reference implementation files developed by Sun in collaboration with the Groovy community. The reference implementation has a BSD-style license. Details can be found in: licenses/jsr223-license.txt
JLine2 License (optional dependency used with groovysh)

This product bundles the JLine2 jar, which is available under a BSD License. For details, see licenses/jline2-license.txt.
JAXB License (for optional groovy-jaxb extensions)

This product bundles several JAXB related jars in binary form.

The "javax.activation" jar is available under the CDDL 1.0 license:
licenses/activation-license.txt

The jaxb-api, jaxb-core and jaxb-impl jars use the CDDL 1.1 license:
licenses/jaxb-license.txt
Apache Groovy
Copyright 2003-2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

This product includes/uses ANTLR2 (<http://www.antlr2.org/>)
developed by Terence Parr 1989-2006

This product includes/uses ANTLR4 (<https://github.com/antlr/antlr4>)
Copyright (c) 2012-2017 The ANTLR Project. All rights reserved.

This product bundles the javax.servlet.jsp-api jar in binary form which is available under the CDDL 1.1 license and for which the following copyright notice applies
Copyright (c) 1997-2013 Oracle and/or its affiliates. All rights reserved.

This product bundles the JUnit4 jar (junit.org)
which is available under the terms of the Eclipse Public License v1.0

This product bundles several of the JUnit5 jars (junit.org)
which are available under the terms of the Eclipse Public License v2.0

This product embeds the OpenBeans jar within its grooid jar artifacts
OpenBeans includes/uses files from Apache Harmony and the following notice applies
Copyright 2006, 2010 The Apache Software Foundation.
Portions of Apache Harmony were originally developed by Intel Corporation and are

licensed to the Apache Software Foundation under the "Software Grant and Corporate Contribution License Agreement" and for which the following copyright notices apply

(C) Copyright 2005 Intel Corporation

(C) Copyright 2005-2006 Intel Corporation

(C) Copyright 2006 Intel Corporation

This product bundles the javax.servlet-api jar in binary form which is available under the CDDL 1.0 license and for which the following copyright notice applies
Copyright (c) 1997-2017 Oracle and/or its affiliates. All rights reserved.

This product bundles icons from the famfamfam.com silk icons set

<http://www.famfamfam.com/lab/icons/silk/>

Licensed under the Creative Commons Attribution Licence v2.5

<http://creativecommons.org/licenses/by/2.5/>

ANTLR 4 License

Antlr4 is released under a BSD 3-clause license.

See licenses/antlr4-license.txt for details.

BSD License

Copyright (c) 2000-2015 www.hamcrest.org

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of Hamcrest nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN

ANY

WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The person or persons who have associated work with this document (the "Dedicator" or "Certifier") hereby either (a) certifies that, to the best of his knowledge, the work of authorship identified is in the public domain of the country from which the work is published, or (b) hereby dedicates whatever copyright the dedicators holds in the work of authorship identified below (the "Work") to the public domain. A certifier, moreover, dedicates any copyright interest he may have in the associated work, and for these purposes, is described as a "dedicator" below.

A certifier has taken reasonable steps to verify the copyright status of this work. Certifier recognizes that his good faith efforts may not shield him from liability if in fact the work certified is not in the public domain.

Dedicator makes this dedication for the benefit of the public at large and to the detriment of the Dedicator's heirs and successors. Dedicator intends this dedication to be an overt act of relinquishment in perpetuity of all present and future rights under copyright law, whether vested or contingent, in the Work. Dedicator understands that such relinquishment of all rights includes the relinquishment of all rights to enforce (by lawsuit or otherwise) those copyrights in the Work.

Dedicator recognizes that, once placed in the public domain, the Work may be freely reproduced, distributed, transmitted, used, modified, built upon, or otherwise exploited by anyone for any purpose, commercial or non-commercial, and in any way, including by methods that have not yet been invented or conceived.

Apache Groovy

Copyright 2003-2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

The Java source files in `src/main/java/org/apache/groovy/util/concurrent/concurrentlinkedhashmap/` are from <https://github.com/ben-manes/concurrentlinkedhashmap> and the following notice applies:
Copyright 2010-2012 Google Inc. All Rights Reserved.

This product bundles icons from the famfamfam.com silk icons set
<http://www.famfamfam.com/lab/icons/silk/>
Licensed under the Creative Commons Attribution Licence v2.5
<http://creativecommons.org/licenses/by/2.5/>

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of

the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works

that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A

PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

AsciiDoc License

This product uses the style.css from asciidoctor.org within documentation. The file is available under the MIT License. For details, see licenses/asciidoc-style-license.txt.

JQuery License

The following file is used within documentation:

`src/spec/assets/css/jquery-2.1.1.min.js`

This file is made available under the following MIT license:
licenses/jquery-js-license.txt

normalize.css License

The stylesheet.css file (originally normalize.css) is used by the groovydoc and docgenerator components for groovy-jdk/gapi documentation. It is made available under a MIT License:
licenses/normalize-stylesheet-license.txt

ASM License

ASM: a very small and fast Java bytecode manipulation framework
Copyright (c) 2000-2011 INRIA, France Telecom
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the

documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2003-2006, Joe Walnes

Copyright (c) 2006-2009, 2011 XStream Committers

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of XStream nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY

WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

JQuery License

The following file is used within documentation:

src/spec/assets/css/jquery-2.1.1.min.js

This file is made available under the following MIT license:

licenses/jquery-js-license.txt

////////////////////////////////////

Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to you under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

////////////////////////////////////

= Contributors

The Groovy team would like to thank the contributors of this documentation (in alphabetical order of last/surname):

- * <https://github.com/mojavelinux>[Dan Allen]
- * <https://github.com/and-dmitry>[Dmitry Andreychuk]
- * <http://hamletdarcy.blogspot.fr/>[Hamlet D'Arcy]
- * <https://github.com/anshbansal>[Aseem Bansal]
- * <https://github.com/bura>[Andrey Bloschetsov]
- * <https://github.com/JBrownVisualSpection>[J Brown]
- * <https://github.com/jeffbrown>[Jeff Scott Brown]
- * <http://twitter.com/CedricChampeau>[Cdric Champeau]
- * <https://github.com/tobia>[Tobia Conforto]
- * <https://github.com/ddimitrov>[Dimitar Dimitrov]
- * <http://twitter.com/werdnagreb>[Andrew Eisenberg]
- * <https://github.com/erdi>[Marcin Erdmann]
- * <https://github.com/christoph-frick>[Christoph Frick]

- * <http://twitter.com/marioggar>[Mario Garca]
- * <https://github.com/davidmichaekarr>[David Michael Karr]
- * http://twitter.com/paulk_asert[Paul King]
- * <http://twitter.com/glaforge>[Guillaume Laforge]
- * <http://twitter.com/pledbrook>[Peter Ledbrook]
- * <http://grantmcconnaughey.github.io/>[Grant McConnaughey]
- * <https://github.com/eric-milles>[Eric Milles]
- * <https://github.com/dnahodil>[David Nahodil]
- * <https://github.com/jnorthr>[James Northrop]
- * <https://github.com/marcpa00>[Marc Paquette]
- * <https://github.com/michaelss>[Michael Schuenck]
- * <https://github.com/PascalSchumacher>[Pascal Schumacher]
- * <https://github.com/shils>[Shil Sinha]
- * <https://github.com/stavytskyi>[Maksym Stavytskyi]
- * <https://twitter.com/asteingr>[Andr Steingre]
- * https://twitter.com/daniel_sun[Daniel Sun]
- * <https://github.com/EPadronU>[Edinson Padrn Urdaneta]
- * <https://github.com/keeganwitt>[Keegan Witt]

Copyright (c) 2002-2012, the original author or authors.

All rights reserved.

<http://www.opensource.org/licenses/bsd-license.php>

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of JLine nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED

AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes

of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You

meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor,

except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

NORMALIZE.CSS LICENSE

The following file is used with documentation:

<org/codehaus/groovy/tools/groovydoc/gstringTemplates/topLevel/stylesheet.css>

Copyright (c) Nicolas Gallagher and Jonathan Neal

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
of the Software, and to permit persons to whom the Software is furnished to do
so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and

attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the

appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

This directory contains generated LICENSE files and snippets used to generate those files.
See the assemble.gradle file (updateLicenses task) for details on how this is done.

Snippets have predefined suffix values in their name to determine which files they go into.

LICENSE (the one for source), LICENSE-DOC and LICENSE-JARJAR

get snippets containing SRC, DOC and JARJAR respectively.

LICENSE-BINZIP gets JARJAR and BINZIP snippets.

In addition, LICENSE files are generated for these subprojects:

groovy-docgenerator, groovy-groovydoc, groovy-groovysh, groovy-jsr223

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or

Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work

or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work

by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

ANTLR 2 License

Antlr2 is released in the public domain.
See licenses/antlr2-license.txt for details.

ANTLR 4 License

Antlr4 is released under a BSD 3-clause license.
See licenses/antlr4-license.txt for details.

ASM License

ASM uses a 3-clause BSD license. For details, see licenses/asm-license.txt.

Hamcrest License (needed when using optional JUnit dependency)

This product bundles the Hamcrest jar, which is available under a

BSD license. For details, see licenses/hamcrest-license.txt.

JAXB License (for optional groovy-jaxb extensions)

This product bundles several JAXB related jars in binary form.

The "javax.activation" jar is available under the CDDL 1.0 license:
licenses/activation-license.txt

The jaxb-api, jaxb-core and jaxb-impl jars use the CDDL 1.1 license:
licenses/jaxb-license.txt

JLine2 License (optional dependency used with groovysh)

This product bundles the JLine2 jar, which is available under a
BSD License. For details, see licenses/jline2-license.txt.

javax.servlet.jsp-api License (for groovy-servlet module)

This product bundles the javax.servlet.jsp-api jar in binary form
which is available under the CDDL 1.1 license:
licenses/jsp-api-license.txt

JSR166y License (optionally used by the optional GParas dependency)

This product bundles the jsr166y jar (containing works from
the JSR-166 EG, Doug Lea, and Jason T. Greene) made available in
the public domain. For details, see licenses/jsr166y-license.txt.

JSR223 License

The following classes within this product:

org.codehaus.groovy.jsr223.GroovyCompiledScript
org.codehaus.groovy.jsr223.GroovyScriptEngineFactory
org.codehaus.groovy.jsr223.GroovyScriptEngineImpl

were derived from reference implementation files developed by Sun in

collaboration with the Groovy community. The reference implementation has a BSD-style license. Details can be found in: licenses/jsr223-license.txt

JUnit Licenses (optional dependencies when using Groovy for testing)

This product bundles the JUnit 4 jar, which is available under the Eclipse Public License v1.0. For details, see licenses/junit4-license.txt.

This product bundles several JUnit 5 jars, which are available under the Eclipse Public License v2.0. For details, see licenses/junit5-license.txt.

normalize.css License

The stylesheet.css file (originally normalize.css) is used by the groovydoc and docgenerator components for groovy-jdk/gapi documentation. It is made available under a MIT License:
licenses/normalize-stylesheet-license.txt

javax.servlet.jsp-api License (for groovy-servlet module)

This product bundles the javax.servlet-api jar in binary form which is available under the CDDL 1.0 license:
licenses/servlet-api-license.txt

XStream License (optional dependency when serializing AST as XML)

This product bundles the XStream jar, which is available under a "3-clause BSD" license. For details, see licenses/xstream-license.txt.

Apache Groovy
Copyright 2003-2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

This product includes/uses ANTLR2 (<http://www.antlr2.org/>)
developed by Terence Parr 1989-2006

This product includes/uses ANTLR4 (<https://github.com/antlr/antlr4>)
Copyright (c) 2012-2017 The ANTLR Project. All rights reserved.

This product bundles the javax.servlet.jsp-api jar in binary form which is available under the CDDL 1.1 license and for which the following copyright notice applies
Copyright (c) 1997-2013 Oracle and/or its affiliates. All rights reserved.

This product bundles the JUnit4 jar (junit.org) which is available under the terms of the Eclipse Public License v1.0

This product bundles several of the JUnit5 jars (junit.org) which are available under the terms of the Eclipse Public License v2.0

This product embeds the OpenBeans jar within its grooid jar artifacts
OpenBeans includes/uses files from Apache Harmony and the following notice applies
Copyright 2006, 2010 The Apache Software Foundation.
Portions of Apache Harmony were originally developed by Intel Corporation and are licensed to the Apache Software Foundation under the "Software Grant and Corporate Contribution License Agreement" and for which the following copyright notices apply
(C) Copyright 2005 Intel Corporation
(C) Copyright 2005-2006 Intel Corporation
(C) Copyright 2006 Intel Corporation

This product bundles the javax.servlet-api jar in binary form which is available under the CDDL 1.0 license and for which the following copyright notice applies
Copyright (c) 1997-2017 Oracle and/or its affiliates. All rights reserved.

This product bundles icons from the famfamfam.com silk icons set
<http://www.famfamfam.com/lab/icons/silk/>
Licensed under the Creative Commons Attribution Licence v2.5
<http://creativecommons.org/licenses/by/2.5/>

This convenience zip embeds Groovy's src and doc zips.
See also src/NOTICE and doc/NOTICE files for additional notice information.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity

on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one

of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a

result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

JSR223 License

The following classes within this product:

```
org.codehaus.groovy.jsr223.GroovyCompiledScript  
org.codehaus.groovy.jsr223.GroovyScriptEngineFactory  
org.codehaus.groovy.jsr223.GroovyScriptEngineImpl
```

were derived from reference implementation files developed by Sun in collaboration with the Groovy community. The reference implementation has a BSD-style license. Details can be found in: licenses/jsr223-license.txt
Copyright jQuery Foundation and other contributors, <https://jquery.org/>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

XStream License (optional dependency when serializing AST as XML)

This product bundles the XStream jar, which is available under a "3-clause BSD" license. For details, see licenses/xstream-license.txt.
ANTLR 4 License

[The "BSD 3-clause license"]

Copyright (c) 2012-2017 The ANTLR Project. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors

may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
NORMALIZE.CSS LICENSE

The following file is used with documentation:

`org/codehaus/groovy/tools/stylesheet.css`

Copyright (c) Nicolas Gallagher and Jonathan Neal

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

////////////////////////////////////

Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to you under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

////////////////////////////////////

= License

This work is licensed under the <http://www.apache.org/licenses/LICENSE-2.0> [Apache License, Version 2.0].
Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and

b) in the case of each subsequent Contributor:

i) changes to the Program, and

ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:

i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and

iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and

b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the

Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights

to a jury trial in any resulting litigation.
JSR166y License (optionally used by the optional GPar dependency)

This product bundles the jsr166y jar (containing works from the JSR-166 EG, Doug Lea, and Jason T. Greene) made available in the public domain. For details, see licenses/jsr166y-license.txt.
ANTLR 2 License

Antlr2 is released in the public domain.
See licenses/antlr2-license.txt for details.
Apache Groovy
Copyright 2003-2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

This product includes/uses ANTLR2 (<http://www.antlr2.org/>)
developed by Terence Parr 1989-2006

This product includes/uses ANTLR4 (<https://github.com/antlr/antlr4>)
Copyright (c) 2012-2017 The ANTLR Project. All rights reserved.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,

including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf

of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

ANTLR 2 License

Antlr2 is released in the public domain.
See licenses/antlr2-license.txt for details.

ANTLR 4 License

Antlr4 is released under a BSD 3-clause license.
See licenses/antlr4-license.txt for details.

ASM License

ASM uses a 3-clause BSD license. For details, see licenses/asm-license.txt.

Apache Groovy

Copyright 2003-2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

This product bundles icons from the famfamfam.com silk icons set
<http://www.famfamfam.com/lab/icons/silk/>
Licensed under the Creative Commons Attribution Licence v2.5
<http://creativecommons.org/licenses/by/2.5/>
Eclipse Public License - v 2.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE (AGREEMENT). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. Definitions

Contribution means:

- a) in the case of the initial Contributor, the initial content Distributed under this Agreement, and
- b) in the case of each subsequent Contributor:
 - i) changes to the Program, and
 - ii) additions to the Program; where such changes and/or additions to the Program originate from and are Distributed by that particular Contributor. A Contribution originates from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include changes or additions to the Program that are not Modified Works.

Contributor means any person or entity that Distributes the Program.

Licensed Patents mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

Program means the Contributions Distributed in accordance with this Agreement.

Recipient means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

Derivative Works shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

Modified Works shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

Distribute means the acts of a) distributing or b) making available in any manner that enables the transfer of a copy.

Source Code means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

Secondary License means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.

2. Grant of Rights

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

e) Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

3. Requirements

3.1 If a Contributor Distributes the Program in any form, then:

a) the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and

b) the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:

i) effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied,

including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and

iv) requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.

3.2 When the Program is Distributed as Source Code:

a) it must be made available under this Agreement, or if the Program (i) is combined with other material in a separate file or files made available under a Secondary License, and (ii) the initial Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available under the terms of such Secondary Licenses, and

b) a copy of this Agreement must be included with each copy of the Program.

3.3 Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability (notices) contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

4. Commercial Distribution

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor (Commercial Contributor) hereby agrees to defend and indemnify every other Contributor (Indemnified Contributor) against any losses, damages and costs (collectively Losses) arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. No Warranty

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN AS IS BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable

laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. Disclaimer of Liability

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. General

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.

Exhibit A - Form of Secondary Licenses Notice

This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: {name license(s), version(s), and exceptions or additional permissions here}.

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

1. Definitions.

1.1. Contributor means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable means the Covered Software in any form other than Source Code.

1.5. Initial Developer means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License means this document.

1.8. Licensable means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications means the Source Code and Executable form of any of the following: A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications; B. Any new file that contains any part of the Original Software or previous Modification; or C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims means any patent claim(s), now owned or hereafter acquired,

including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You (or Your) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant. Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof);

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License;

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant. Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code. Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications. The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices. You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms. You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or

liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions. You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipients rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works. You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions. Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions. You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions. When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any

references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. **DISCLAIMER OF WARRANTY.** COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. **TERMINATION.**

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. **LIMITATION OF LIABILITY.** UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK

STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTYS NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS. The Covered Software is a commercial item, as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of commercial computer software (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer software documentation as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS. This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdictions conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS. As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

For Covered Software in this distribution, this License shall be governed by the laws of the State of California (excluding conflict-of-law provisions).

Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

Apache Groovy

Copyright 2003-2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

Apache Groovy

Copyright 2003-2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

This product includes/uses ANTLR2 (<http://www.antlr2.org/>)
developed by Terence Parr 1989-2006

This product includes/uses ANTLR4 (<https://github.com/antlr/antlr4>)
Copyright (c) 2012-2017 The ANTLR Project. All rights reserved.

This product embeds the OpenBeans jar within its grooid jar artifacts
OpenBeans includes/uses files from Apache Harmony and the following notice applies
Copyright 2006, 2010 The Apache Software Foundation.

Portions of Apache Harmony were originally developed by Intel Corporation and are licensed to the Apache Software Foundation under the "Software Grant and Corporate Contribution License Agreement" and for which the following copyright notices apply

(C) Copyright 2005 Intel Corporation

(C) Copyright 2005-2006 Intel Corporation

(C) Copyright 2006 Intel Corporation

Apache Groovy

Copyright 2003-2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

This product embeds the OpenBeans jar within its grooid jar artifacts
OpenBeans includes/uses files from Apache Harmony and the following notice applies
Copyright 2006, 2010 The Apache Software Foundation.

Portions of Apache Harmony were originally developed by Intel Corporation and are licensed to the Apache Software Foundation under the "Software Grant and Corporate Contribution License Agreement" and for which the following copyright notices apply

(C) Copyright 2005 Intel Corporation

(C) Copyright 2005-2006 Intel Corporation

(C) Copyright 2006 Intel Corporation

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or

agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

NORMALIZE.CSS LICENSE

The following file is used with documentation:

org/codehaus/groovy/tools/stylesheet.css

Copyright (c) Nicolas Gallagher and Jonathan Neal

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License

Copyright (C) 2012-2015 Dan Allen, Ryan Waldron and the AsciiDoctor Project

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Hamcrest License (needed when using optional JUnit dependency)

This product bundles the Hamcrest jar, which is available under a BSD license. For details, see licenses/hamcrest-license.txt.

NORMALIZE.CSS LICENSE

The following file is used with documentation:

`org/codehaus/groovy/tools/groovydoc/gstringTemplates/topLevel/stylesheet.css`

Copyright (c) Nicolas Gallagher and Jonathan Neal

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

JUnit Licenses (optional dependencies when using Groovy for testing)

This product bundles the JUnit 4 jar, which is available under the Eclipse Public License v1.0. For details, see licenses/junit4-license.txt.

This product bundles several JUnit 5 jars, which are available under the Eclipse Public License v2.0. For details, see licenses/junit5-license.txt.

ASM License

ASM uses a 3-clause BSD license. For details, see licenses/asm-license.txt.

Asciidoc License

This product uses the style.css from asciidoctor.org within documentation. The file is available under the MIT License.

For details, see licenses/asciidoc-style-license.txt.

Copyright (c) 2006, Sun Microsystems, Inc.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the Sun Microsystems, Inc. nor the names of contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.138 asm-analysis 7.2

1.138.1 Available under license :

<OWNER> = Regents of the University of California

<ORGANIZATION> = University of California, Berkeley

<YEAR> = 1998

In the original BSD license, both occurrences of the phrase "COPYRIGHT HOLDERS AND CONTRIBUTORS" in the disclaimer read "REGENTS AND CONTRIBUTORS".

Here is the license template:

Copyright (c) <YEAR>, <OWNER>

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the <ORGANIZATION> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.139 micronaut-mqtt 2.3.0

1.139.1 Available under license :

Apache License

Version 2.0, January 2004

<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of

the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works

that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A

PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.140 jackson-datatype-jsr310 2.14.0

1.140.1 Available under license :

This copy of Jackson JSON processor streaming parser/generator is licensed under the Apache (Software) License, version 2.0 ("the License").

See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

<http://www.apache.org/licenses/LICENSE-2.0>

1.141 jopt-simple 5.0.4

1.141.1 Available under license :

The MIT License

Copyright (c) 2004-2015 Paul R. Holser, Jr.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.142 feign-core 8.18.0

1.142.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2015 Netflix, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1274700542_1645235077.45/0/feign-core-8-18-0-sources-jar/feign/Param.java
* /opt/cola/permits/1274700542_1645235077.45/0/feign-core-8-18-0-sources-jar/feign/QueryMap.java
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2014 Netflix, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1274700542_1645235077.45/0/feign-core-8-18-0-sources-
jar/feign/InvocationHandlerFactory.java
* /opt/cola/permits/1274700542_1645235077.45/0/feign-core-8-18-0-sources-
jar/feign/SynchronousMethodHandler.java
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2008 Google Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1274700542_1645235077.45/0/feign-core-8-18-0-sources-jar/feign/Types.java
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2013 Netflix, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1274700542_1645235077.45/0/feign-core-8-18-0-sources-jar/feign/Response.java
* /opt/cola/permits/1274700542_1645235077.45/0/feign-core-8-18-0-sources-jar/feign/codec/ErrorDecoder.java
* /opt/cola/permits/1274700542_1645235077.45/0/feign-core-8-18-0-sources-jar/feign/Contract.java
* /opt/cola/permits/1274700542_1645235077.45/0/feign-core-8-18-0-sources-jar/feign/codec/EncodeException.java
* /opt/cola/permits/1274700542_1645235077.45/0/feign-core-8-18-0-sources-jar/feign/Request.java
* /opt/cola/permits/1274700542_1645235077.45/0/feign-core-8-18-0-sources-jar/feign/RequestInterceptor.java
* /opt/cola/permits/1274700542_1645235077.45/0/feign-core-8-18-0-sources-jar/feign/auth/Base64.java
* /opt/cola/permits/1274700542_1645235077.45/0/feign-core-8-18-0-sources-jar/feign/Util.java
* /opt/cola/permits/1274700542_1645235077.45/0/feign-core-8-18-0-sources-jar/feign/RequestTemplate.java
```

```
* /opt/cola/permits/1274700542_1645235077.45/0/feign-core-8-18-0-sources-jar/feign/Client.java
* /opt/cola/permits/1274700542_1645235077.45/0/feign-core-8-18-0-sources-jar/feign/codec/StringDecoder.java
* /opt/cola/permits/1274700542_1645235077.45/0/feign-core-8-18-0-sources-jar/feign/codec/DecodeException.java
* /opt/cola/permits/1274700542_1645235077.45/0/feign-core-8-18-0-sources-jar/feign/FeignException.java
* /opt/cola/permits/1274700542_1645235077.45/0/feign-core-8-18-0-sources-jar/feign/Target.java
* /opt/cola/permits/1274700542_1645235077.45/0/feign-core-8-18-0-sources-
jar/feign/auth/BasicAuthRequestInterceptor.java
* /opt/cola/permits/1274700542_1645235077.45/0/feign-core-8-18-0-sources-jar/feign/RetryableException.java
* /opt/cola/permits/1274700542_1645235077.45/0/feign-core-8-18-0-sources-jar/feign/Logger.java
* /opt/cola/permits/1274700542_1645235077.45/0/feign-core-8-18-0-sources-jar/feign/ReflectiveFeign.java
* /opt/cola/permits/1274700542_1645235077.45/0/feign-core-8-18-0-sources-jar/feign/Retryer.java
* /opt/cola/permits/1274700542_1645235077.45/0/feign-core-8-18-0-sources-jar/feign/Feign.java
* /opt/cola/permits/1274700542_1645235077.45/0/feign-core-8-18-0-sources-jar/feign/MethodMetadata.java
No license file was found, but licenses were detected in source scan.
```

```
/*
* Copyright 2013 Netflix, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):

```
* /opt/cola/permits/1274700542_1645235077.45/0/feign-core-8-18-0-sources-jar/feign/codec/Decoder.java
* /opt/cola/permits/1274700542_1645235077.45/0/feign-core-8-18-0-sources-jar/feign/codec/Encoder.java
```

1.143 micronaut 3.7.2

1.143.1 Available under license :

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,

and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the

Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,

whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.144 apache-kafka 3.3.1

1.144.1 Available under license :

Apache Kafka

Copyright 2022 The Apache Software Foundation.

This product includes software developed at
The Apache Software Foundation (<https://www.apache.org/>).

This distribution has a binary dependency on jersey, which is available under the CDDL License. The source code of jersey can be found at <https://github.com/jersey/jersey/>.

This distribution has a binary test dependency on jqwik, which is available under the Eclipse Public License 2.0. The source code can be found at <https://github.com/jlink/jqwik>.

The streams-scala (streams/streams-scala) module was donated by Lightbend and the original code was copyrighted by them:

Copyright (C) 2018 Lightbend Inc. <<https://www.lightbend.com>>

Copyright (C) 2017-2018 Alexis Seigneurin.

This project contains the following code copied from Apache Hadoop:

`clients/src/main/java/org/apache/kafka/common/utils/PureJavaCrc32C.java`

Some portions of this file Copyright (c) 2004-2006 Intel Corporation and licensed under the BSD license.

This project contains the following code copied from Apache Hive:

`streams/src/main/java/org/apache/kafka/streams/state/internals/Murmur3.java`

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the

direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of

this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and

wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor

has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.145 apache-avro 1.11.1

1.145.1 Available under license :

Apache Avro

Copyright 2010-2019 The Apache Software Foundation

This product includes software developed at

The Apache Software Foundation (<https://www.apache.org/>).

NUnit license acknowledgement:

| Portions Copyright 2002-2012 Charlie Poole or Copyright 2002-2004 James
| W. Newkirk, Michael C. Two, Alexei A. Vorontsov or Copyright 2000-2002
| Philip A. Craig

Based upon the representations of upstream licensors, it is understood that portions of the mapreduce API included in the Java implementation are licensed from various contributors under one or more contributor license agreements to Odiago, Inc. and were then contributed by Odiago to Apache Avro, which has now made them available under the Apache 2.0 license. The original file header text is:

| Licensed to Odiago, Inc. under one or more contributor license
| agreements. See the NOTICE file distributed with this work for
| additional information regarding copyright ownership. Odiago, Inc.
| licenses this file to you under the Apache License, Version 2.0
| (the "License"); you may not use this file except in compliance
| with the License. You may obtain a copy of the License at
|
| <https://www.apache.org/licenses/LICENSE-2.0>
|
| Unless required by applicable law or agreed to in writing, software
| distributed under the License is distributed on an "AS IS" BASIS,
| WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
| implied. See the License for the specific language governing
| permissions and limitations under the License.

The Odiago NOTICE at the time of the contribution:

| This product includes software developed by Odiago, Inc.
| (<https://www.wibidata.com>).

Apache Ivy includes the following in its NOTICE file:

| Apache Ivy
| Copyright 2007-2010 The Apache Software Foundation
|
| This product includes software developed by
| The Apache Software Foundation (<https://www.apache.org/>).
|
| Portions of Ivy were originally developed by
| Jayasoft SARL (<http://www.jayasoft.fr/>)
| and are licensed to the Apache Software Foundation under the
| "Software Grant License Agreement"
|

| SSH and SFTP support is provided by the JCraft JSch package,
| which is open source software, available under
| the terms of a BSD style license.
| The original software and related information is available
| at <http://www.jcraft.com/jsch/>.

Apache Log4Net includes the following in its NOTICE file:

| Apache log4net
| Copyright 2004-2015 The Apache Software Foundation
|
| This product includes software developed at
| The Apache Software Foundation (<https://www.apache.org/>).

csharp reflect serializers were contributed by Pitney Bowes Inc.

| Copyright 2019 Pitney Bowes Inc.
| Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance
| with the License.
| You may obtain a copy of the License at <https://www.apache.org/licenses/LICENSE-2.0>.
| Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on
| an "AS IS" BASIS,
| WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
| See the License for the specific language governing permissions and limitations under the License.

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity

exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided

that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity,

or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<https://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

License for Guava classes included in this binary artifact:

Copyright: 2006-2015 The Guava Authors

License: <https://www.apache.org/licenses/LICENSE-2.0> (see above)

Apache Avro

Copyright 2010-2015 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<https://www.apache.org/>).

Based upon the representations of upstream licensors, it is understood that portions of the mapreduce API included in the Java implementation are licensed

from various contributors under one or more contributor license agreements to Odiago, Inc. and were then contributed by Odiago to Apache Avro, which has now made them available under the Apache 2.0 license. The original file header text is:

```
| Licensed to Odiago, Inc. under one or more contributor license
| agreements. See the NOTICE file distributed with this work for
| additional information regarding copyright ownership. Odiago, Inc.
| licenses this file to you under the Apache License, Version 2.0
| (the "License"); you may not use this file except in compliance
| with the License. You may obtain a copy of the License at
|
| https://www.apache.org/licenses/LICENSE-2.0
|
| Unless required by applicable law or agreed to in writing, software
| distributed under the License is distributed on an "AS IS" BASIS,
| WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
| implied. See the License for the specific language governing
| permissions and limitations under the License.
```

The Odiago NOTICE at the time of the contribution:

```
| This product includes software developed by Odiago, Inc.
| (https://www.wibidata.com).
```

The documentation contains the default Apache Forrest skin.
Apache Forrest includes the following in its NOTICE file:

```
| Apache Forrest
| Copyright 2002-2007 The Apache Software Foundation.
|
| This product includes software developed at
| The Apache Software Foundation (https://www.apache.org/).
|
| See also the file LICENSE.txt
|
| -----
| The purpose of this NOTICE.txt file is to contain notices that are
| required by the copyright owner and their license. Some of the
| accompanying products have an attribution requirement, so see below.
| Other accompanying products do not require attribution, so are not listed.
|
| -----
| This product includes software developed by the OpenSymphony Group
| http://www.opensymphony.com/
|
| This product includes software developed for project Krysalis
| http://www.krysalis.org/
```

|
| This product includes software developed by Andy Clark.
| <https://people.apache.org/~andyc/neko/>
|
| This product includes software developed by the ExoLab Project
| <https://www.exolab.org/>
|
| This product includes software developed by TouchGraph LLC
| <https://www.touchgraph.com/>
|
| This product includes software developed by Marc De Scheemaeker
| <http://nanoxml.cyberelf.be/>
|
| This product includes software developed by the ANTLR project
| <https://wwwantlr.org/>
|
| This product includes software developed by Chaperon
| <http://chaperon.sourceforge.net/>
|
| This product includes software developed by Sal Mangano (included in the XSLT Cookbook published by O'Reilly)
| <https://www.oreilly.com/catalog/xsltckbk/>
|
| This product includes software developed by The Werken Company.
| <http://jaxen.werken.com/>
|
| This product includes software developed by the jfor project
| <http://www.jfor.org/>

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or

otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents

of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<https://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

License for the m4 macros used by the C++ implementation:

Files:

* lang/c++/m4/m4_ax_boost_system.m4

Copyright (c) 2008 Thomas Porschberg <thomas@randspringer.de>

Copyright (c) 2008 Michael Tindal

Copyright (c) 2008 Daniel Casimiro <dan.casimiro@gmail.com>

* lang/c++/m4/m4_ax_boost_asio.m4

Copyright (c) 2008 Thomas Porschberg <thomas@randspringer.de>

Copyright (c) 2008 Pete Greenwell <pete@mu.org>

* lang/c++/m4/m4_ax_boost_filesystem.m4

Copyright (c) 2009 Thomas Porschberg <thomas@randspringer.de>

Copyright (c) 2009 Michael Tindal

Copyright (c) 2009 Roman Rybalko <libtorrent@romanr.info>

* lang/c++/m4/m4_ax_boost_thread.m4

Copyright (c) 2009 Thomas Porschberg <thomas@randspringer.de>

Copyright (c) 2009 Michael Tindal

* lang/c++/m4/m4_ax_boost_regex.m4

Copyright (c) 2008 Thomas Porschberg <thomas@randspringer.de>

Copyright (c) 2008 Michael Tindal

* lang/c++/m4/m4_ax_boost_base.m4

Copyright (c) 2008 Thomas Porschberg <thomas@randspringer.de>

License text:

| Copying and distribution of this file, with or without modification, are
| permitted in any medium without royalty provided the copyright notice
| and this notice are preserved. This file is offered as-is, without any
| warranty.

License for the AVRO_BOOT_NO_TRAIT code in the C++ implementation:

File: lang/c++/api/Boost.hh

| Boost Software License - Version 1.0 - August 17th, 2003

|

| Permission is hereby granted, free of charge, to any person or organization
| obtaining a copy of the software and accompanying documentation covered by
| this license (the "Software") to use, reproduce, display, distribute,
| execute, and transmit the Software, and to prepare derivative works of the
| Software, and to permit third-parties to whom the Software is furnished to
| do so, all subject to the following:

|

| The copyright notices in the Software and this entire statement, including
| the above license grant, this restriction and the following disclaimer,
| must be included in all copies of the Software, in whole or in part, and
| all derivative works of the Software, unless such copies or derivative
| works are solely in the form of machine-executable object code generated by
| a source language processor.

|

| THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
| IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
| FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT
| SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE
| FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE,
| ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
| DEALINGS IN THE SOFTWARE.

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or

agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<https://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

License for portions of idl.jj in the Java compiler implementation:

Portions of idl.jj were modeled after the example Java 1.5 parser included with JavaCC. For those portions:

Copyright (c) 2006, Sun Microsystems, Inc.
All rights reserved.

| Redistribution and use in source and binary forms, with or without
| modification, are permitted provided that the following conditions are met:

- | * Redistributions of source code must retain the above copyright notice,
| this list of conditions and the following disclaimer.
- | * Redistributions in binary form must reproduce the above copyright
| notice, this list of conditions and the following disclaimer in the
| documentation and/or other materials provided with the distribution.
- | * Neither the name of the Sun Microsystems, Inc. nor the names of its
| contributors may be used to endorse or promote products derived from
| this software without specific prior written permission.

| THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
| AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
| IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
| ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
| LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
| CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
| SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
| INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
| CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
| ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
| THE POSSIBILITY OF SUCH DAMAGE.

License for Jackson, included in this binary artifact:

Copyright: 2007-2015 Tatu Saloranta and other contributors

Home page: <http://jackson.codehaus.org/>

License: <https://www.apache.org/licenses/LICENSE-2.0.txt> (see above)

License for SLF4J, include in this binary artifact:

Copyright (c) 2004-2013 QOS.ch

All rights reserved.

Home page: <https://www.slf4j.org/>

License: <https://slf4j.org/license.html> (MIT license)

SLF4J license text (MIT):

| Permission is hereby granted, free of charge, to any person obtaining
| a copy of this software and associated documentation files (the
| "Software"), to deal in the Software without restriction, including
| without limitation the rights to use, copy, modify, merge, publish,
| distribute, sublicense, and/or sell copies of the Software, and to
| permit persons to whom the Software is furnished to do so, subject to
| the following conditions:

|
| The above copyright notice and this permission notice shall be
| included in all copies or substantial portions of the Software.

|
| THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
| EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
| MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
| NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
| LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
| OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
| WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

License for snappy-java, included in this binary artifact:

Copyright: 2011 Taro L. Saito and other contributors

Home page: <http://www.xerial.org/>

License: <https://www.apache.org/licenses/LICENSE-2.0.txt> (see above)

License for Apache Commons Compress, included in this binary artifact:

Copyright: 2004-2015 The Apache Software Foundation

Home page: <https://commons.apache.org/compress/>

License: <https://www.apache.org/licenses/LICENSE-2.0.txt> (see above)

Commons Compress includes files derived from the LZMA SDK, version 9.20 (C/ and
CPP/7zip/), in the package org.apache.commons.compress.archivers.sevenz:

| LZMA SDK is placed in the public domain. (<https://www.7-zip.org/sdk.html>)

License for xz compression, included in this binary artifact:

Home page: <https://tukaani.org/xz/java.html>

| This Java implementation of XZ has been put into the public domain, thus you
| can do whatever you want with it. All the files in the package have been
| written by Lasse Collin, but some files are heavily based on public domain code
| written by Igor Pavlov.

License for Apache Commons Lang, included in this binary artifact:

Copyright: 2002-2014 The Apache Software Foundation

Home page: <https://commons.apache.org/lang/>

License: <https://www.apache.org/licenses/LICENSE-2.0.txt> (see above)

Commons Lang includes software from the Spring Framework, under the
Apache License 2.0:

* `StringUtils.containsWhitespace()`

License for Apache Velocity, included in this binary artifact:

Copyright: 2000-2015 The Apache Software Foundation

Home page: <https://velocity.apache.org/>

License: <https://www.apache.org/licenses/LICENSE-2.0.txt> (see above)

License for Apache Commons Collections, included in this binary artifact:

Copyright: 2001-2015 The Apache Software Foundation

Home page: <https://commons.apache.org/proper/commons-collections/>

License: <https://www.apache.org/licenses/LICENSE-2.0.txt> (see above)

License for Jetty, included in this binary artifact:

Copyright: 1995-2015 Mort Bay Consulting Pty Ltd.

Home page: <https://eclipse.org/jetty/licenses.php>

License: <https://www.apache.org/licenses/LICENSE-2.0.txt> (see above)

License for Netty, included in this binary artifact:

Copyright: 2011-2013 The Netty Project

Home page: <https://netty.io/>

License: <https://www.apache.org/licenses/LICENSE-2.0.txt> (see above)

Netty contains the following code (copied from its NOTICE file with licenses added inline):

| This product contains the extensions to Java Collections Framework which has
| been derived from the works by JSR-166 EG, Doug Lea, and Jason T. Greene:

|

| * LICENSE:

|| The person or persons who have associated work with this document (the
|| "Dedicator" or "Certifier") hereby either (a) certifies that, to the best of
|| his knowledge, the work of authorship identified is in the public domain of
|| the country from which the work is published, or (b) hereby dedicates whatever
|| copyright the dedicators holds in the work of authorship identified below (the
|| "Work") to the public domain. A certifier, moreover, dedicates any copyright
|| interest he may have in the associated work, and for these purposes, is
|| described as a "dedicator" below.

||

|| A certifier has taken reasonable steps to verify the copyright status of this
|| work. Certifier recognizes that his good faith efforts may not shield him from
|| liability if in fact the work certified is not in the public domain.

||

|| Dedicator makes this dedication for the benefit of the public at large and to
|| the detriment of the Dedicator's heirs and successors. Dedicator intends this
|| dedication to be an overt act of relinquishment in perpetuate of all present
|| and future rights under copyright law, whether vested or contingent, in the
|| Work. Dedicator understands that such relinquishment of all rights includes
|| the relinquishment of all rights to enforce (by lawsuit or otherwise) those
|| copyrights in the Work.

||

|| Dedicator recognizes that, once placed in the public domain, the Work may be
|| freely reproduced, distributed, transmitted, used, modified, built upon, or
|| otherwise exploited by anyone for any purpose, commercial or non-commercial,
|| and in any way, including by methods that have not yet been invented or
|| conceived.

| * HOMEPAGE:

| * <http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/>

| * <http://viewvc.jboss.org/cgi-bin/viewvc.cgi/jboss-cache/experimental/jsr166/>

|

| This product contains a modified version of Robert Harder's Public Domain
| Base64 Encoder and Decoder, which can be obtained at:

|

| * LICENSE: public domain (see JSR-166 license above)

| * HOMEPAGE:

| * <http://iharder.sourceforge.net/current/java/base64/>

|

| This product contains a modified version of 'JZlib', a re-implementation of

| zlib in pure Java, which can be obtained at:

|

| * LICENSE:

|| Copyright (c) 2000,2001,2002,2003,2004 ymnk, JCraft,Inc. All rights reserved.

||

|| Redistribution and use in source and binary forms, with or without

|| modification, are permitted provided that the following conditions are met:

||

|| 1. Redistributions of source code must retain the above copyright notice,

|| this list of conditions and the following disclaimer.

||

|| 2. Redistributions in binary form must reproduce the above copyright

|| notice, this list of conditions and the following disclaimer in

|| the documentation and/or other materials provided with the distribution.

||

|| 3. The names of the authors may not be used to endorse or promote products

|| derived from this software without specific prior written permission.

||

|| THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES,

|| INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND

|| FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL JCRAFT,

|| INC. OR ANY CONTRIBUTORS TO THIS SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT,

|| INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT

|| LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA,

|| OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF

|| LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING

|| NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE,

|| EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

| * HOMEPAGE:

| * <http://www.jcraft.com/jzlib/>

License for the javax.servlet API, included in this binary artifact:

Copyright (c) 2003-2004 The Apache Software Foundation

License: <https://www.apache.org/licenses/LICENSE-2.0.txt> (see above)

Source: <http://grepcode.com/project/repo1.maven.org/maven2/javax.servlet/servlet-api/>

License for Apache Commons Codec, included in this binary artifact:

Copyright: 2002-2015 The Apache Software Foundation

Home page: <https://commons.apache.org/proper/commons-codec/>

License: <https://www.apache.org/licenses/LICENSE-2.0.txt> (see above)

License for Apache Commons CLI, included in this binary artifact:

Copyright: 2001-2015 The Apache Software Foundation
Home page: <https://commons.apache.org/proper/commons-cli/>
License: <https://www.apache.org/licenses/LICENSE-2.0.txt> (see above)

License for Apache Commons Logging, included in this binary artifact:

Copyright: 2002-2014 The Apache Software Foundation
Home page: <https://commons.apache.org/proper/commons-logging/>
License: <https://www.apache.org/licenses/LICENSE-2.0.txt> (see above)

License for Apache Commons HttpClient, included in this binary artifact:

Copyright: 1999-2005 The Apache Software Foundation
License: <https://www.apache.org/licenses/LICENSE-2.0.txt> (see above)

License for Apache Hadoop, included in this binary artifact:

Copyright: 2001-2015 The Apache Software Foundation
Home page: <https://commons.apache.org/proper/commons-collections/>
License: <https://www.apache.org/licenses/LICENSE-2.0.txt> (see above)

Hadoop contains the following code (from its LICENSE file):

The org.apache.hadoop.util.bloom.* classes:
| Copyright (c) 2005, European Commission project OneLab under contract
| 034819 (<http://www.one-lab.org>)
| All rights reserved.
| Redistribution and use in source and binary forms, with or
| without modification, are permitted provided that the following
| conditions are met:
| - Redistributions of source code must retain the above copyright
| notice, this list of conditions and the following disclaimer.
| - Redistributions in binary form must reproduce the above copyright
| notice, this list of conditions and the following disclaimer in
| the documentation and/or other materials provided with the distribution.
| - Neither the name of the University Catholique de Louvain - UCL
| nor the names of its contributors may be used to endorse or
| promote products derived from this software without specific prior
| written permission.
|
| THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
| "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
| LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
| FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
| COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,

| INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
| BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
| LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
| CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
| LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN
| ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
| POSSIBILITY OF SUCH DAMAGE.

License for Google Guava, included in this binary artifact:

Copyright: 2007-2015 The Guava Authors
Home page: <https://github.com/google/guava>
License: <https://www.apache.org/licenses/LICENSE-2.0> (see above)

License for Apache Commons Math, included in this binary artifact:

Copyright: 2001-2015 The Apache Software Foundation
Home page: <https://commons.apache.org/proper/commons-math/>
License: <https://www.apache.org/licenses/LICENSE-2.0> (see above)

Commons Math includes other works under licenses compatible with the
Apache Software License:

| APACHE COMMONS MATH DERIVATIVE WORKS:

|
| The Apache commons-math library includes a number of subcomponents
| whose implementation is derived from original sources written
| in C or Fortran. License terms of the original sources
| are reproduced below.

|
| =====
| For the lmdcr, lmpar and qrsolv Fortran routine from minpack and translated in
| the LevenbergMarquardtOptimizer class in package
| org.apache.commons.math3.optimization.general
| Original source copyright and license statement:

|
| Minpack Copyright Notice (1999) University of Chicago. All rights reserved

|
| Redistribution and use in source and binary forms, with or
| without modification, are permitted provided that the
| following conditions are met:

|
| 1. Redistributions of source code must retain the above
| copyright notice, this list of conditions and the following
| disclaimer.

|
| 2. Redistributions in binary form must reproduce the above

| copyright notice, this list of conditions and the following
| disclaimer in the documentation and/or other materials
| provided with the distribution.

|

| 3. The end-user documentation included with the
| redistribution, if any, must include the following
| acknowledgment:

|

| "This product includes software developed by the
| University of Chicago, as Operator of Argonne National
| Laboratory.

|

| Alternately, this acknowledgment may appear in the software
| itself, if and wherever such third-party acknowledgments
| normally appear.

|

| 4. WARRANTY DISCLAIMER. THE SOFTWARE IS SUPPLIED "AS IS"
| WITHOUT WARRANTY OF ANY KIND. THE COPYRIGHT HOLDER, THE
| UNITED STATES, THE UNITED STATES DEPARTMENT OF ENERGY, AND
| THEIR EMPLOYEES: (1) DISCLAIM ANY WARRANTIES, EXPRESS OR
| IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES
| OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE
| OR NON-INFRINGEMENT, (2) DO NOT ASSUME ANY LEGAL LIABILITY
| OR RESPONSIBILITY FOR THE ACCURACY, COMPLETENESS, OR
| USEFULNESS OF THE SOFTWARE, (3) DO NOT REPRESENT THAT USE OF
| THE SOFTWARE WOULD NOT INFRINGE PRIVATELY OWNED RIGHTS, (4)
| DO NOT WARRANT THAT THE SOFTWARE WILL FUNCTION
| UNINTERRUPTED, THAT IT IS ERROR-FREE OR THAT ANY ERRORS WILL
| BE CORRECTED.

|

| 5. LIMITATION OF LIABILITY. IN NO EVENT WILL THE COPYRIGHT
| HOLDER, THE UNITED STATES, THE UNITED STATES DEPARTMENT OF
| ENERGY, OR THEIR EMPLOYEES: BE LIABLE FOR ANY INDIRECT,
| INCIDENTAL, CONSEQUENTIAL, SPECIAL OR PUNITIVE DAMAGES OF
| ANY KIND OR NATURE, INCLUDING BUT NOT LIMITED TO LOSS OF
| PROFITS OR LOSS OF DATA, FOR ANY REASON WHATSOEVER, WHETHER
| SUCH LIABILITY IS ASSERTED ON THE BASIS OF CONTRACT, TORT
| (INCLUDING NEGLIGENCE OR STRICT LIABILITY), OR OTHERWISE,
| EVEN IF ANY OF SAID PARTIES HAS BEEN WARNED OF THE
| POSSIBILITY OF SUCH LOSS OR DAMAGES.

| =====

|

| Copyright and license statement for the odex Fortran routine developed by
| E. Hairer and G. Wanner and translated in GraggBulirschStoerIntegrator class
| in package org.apache.commons.math3.ode.nonstiff:

|

|

| Copyright (c) 2004, Ernst Hairer

|
| Redistribution and use in source and binary forms, with or without
| modification, are permitted provided that the following conditions are
| met:

|
| - Redistributions of source code must retain the above copyright
| notice, this list of conditions and the following disclaimer.

|
| - Redistributions in binary form must reproduce the above copyright
| notice, this list of conditions and the following disclaimer in the
| documentation and/or other materials provided with the distribution.

|
| THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
| IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED
| TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
| PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR
| CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
| EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
| PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
| PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
| LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
| NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
| SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

| =====

|
| Copyright and license statement for the original lapack fortran routines
| translated in EigenDecompositionImpl class in package
| org.apache.commons.math3.linear:

|
| Copyright (c) 1992-2008 The University of Tennessee. All rights reserved.

|
| \$COPYRIGHT\$

|
| Additional copyrights may follow

|
| \$HEADERS\$

|
| Redistribution and use in source and binary forms, with or without
| modification, are permitted provided that the following conditions are
| met:

|
| - Redistributions of source code must retain the above copyright
| notice, this list of conditions and the following disclaimer.

|
| - Redistributions in binary form must reproduce the above copyright
| notice, this list of conditions and the following disclaimer listed
| in this license in the documentation and/or other materials
| provided with the distribution.

| - Neither the name of the copyright holders nor the names of its
| contributors may be used to endorse or promote products derived from
| this software without specific prior written permission.

| THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
| "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
| LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
| A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
| OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
| SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
| LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
| DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
| THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
| (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
| OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

| =====
| Copyright and license statement for the original Mersenne twister C
| routines translated in MersenneTwister class in package
| org.apache.commons.math3.random:

| Copyright (C) 1997 - 2002, Makoto Matsumoto and Takuji Nishimura,
| All rights reserved.

| Redistribution and use in source and binary forms, with or without
| modification, are permitted provided that the following conditions
| are met:

- | 1. Redistributions of source code must retain the above copyright
| notice, this list of conditions and the following disclaimer.
- | 2. Redistributions in binary form must reproduce the above copyright
| notice, this list of conditions and the following disclaimer in the
| documentation and/or other materials provided with the distribution.
- | 3. The names of its contributors may not be used to endorse or promote
| products derived from this software without specific prior written
| permission.

| THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
| "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
| LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
| A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
| CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
| EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
| PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
| PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF

| LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
| NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
| SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

|
|=====

| The class "org.apache.commons.math3.exception.util.LocalizedFormatsTest" is
| an adapted version of "OrekitMessagesTest" test class for the Orekit library
| The "org.apache.commons.math3.analysis.interpolation.HermiteInterpolator"
| has been imported from the Orekit space flight dynamics library.

|
| Th Orekit library is described at:
| <https://www.orekit.org/forged/projects/orekit>
| The original files are distributed under the terms of the Apache 2 license
| which is: Copyright 2010 CS Communication & Systemes

License for XMLenc, included in this binary artifact:

Copyright 2003-2011, Ernst de Haan
All rights reserved.

| Redistribution and use in source and binary forms, with or without
| modification, are permitted provided that the following conditions are met:

- |
- | 1. Redistributions of source code must retain the above copyright notice, this
| list of conditions and the following disclaimer.
 - | 2. Redistributions in binary form must reproduce the above copyright notice,
| this list of conditions and the following disclaimer in the documentation
| and/or other materials provided with the distribution.

| THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
| AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
| IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
| DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
| FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
| DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
| SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
| CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
| OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
| OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

License for Apache Commons IO, included in this binary artifact:

Copyright: 2002-2015 The Apache Software Foundation
Home page: <https://commons.apache.org/proper/io/>

License: <https://www.apache.org/licenses/LICENSE-2.0> (see above)

License for Apache Commons Net, included in this binary artifact:

Copyright: 2001-2015 The Apache Software Foundation

Home page: <https://commons.apache.org/proper/commons-net/>

License: <https://www.apache.org/licenses/LICENSE-2.0> (see above)

License for Apache Log4j, included in this binary artifact:

Copyright: 1999-2015 The Apache Software Foundation

Home page: <https://logging.apache.org/log4j/>

License: <https://www.apache.org/licenses/LICENSE-2.0> (see above)

License for Apache Commons Configuration, included in this binary artifact:

Copyright: 2001-2015 The Apache Software Foundation

Home page: <https://commons.apache.org/proper/commons-configuration/>

License: <https://www.apache.org/licenses/LICENSE-2.0> (see above)

License for Apache Commons Digester, included in this binary artifact:

Copyright: 2001-2015 The Apache Software Foundation

License: <https://www.apache.org/licenses/LICENSE-2.0> (see above)

License for Apache Commons Beanutils, included in this binary artifact:

Copyright: 2000-2015 The Apache Software Foundation

License: <https://www.apache.org/licenses/LICENSE-2.0> (see above)

License for Google Protocol Buffers, included in this binary artifact:

Copyright 2014, Google Inc. All rights reserved.

| Redistribution and use in source and binary forms, with or without
| modification, are permitted provided that the following conditions are
| met:

| * Redistributions of source code must retain the above copyright
| notice, this list of conditions and the following disclaimer.

| * Redistributions in binary form must reproduce the above
| copyright notice, this list of conditions and the following disclaimer

| in the documentation and/or other materials provided with the
| distribution.

| * Neither the name of Google Inc. nor the names of its
| contributors may be used to endorse or promote products derived from
| this software without specific prior written permission.

| THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
| "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
| LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
| A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
| OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
| SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
| LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
| DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
| THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
| (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
| OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

| Code generated by the Protocol Buffer compiler is owned by the owner
| of the input file used when generating it. This code is not
| standalone and requires a support library to be linked with it. This
| support library is itself covered by the above license.

License for Apache HttpClient, included in this binary artifact:

Copyright: 1999-2015 The Apache Software Foundation
License: <https://www.apache.org/licenses/LICENSE-2.0> (see above)

HttpClient contains the following data under the terms of the MPL:

| This project includes Public Suffix List copied from
| <https://publicsuffix.org/list/effective_tld_names.dat>
| licensed under the terms of the Mozilla Public License, v. 2.0
|
| Full license text: META-INF/mpl-2.0.text

License for Apache Directory, included in this binary artifact:

Copyright: 2003-2015 The Apache Software Foundation
License: <https://www.apache.org/licenses/LICENSE-2.0> (see above)

Apache Directory includes other works under licenses compatible with the
Apache Software License:

|
| The OpenLDAP Public License

| Version 2.8, 17 August 2003

|
| Redistribution and use of this software and associated documentation
| ("Software"), with or without modification, are permitted provided
| that the following conditions are met:

- | 1. Redistributions in source form must retain copyright statements
| and notices,
- | 2. Redistributions in binary form must reproduce applicable copyright
| statements and notices, this list of conditions, and the following
| disclaimer in the documentation and/or other materials provided
| with the distribution, and
- | 3. Redistributions must contain a verbatim copy of this document.

| The OpenLDAP Foundation may revise this license from time to time.
| Each revision is distinguished by a version number. You may use
| this Software under terms of this license revision or under the
| terms of any subsequent revision of the license.

| THIS SOFTWARE IS PROVIDED BY THE OPENLDAP FOUNDATION AND ITS
| CONTRIBUTORS ``AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES,
| INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY
| AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT
| SHALL THE OPENLDAP FOUNDATION, ITS CONTRIBUTORS, OR THE AUTHOR(S)
| OR OWNER(S) OF THE SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT,
| INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
| BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
| LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
| CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
| LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN
| ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
| POSSIBILITY OF SUCH DAMAGE.

| The names of the authors and copyright holders must not be used in
| advertising or otherwise to promote the sale, use or other dealing
| in this Software without specific, written prior permission. Title
| to copyright in this Software shall at all times remain with copyright
| holders.

| OpenLDAP is a registered trademark of the OpenLDAP Foundation.

| Copyright 1999-2003 The OpenLDAP Foundation, Redwood City,
| California, USA. All Rights Reserved. Permission to copy and
| distribute verbatim copies of this document is granted.

|-----

| Copyright (c) 2000 - 2011 The Legion Of The Bouncy Castle (<https://www.bouncycastle.org>)

| Permission is hereby granted, free of charge, to any person obtaining a
| copy of this software and associated documentation files (the "Software"),
| to deal in the Software without restriction, including without limitation
| the rights to use, copy, modify, merge, publish, distribute, sublicense,
| and/or sell copies of the Software, and to permit persons to whom the
| Software is furnished to do so, subject to the following conditions:

| The above copyright notice and this permission notice shall be included in
| all copies or substantial portions of the Software.

| THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
| IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
| FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
| AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
| LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
| OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
| THE SOFTWARE.

=====
slf4j 1.7.10 license:

| Copyright (c) 2004-2013 QOS.ch
| All rights reserved.

| Permission is hereby granted, free of charge, to any person obtaining
| a copy of this software and associated documentation files (the
| "Software"), to deal in the Software without restriction, including
| without limitation the rights to use, copy, modify, merge, publish,
| distribute, sublicense, and/or sell copies of the Software, and to
| permit persons to whom the Software is furnished to do so, subject to
| the following conditions:

| The above copyright notice and this permission notice shall be
| included in all copies or substantial portions of the Software.

| THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
| EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
| MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
| NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
| LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
| OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
| WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=====

| For the AVL Set code : <http://bobah.net/d4d/source-code/misc/java-avl-tree>

| -----

| Copyright 2001-2014 Vladimir Lysyy

| Licensed under the Apache License, Version 2.0 (the "License");
| you may not use this source code except in compliance with the License.
| You may obtain a copy of the License at

| <https://www.apache.org/licenses/LICENSE-2.0>

| Unless required by applicable law or agreed to in writing, software
| distributed under the License is distributed on an "AS IS" BASIS,
| WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
| See the License for the specific language governing permissions and
| limitations under the License.

License for the JSR-305 annotations, included in this binary artifact:

Copyright: 2011-2015 Stephen Connolly, Greg Lucas

License: <https://www.apache.org/licenses/LICENSE-2.0> (see above)

License for Apache ZooKeeper, included in this binary artifact:

Copyright: 2009-2015 The Apache Software Foundation

License: <https://www.apache.org/licenses/LICENSE-2.0> (see above)

License for Jersey, included in this binary artifact:

Copyright (c) 2015 Oracle and/or its affiliates.

All rights reserved.

License: CDDL 1.1: META-INF/cddl-1.1.text

Source: <https://github.com/jersey/jersey-1.x-old>

License for LevelDB JNI, included in this binary artifact:

Copyright (c) 2011 FuseSource Corp. All rights reserved.

| Redistribution and use in source and binary forms, with or without
| modification, are permitted provided that the following conditions are
| met:

| * Redistributions of source code must retain the above copyright
| notice, this list of conditions and the following disclaimer.

| * Redistributions in binary form must reproduce the above

| copyright notice, this list of conditions and the following disclaimer
| in the documentation and/or other materials provided with the
| distribution.

| * Neither the name of FuseSource Corp. nor the names of its
| contributors may be used to endorse or promote products derived from
| this software without specific prior written permission.

| THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
| "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
| LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
| A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
| OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
| SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
| LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
| DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
| THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
| (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
| OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

License for iq80 LevelDB Java API, included in this binary artifact:

Copyright 2011 Dain Sundstrom <dain@iq80.com>
Copyright 2011 FuseSource Corp. <http://fusesource.com>
License: <https://www.apache.org/licenses/LICENSE-2.0> (see above)

License for jquery and jquery-ui, included in this binary artifact:

License: The MIT License (MIT): <https://tldrlegal.com/license/mit-license>
Home page: <https://jquery.org/license/>

Copyright (c) <year> <copyright holders>

| Permission is hereby granted, free of charge, to any person obtaining a copy of
| this software and associated documentation files (the "Software"), to deal in
| the Software without restriction, including without limitation the rights to
| use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
| of the Software, and to permit persons to whom the Software is furnished to do
| so, subject to the following conditions:

| The above copyright notice and this permission notice shall be included in all
| copies or substantial portions of the Software.

| THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
| IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
| FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
| AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER

| LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
| OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
| SOFTWARE.

License for the javax.xml.bind JAXB API, included in this binary artifact:

Copyright (c) 2004-2010 Oracle and/or its affiliates.

All rights reserved.

License: CDDL 1.0: META-INF/cddl-1.0.text

Source: <http://www.greppcode.com/project/repo1.maven.org/maven2/javax.xml.bind/jaxb-api/>

License for the javax.xml.stream StAX API, included in this binary artifact:

Copyright (c) 2004-2006 Oracle and/or its affiliates.

All rights reserved.

License: CDDL 1.0: META-INF/cddl-1.0.text

Source: <http://greppcode.com/project/repo1.maven.org/maven2/javax.xml.stream/stax-api/>

License for the javax.activation API, included in this binary artifact:

Copyright (c) 2004-2006 Oracle and/or its affiliates.

All rights reserved.

License: CDDL 1.0: META-INF/cddl-1.0.text

Source: <http://greppcode.com/project/repo1.maven.org/maven2/javax.activation/activation/>

License for the javax.ws.rs API, included in this binary artifact:

Copyright (c) 1996-2015, Oracle Corporation and/or its affiliates.

All rights reserved.

License: CDDL 1.1: META-INF/cddl-1.1.text

Source: <http://greppcode.com/project/repo1.maven.org/maven2/javax.ws.rs/javax.ws.rs-api/>

License for JOpt Simple, included in this binary artifact:

Copyright (c) 2004-2015 Paul R. Holser, Jr.

| Permission is hereby granted, free of charge, to any person obtaining
| a copy of this software and associated documentation files (the
| "Software"), to deal in the Software without restriction, including
| without limitation the rights to use, copy, modify, merge, publish,
| distribute, sublicense, and/or sell copies of the Software, and to
| permit persons to whom the Software is furnished to do so, subject to
| the following conditions:

|
| The above copyright notice and this permission notice shall be
| included in all copies or substantial portions of the Software.

|
| THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
| EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
| MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
| NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
| LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
| OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
| WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

License for SLF4J API, included in this binary artifact:

Copyright (c) 2004-2013 QOS.ch
All rights reserved.

| Permission is hereby granted, free of charge, to any person obtaining
| a copy of this software and associated documentation files (the
| "Software"), to deal in the Software without restriction, including
| without limitation the rights to use, copy, modify, merge, publish,
| distribute, sublicense, and/or sell copies of the Software, and to
| permit persons to whom the Software is furnished to do so, subject to
| the following conditions:

|
| The above copyright notice and this permission notice shall be
| included in all copies or substantial portions of the Software.

|
| THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
| EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
| MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
| NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
| LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
| OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
| WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

License for Guava classes included in this binary artifact:

Copyright: 2006-2015 The Guava Authors
License: <https://www.apache.org/licenses/LICENSE-2.0> (see above)

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and

attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the

appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<https://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

License for the Jansson C JSON parser used in the C implementation:

Copyright (c) 2009-2011 Petri Lehtinen <petri@digip.org>

Some files include an additional copyright notice:

* lang/c/jansson/src/pack_unpack.c

Copyright (c) 2011 Graeme Smecher <graeme.smecher@mail.mcgill.ca>

* lang/c/jansson/test/suites/api/test_unpack.c

Copyright (c) 2011 Graeme Smecher <graeme.smecher@mail.mcgill.ca>

* lang/c/jansson/src/memory.c

Copyright (c) 2011 Basile Starynkevitch <basile@starynkevitch.net>

| Permission is hereby granted, free of charge, to any person obtaining a copy
| of this software and associated documentation files (the "Software"), to deal
| in the Software without restriction, including without limitation the rights
| to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
| copies of the Software, and to permit persons to whom the Software is
| furnished to do so, subject to the following conditions:

|
| The above copyright notice and this permission notice shall be included in
| all copies or substantial portions of the Software.

|
| THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
| IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
| FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
| AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
| LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
| OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
| THE SOFTWARE.

License for msinttypes.h and msstdint.h used in the C implementation:

Source from:

<https://code.google.com/p/msinttypes/downloads/detail?name=msinttypes-r26.zip>

Copyright (c) 2006-2008 Alexander Chemeris

| Redistribution and use in source and binary forms, with or without
| modification, are permitted provided that the following conditions are met:

|
| 1. Redistributions of source code must retain the above copyright notice,
| this list of conditions and the following disclaimer.

| 2. Redistributions in binary form must reproduce the above copyright
| notice, this list of conditions and the following disclaimer in the
| documentation and/or other materials provided with the distribution.

| 3. The name of the author may be used to endorse or promote products
| derived from this software without specific prior written permission.

| THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED
| WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
| MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO
| EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
| SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
| PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS;
| OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
| WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
| OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF
| ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

License for st.c and st.h used in the C implementation:

| This is a public domain general purpose hash table package written by
| Peter Moore @ UCB.

License for Diredt API for Microsoft Visual Studio used in the C implementation:

Source from:

<http://www.softagalleria.net/download/diredt/diredt-1.11.zip>

Copyright (C) 2006 Toni Ronkko

| Permission is hereby granted, free of charge, to any person obtaining
| a copy of this software and associated documentation files (the
| ``Software"), to deal in the Software without restriction, including
| without limitation the rights to use, copy, modify, merge, publish,
| distribute, sublicense, and/or sell copies of the Software, and to
| permit persons to whom the Software is furnished to do so, subject to
| the following conditions:

| The above copyright notice and this permission notice shall be included
| in all copies or substantial portions of the Software.

| THE SOFTWARE IS PROVIDED ``AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
| OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
| MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
| IN NO EVENT SHALL TONI RONKKO BE LIABLE FOR ANY CLAIM, DAMAGES OR
| OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,

| ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
| OTHER DEALINGS IN THE SOFTWARE.

License for ivy-2.2.0.jar used in the python implementation:

Apache License version 2.0 (see above)

License for pyAntTasks-1.3.jar used in the python implementation:

Apache License version 2.0 (see above)

License for NUnit binary included with the C# implementation:

File: nunit.framework.dll

| NUnit License

|

| Copyright 2002-2015 Charlie Poole

| Copyright 2002-2004 James W. Newkirk, Michael C. Two, Alexei A. Vorontsov

| Copyright 2000-2002 Philip A. Craig

|

| This software is provided 'as-is', without any express or implied warranty. In
| no event will the authors be held liable for any damages arising from the use
| of this software.

|

| Permission is granted to anyone to use this software for any purpose, including
| commercial applications, and to alter it and redistribute it freely, subject to
| the following restrictions:

|

| The origin of this software must not be misrepresented; you must not claim that
| you wrote the original software. If you use this software in a product, an
| acknowledgment (see the following) in the product documentation is required.

|

| Portions Copyright 2002-2012 Charlie Poole or Copyright 2002-2004 James W.
| Newkirk, Michael C. Two, Alexei A. Vorontsov or Copyright 2000-2002 Philip A.
| Craig

|

| Altered source versions must be plainly marked as such, and must not be
| misrepresented as being the original software.

|

| This notice may not be removed or altered from any source distribution.

| License Note

|

| This license is based on the open source zlib/libpng license. The idea was to
| keep the license as simple as possible to encourage use of NUnit in free and
| commercial applications and libraries, but to keep the source code together and

| to give credit to the NUnit contributors for their efforts. While this license
| allows shipping NUnit in source and binary form, if shipping a NUnit variant is
| the sole purpose of your product, please let us know.

License for the Json.NET binary included with the C# implementation:

File: Newtonsoft.Json.dll

Copyright (c) 2007 James Newton-King

| Permission is hereby granted, free of charge, to any person obtaining
| a copy of this software and associated documentation files (the
| "Software"), to deal in the Software without restriction, including
| without limitation the rights to use, copy, modify, merge, publish,
| distribute, sublicense, and/or sell copies of the Software, and to
| permit persons to whom the Software is furnished to do so, subject to
| the following conditions:

|
| The above copyright notice and this permission notice shall be
| included in all copies or substantial portions of the Software.

|
| THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
| EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
| MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
| NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
| LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
| OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
| WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

License for the Castle Core binary included with the C# implementation:

File: Castle.Core.dll

Copyright (c) 2004-2015 Castle Project

License: Apache License version 2.0 (see above)

URL: <https://opensource.org/licenses/Apache-2.0>

License for the log4net binary included with the C# implementation:

File: log4net.dll

Copyright 2004-2015 The Apache Software Foundation.

License: Apache License version 2.0 (see above)

License for the m4 macros used by the C++ implementation:

Files:

* lang/c++/m4/m4_ax_boost_system.m4
Copyright (c) 2008 Thomas Porschberg <thomas@randspringer.de>
Copyright (c) 2008 Michael Tindal
Copyright (c) 2008 Daniel Casimiro <dan.casimiro@gmail.com>
* lang/c++/m4/m4_ax_boost_asio.m4
Copyright (c) 2008 Thomas Porschberg <thomas@randspringer.de>
Copyright (c) 2008 Pete Greenwell <pete@mu.org>
* lang/c++/m4/m4_ax_boost_filesystem.m4
Copyright (c) 2009 Thomas Porschberg <thomas@randspringer.de>
Copyright (c) 2009 Michael Tindal
Copyright (c) 2009 Roman Rybalko <libtorrent@romanr.info>
* lang/c++/m4/m4_ax_boost_thread.m4
Copyright (c) 2009 Thomas Porschberg <thomas@randspringer.de>
Copyright (c) 2009 Michael Tindal
* lang/c++/m4/m4_ax_boost_regex.m4
Copyright (c) 2008 Thomas Porschberg <thomas@randspringer.de>
Copyright (c) 2008 Michael Tindal
* lang/c++/m4/m4_ax_boost_base.m4
Copyright (c) 2008 Thomas Porschberg <thomas@randspringer.de>

License text:

| Copying and distribution of this file, with or without modification, are
| permitted in any medium without royalty provided the copyright notice
| and this notice are preserved. This file is offered as-is, without any
| warranty.

License for the AVRO_BOOT_NO_TRAIT code in the C++ implementation:

File: lang/c++/api/Boost.hh

| Boost Software License - Version 1.0 - August 17th, 2003
|
| Permission is hereby granted, free of charge, to any person or organization
| obtaining a copy of the software and accompanying documentation covered by
| this license (the "Software") to use, reproduce, display, distribute,
| execute, and transmit the Software, and to prepare derivative works of the
| Software, and to permit third-parties to whom the Software is furnished to
| do so, all subject to the following:
|
| The copyright notices in the Software and this entire statement, including
| the above license grant, this restriction and the following disclaimer,
| must be included in all copies of the Software, in whole or in part, and
| all derivative works of the Software, unless such copies or derivative
| works are solely in the form of machine-executable object code generated by
| a source language processor.
|

| THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
| IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
| FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT
| SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE
| FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE,
| ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
| DEALINGS IN THE SOFTWARE.

License for jquery.tipsy.js, tipsy.js, and tipsy.css used by the Java IPC implementation:

Copyright (c) 2008 Jason Frame (jason@onehackoranother.com)

| Permission is hereby granted, free of charge, to any person obtaining a copy
| of this software and associated documentation files (the "Software"), to deal
| in the Software without restriction, including without limitation the rights
| to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
| copies of the Software, and to permit persons to whom the Software is
| furnished to do so, subject to the following conditions:

| The above copyright notice and this permission notice shall be included in
| all copies or substantial portions of the Software.

| THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
| IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
| FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE
| AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
| LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
| OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
| THE SOFTWARE.

License for protovis-r3.2.js used by the Java IPC implementation:

Copyright (c) 2010, Stanford Visualization Group
All rights reserved.

| Redistribution and use in source and binary forms, with or without modification,
| are permitted provided that the following conditions are met:

| * Redistributions of source code must retain the above copyright notice,
| this list of conditions and the following disclaimer.

| * Redistributions in binary form must reproduce the above copyright notice,
| this list of conditions and the following disclaimer in the documentation
| and/or other materials provided with the distribution.

| * Neither the name of Stanford University nor the names of its contributors

| may be used to endorse or promote products derived from this software
| without specific prior written permission.

| THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
| ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
| WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
| DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR
| ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
| (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
| LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON
| ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
| (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
| SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

License for g.Raphael 0.4.1 used by the Java IPC implementation:

Copyright (c) 2009 Dmitry Baranovskiy (<http://g.raphaeljs.com>)
Licensed under the MIT (<https://www.opensource.org/licenses/mit-license.php>) license.

License for jQuery v1.4.2 used by the Java IPC implementation:

Copyright 2010, John Resig
Dual licensed under the MIT or GPL Version 2 licenses.
<https://jquery.org/license>

jQuery includes Sizzle.js
<https://sizzlejs.com/>
Copyright 2010, The Dojo Foundation
Released under the MIT, BSD, and GPL Licenses.

Both are included under the terms of the MIT license:

| Permission is hereby granted, free of charge, to any person obtaining a copy
| of this software and associated documentation files (the "Software"), to deal
| in the Software without restriction, including without limitation the rights
| to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
| copies of the Software, and to permit persons to whom the Software is
| furnished to do so, subject to the following conditions:

| The above copyright notice and this permission notice shall be included in
| all copies or substantial portions of the Software.

| THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
| IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
| FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
| AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER

| LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
| OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
| THE SOFTWARE.

License for portions of idl.jj in the Java compiler implementation:

Portions of idl.jj were modeled after the example Java 1.5
parser included with JavaCC. For those portions:

Copyright (c) 2006, Sun Microsystems, Inc.
All rights reserved.

| Redistribution and use in source and binary forms, with or without
| modification, are permitted provided that the following conditions are met:

- | * Redistributions of source code must retain the above copyright notice,
| this list of conditions and the following disclaimer.
- | * Redistributions in binary form must reproduce the above copyright
| notice, this list of conditions and the following disclaimer in the
| documentation and/or other materials provided with the distribution.
- | * Neither the name of the Sun Microsystems, Inc. nor the names of its
| contributors may be used to endorse or promote products derived from
| this software without specific prior written permission.

| THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
| AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
| IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
| ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
| LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
| CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
| SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
| INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
| CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
| ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
| THE POSSIBILITY OF SUCH DAMAGE.

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise

designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. **Grant of Copyright License.** Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. **Grant of Patent License.** Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. **Redistribution.** You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must

include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly

negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<https://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Mozilla Public License
Version 2.0

1. Definitions

1.1. Contributor

means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

1.2. Contributor Version

means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution.

1.3. Contribution

means Covered Software of a particular Contributor.

1.4. Covered Software

means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

1.5. Incompatible With Secondary Licenses

means

a. that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or

b. that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

1.6. Executable Form

means any form of the work other than Source Code Form.

1.7. Larger Work

means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

1.8. License

means this document.

1.9. Licensable

means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

1.10. Modifications

means any of the following:

a. any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or

b. any new file in Source Code Form that contains any Covered Software.

1.11. Patent Claims of a Contributor

means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

1.12. Secondary License

means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

1.13. Source Code Form

means the form of the work preferred for making modifications.

1.14. You (or Your)

means an individual or a legal entity exercising rights under this License. For legal entities, You includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

- a. under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and
- b. under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

- a. for any code that a Contributor has removed from Covered Software; or
- b. for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or
- c. under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

- a. such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and
- b. You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support,

indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

6. Disclaimer of Warranty

Covered Software is provided under this License on an as is basis, without warranty of any kind, either expressed, implied, or statutory, including,

without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

7. Limitation of Liability

Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version

number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <https://mozilla.org/MPL/2.0/>.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - Incompatible With Secondary Licenses Notice

This Source Code Form is Incompatible With Secondary Licenses, as defined by the Mozilla Public License, v. 2.0.

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems,

and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<https://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and

limitations under the License.

License for the Json.NET binary included with the C# implementation:
File: Newtonsoft.Json.dll

Copyright (c) 2007 James Newton-King

| Permission is hereby granted, free of charge, to any person obtaining
| a copy of this software and associated documentation files (the
| "Software"), to deal in the Software without restriction, including
| without limitation the rights to use, copy, modify, merge, publish,
| distribute, sublicense, and/or sell copies of the Software, and to
| permit persons to whom the Software is furnished to do so, subject to
| the following conditions:

|
| The above copyright notice and this permission notice shall be
| included in all copies or substantial portions of the Software.

|
| THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
| EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
| MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
| NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
| LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
| OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
| WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

License for the Castle Core binary included with the C# implementation:
File: Castle.Core.dll

Copyright (c) 2004-2015 Castle Project

License: Apache License version 2.0 (see above)
URL: <https://opensource.org/licenses/Apache-2.0>

License for the log4net binary included with the C# implementation:
File: log4net.dll

Copyright 2004-2015 The Apache Software Foundation.

License: Apache License version 2.0 (see above)
Apache Avro
Copyright 2010 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<https://www.apache.org/>).

Apache Avro
Copyright 2010-2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<https://www.apache.org/>).

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications

represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without

modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. **Trademarks.** This License does not grant permission to use the trade

names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier

identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<https://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

License for the AVRO_BOOT_NO_TRAIT code in the C++ implementation:

File: lang/c++/api/Boost.hh

| Boost Software License - Version 1.0 - August 17th, 2003

|

| Permission is hereby granted, free of charge, to any person or organization
| obtaining a copy of the software and accompanying documentation covered by
| this license (the "Software") to use, reproduce, display, distribute,
| execute, and transmit the Software, and to prepare derivative works of the
| Software, and to permit third-parties to whom the Software is furnished to
| do so, all subject to the following:

|

| The copyright notices in the Software and this entire statement, including
| the above license grant, this restriction and the following disclaimer,
| must be included in all copies of the Software, in whole or in part, and
| all derivative works of the Software, unless such copies or derivative
| works are solely in the form of machine-executable object code generated by
| a source language processor.

|

| THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
| IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
| FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT
| SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE
| FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE,
| ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
| DEALINGS IN THE SOFTWARE.

License for jQuery v1.7.1 used in the C# documentation

Copyright 2010-2011, John Resig

Dual licensed under the MIT or GPL Version 2 licenses.

<https://jquery.org/license>

jQuery includes Sizzle.js

<https://sizzlejs.com/>

Copyright 2010-2011, The Dojo Foundation

Released under the MIT, BSD, and GPL Licenses.

Both are included under the terms of the MIT license:

| Permission is hereby granted, free of charge, to any person obtaining a copy
| of this software and associated documentation files (the "Software"), to deal
| in the Software without restriction, including without limitation the rights
| to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
| copies of the Software, and to permit persons to whom the Software is
| furnished to do so, subject to the following conditions:

|
| The above copyright notice and this permission notice shall be included in
| all copies or substantial portions of the Software.

|
| THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
| IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
| FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
| AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
| LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
| OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
| THE SOFTWARE.

License for portions of idl.jj in the Java compiler implementation:

Portions of idl.jj were modeled after the example Java 1.5
parser included with JavaCC. For those portions:

Copyright (c) 2006, Sun Microsystems, Inc.
All rights reserved.

| Redistribution and use in source and binary forms, with or without
| modification, are permitted provided that the following conditions are met:

- |
- | * Redistributions of source code must retain the above copyright notice,
| this list of conditions and the following disclaimer.
 - | * Redistributions in binary form must reproduce the above copyright
| notice, this list of conditions and the following disclaimer in the
| documentation and/or other materials provided with the distribution.
 - | * Neither the name of the Sun Microsystems, Inc. nor the names of its
| contributors may be used to endorse or promote products derived from
| this software without specific prior written permission.

|
| THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
| AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
| IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
| ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
| LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
| CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
| SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
| INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
| CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
| ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
| THE POSSIBILITY OF SUCH DAMAGE.

License for prototype.js included in the Avro documentation:

Prototype JavaScript framework, version 1.4.0_pre4
(c) 2005 Sam Stephenson <sam@conio.net>

| Prototype is freely distributable under the terms of an MIT-style license.

|
| For details, see the Prototype web site: <http://prototype.conio.net/>

For a copy of the MIT license text, see above.

License for Apache Forrest (skin), included in the Avro documentation:

Copyright: 2009-2015 The Apache Software Foundation
License: <https://www.apache.org/licenses/LICENSE-2.0> (see above)

License for Doxygen-generated documentation for the C++ and C# implementations:

Copyright 1997-2015 by Dimitri van Heesch.

| Doxygen license

|
| Permission to use, copy, modify, and distribute this software and its
| documentation under the terms of the GNU General Public License is hereby
| granted. No representations are made about the suitability of this software for
| any purpose. It is provided "as is" without express or implied warranty. See
| the GNU General Public License for more details.

|
| Documents produced by doxygen are derivative works derived from the input
| used in their production; they are not affected by this license.

Apache Avro
Copyright 2010-2015 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<https://www.apache.org/>).
Apache Avro
Copyright 2011-2015 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<https://www.apache.org/>).

title: "License"
linkTitle: "License"
weight: 3
manualLink: <https://www.apache.org/licenses/>

<!--

Licensed to the Apache Software Foundation (ASF) under one
or more contributor license agreements. See the NOTICE file
distributed with this work for additional information
regarding copyright ownership. The ASF licenses this file
to you under the Apache License, Version 2.0 (the
"License"); you may not use this file except in compliance
with the License. You may obtain a copy of the License at

<https://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing,
software distributed under the License is distributed on an
"AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
KIND, either express or implied. See the License for the
specific language governing permissions and limitations
under the License.

-->

Apache Avro project is licensed under [Apache Software License 2.0](<https://www.apache.org/licenses/LICENSE-2.0>)
Apache Avro
Copyright 2010-2015 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<https://www.apache.org/>).

Based upon the representations of upstream licensors, it is understood that
portions of the mapreduce API included in the Java implementation are licensed
from various contributors under one or more contributor license agreements to
Odiago, Inc. and were then contributed by Odiago to Apache Avro, which has now

made them available under the Apache 2.0 license. The original file header text is:

| Licensed to Odiago, Inc. under one or more contributor license
| agreements. See the NOTICE file distributed with this work for
| additional information regarding copyright ownership. Odiago, Inc.
| licenses this file to you under the Apache License, Version 2.0
| (the "License"); you may not use this file except in compliance
| with the License. You may obtain a copy of the License at
|
| <https://www.apache.org/licenses/LICENSE-2.0>
|
| Unless required by applicable law or agreed to in writing, software
| distributed under the License is distributed on an "AS IS" BASIS,
| WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
| implied. See the License for the specific language governing
| permissions and limitations under the License.

The Odiago NOTICE at the time of the contribution:

| This product includes software developed by Odiago, Inc.
| (<https://www.wibidata.com>).

Apache Commons compress includes the following in its NOTICE file:

| Apache Commons Compress
| Copyright 2002-2014 The Apache Software Foundation
|
| This product includes software developed at
| The Apache Software Foundation (<https://www.apache.org/>).
|
| The files in the package org.apache.commons.compress.archivers.sevenz
| were derived from the LZMA SDK, version 9.20 (C/ and CPP/7zip/),
| which has been placed in the public domain:
|
| "LZMA SDK is placed in the public domain." (<https://www.7-zip.org/sdk.html>)

Apache Commons codec includes the following in its NOTICE file:

| Apache Commons Codec
| Copyright 2002-2015 The Apache Software Foundation
|
| This product includes software developed at
| The Apache Software Foundation (<https://www.apache.org/>).
|
| The content of package org.apache.commons.codec.language.bm has been translated
| from the original php source code available at <https://stevemorse.org/phoneticinfo.htm>
| with permission from the original authors.

| Original source copyright:
| Copyright (c) 2008 Alexander Beider & Stephen P. Morse.

Apache Commons lang includes the following in its NOTICE file:

| Apache Commons Lang
| Copyright 2001-2011 The Apache Software Foundation
|
| This product includes software developed by
| The Apache Software Foundation (<https://www.apache.org/>).

Apache Velocity includes the following in its NOTICE file:

| Apache Velocity
| Copyright (C) 2000-2007 The Apache Software Foundation
|
| This product includes software developed at
| The Apache Software Foundation (<https://www.apache.org/>).

Apache Commons collections includes the following in its NOTICE file:

| Apache Commons Collections
| Copyright 2001-2008 The Apache Software Foundation
|
| This product includes software developed by
| The Apache Software Foundation (<https://www.apache.org/>).

Apache Commons math includes the following in its NOTICE file:

| =====
|
| The BracketFinder (package org.apache.commons.math3.optimization.univariate)
| and PowellOptimizer (package org.apache.commons.math3.optimization.general)
| classes are based on the Python code in module "optimize.py" (version 0.5)
| developed by Travis E. Oliphant for the SciPy library (<https://www.scipy.org/>)
| Copyright 2003-2009 SciPy Developers.

| =====
|
| The LinearConstraint, LinearObjectiveFunction, LinearOptimizer,
| Relationship, SimplexSolver and SimplexTableau classes in package
| org.apache.commons.math3.optimization.linear include software developed by
| Benjamin McCann (<https://www.benmccann.com>) and distributed with
| the following copyright: Copyright 2009 Google Inc.

| =====
|
| This product includes software developed by the
| University of Chicago, as Operator of Argonne National
| Laboratory.

| The LevenbergMarquardtOptimizer class in package
| org.apache.commons.math3.optimization.general includes software
| translated from the lmdcr, lmpar and qrsolv Fortran routines
| from the Minpack package
| Minpack Copyright Notice (1999) University of Chicago. All rights reserved

=====
|
| The GraggBulirschStoerIntegrator class in package
| org.apache.commons.math3.ode.nonstiff includes software translated
| from the odex Fortran routine developed by E. Hairer and G. Wanner.
| Original source copyright:
| Copyright (c) 2004, Ernst Hairer

=====
|
| The EigenDecompositionImpl class in package
| org.apache.commons.math3.linear includes software translated
| from some LAPACK Fortran routines. Original source copyright:
| Copyright (c) 1992-2008 The University of Tennessee. All rights reserved.

=====
|
| The MersenneTwister class in package org.apache.commons.math3.random
| includes software translated from the 2002-01-26 version of
| the Mersenne-Twister generator written in C by Makoto Matsumoto and Takuji
| Nishimura. Original source copyright:
| Copyright (C) 1997 - 2002, Makoto Matsumoto and Takuji Nishimura,
| All rights reserved

=====
|
| The LocalizedFormatsTest class in the unit tests is an adapted version of
| the OrekitMessagesTest class from the orekit library distributed under the
| terms of the Apache 2 licence. Original source copyright:
| Copyright 2010 CS Systmes d'Information

=====
|
| The HermiteInterpolator class and its corresponding test have been imported from
| the orekit library distributed under the terms of the Apache 2 licence. Original
| source copyright:
| Copyright 2010-2012 CS Systmes d'Information

=====
|
| The creation of the package "o.a.c.m.analysis.integration.gauss" was inspired
| by an original code donated by Sbastien Brisard.

=====
|
| The complete text of licenses and disclaimers associated with the the original
| sources enumerated above at the time of code translation are in the LICENSE.txt
| file.

Jetty 6.1.26 includes the following in its NOTICE file:

```
=====
| Jetty Web Container
| Copyright 1995-2009 Mort Bay Consulting Pty Ltd
|=====
|
| The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd
| unless otherwise noted. It is licensed under the apache 2.0
| license.
|
| The javax.servlet package used by Jetty is copyright
| Sun Microsystems, Inc and Apache Software Foundation. It is
| distributed under the Common Development and Distribution License.
| You can obtain a copy of the license at
| https://glassfish.dev.java.net/public/CDDLv1.0.html.
|
| The UnixCrypt.java code ~Implements the one way cryptography used by
| Unix systems for simple password protection. Copyright 1996 Aki Yoshida,
| modified April 2001 by Iris Van den Broeke, Daniel Deville.
| Permission to use, copy, modify and distribute UnixCrypt
| for non-commercial or commercial purposes and without fee is
| granted provided that the copyright notice appears in all copies.
|
| The default JSP implementation is provided by the Glassfish JSP engine
| from project Glassfish https://glassfish.dev.java.net. Copyright 2005
| Sun Microsystems, Inc. and portions Copyright Apache Software Foundation.
|
| Some portions of the code are Copyright:
| 2006 Tim Venum
| 1999 Jason Gilbert.
|
| The jboss integration module contains some LGPL code.
| [JBOSS INTEGRATION IS NOT INCLUDED IN AVRO TOOLS.]
|
| The win32 Java Service Wrapper (v3.2.3) is Copyright (c) 1999, 2006
| Tanuki Software, Inc. and 2001 Silver Egg Technology. It is
| covered by an open license which is viewable at
| http://svn.codehaus.org/jetty/jetty/branches/jetty-6.1/extras/win32service/LICENSE.txt
| [WIN32 WRAPPER IS NOT INCLUDED IN AVRO TOOLS.]
```

Netty 3.5.13.Final includes the following in its NOTICE file:

```
| The Netty Project
|=====
|
| Please visit the Netty web site for more information:
```

| * <https://netty.io/>

| Copyright 2011 The Netty Project

| The Netty Project licenses this file to you under the Apache License,
| version 2.0 (the "License"); you may not use this file except in compliance
| with the License. You may obtain a copy of the License at:

| <https://www.apache.org/licenses/LICENSE-2.0>

| Unless required by applicable law or agreed to in writing, software
| distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
| WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
| License for the specific language governing permissions and limitations
| under the License.

| Also, please refer to each LICENSE.<component>.txt file, which is located in
| the 'license' directory of the distribution file, for the license terms of the
| components that this product depends on.

| -----
| This product contains the extensions to Java Collections Framework which has
| been derived from the works by JSR-166 EG, Doug Lea, and Jason T. Greene:

| * LICENSE:
| * <license/LICENSE.jsr166y.txt> (Public Domain)
| * HOMEPAGE:
| * <http://gee.cs.oswego.edu/cgi-bin/viewcvcs.cgi/jsr166/>
| * <http://viewvc.jboss.org/cgi-bin/viewvc.cgi/jboss/cache/experimental/jsr166/>

| This product contains a modified version of Robert Harder's Public Domain
| Base64 Encoder and Decoder, which can be obtained at:

| * LICENSE:
| * <license/LICENSE.base64.txt> (Public Domain)
| * HOMEPAGE:
| * <http://iharder.sourceforge.net/current/java/base64/>

| This product contains a modified version of 'JZlib', a re-implementation of
| zlib in pure Java, which can be obtained at:

| * LICENSE:
| * <license/LICENSE.jzlib.txt> (BSD Style License)
| * HOMEPAGE:
| * <http://www.jcraft.com/jzlib/>

| This product optionally depends on 'Protocol Buffers', Google's data

| interchange format, which can be obtained at:

|
| * LICENSE:
| * license/LICENSE.protobuf.txt (New BSD License)
| * HOMEPAGE:
| * <https://code.google.com/p/protobuf/>

| This product optionally depends on 'SLF4J', a simple logging facade for Java,
| which can be obtained at:

|
| * LICENSE:
| * license/LICENSE.slf4j.txt (MIT License)
| * HOMEPAGE:
| * <https://www.slf4j.org/>

| This product optionally depends on 'Apache Commons Logging', a logging
| framework, which can be obtained at:

|
| * LICENSE:
| * license/LICENSE.commons-logging.txt (Apache License 2.0)
| * HOMEPAGE:
| * <https://commons.apache.org/logging/>

| This product optionally depends on 'Apache Log4J', a logging framework,
| which can be obtained at:

|
| * LICENSE:
| * license/LICENSE.log4j.txt (Apache License 2.0)
| * HOMEPAGE:
| * <https://logging.apache.org/log4j/>

| This product optionally depends on 'JBoss Logging', a logging framework,
| which can be obtained at:

|
| * LICENSE:
| * license/LICENSE.jboss-logging.txt (GNU LGPL 2.1)
| * HOMEPAGE:
| * <https://anonsvn.jboss.org/repos/common/common-logging-spi/>

| [JBoss Logging is NOT included in Avro Tools.]

| This product optionally depends on 'Apache Felix', an open source OSGi
| framework implementation, which can be obtained at:

|
| * LICENSE:
| * license/LICENSE.felix.txt (Apache License 2.0)
| * HOMEPAGE:
| * <https://felix.apache.org/>

|
| [FELIX IS NOT INCLUDED IN AVRO TOOLS.]
|
| This product optionally depends on 'Webbit', a Java event based
| WebSocket and HTTP server:
|
| * LICENSE:
| * license/LICENSE.webbit.txt (BSD License)
| * HOMEPAGE:
| * <https://github.com/joewalnes/webbit>
|
| [WEBBIT IS NOT INCLUDED IN AVRO TOOLS.]

Apache Commons CLI includes the following in its NOTICE file:

| Apache Commons CLI
| Copyright 2001-2009 The Apache Software Foundation
|
| This product includes software developed by
| The Apache Software Foundation (<https://www.apache.org/>).

Apache Commons logging includes the following in its NOTICE file:

| Apache Commons Logging
| Copyright 2003-2007 The Apache Software Foundation
|
| This product includes software developed by
| The Apache Software Foundation (<https://www.apache.org/>).

Apache Commons HttpClient includes the following in its NOTICE file:

| Apache Jakarta HttpClient
| Copyright 1999-2007 The Apache Software Foundation
|
| This product includes software developed by
| The Apache Software Foundation (<https://www.apache.org/>).

Apache Hadoop includes the following in its NOTICE file:

| This product includes software developed by The Apache Software
| Foundation (<https://www.apache.org/>).

Apache Commons IO includes the following in its NOTICE file:

| Apache Commons IO
| Copyright 2002-2012 The Apache Software Foundation
|
| This product includes software developed by

| The Apache Software Foundation (<https://www.apache.org/>).

Apache Commons Net includes the following in its NOTICE file:

| Apache Commons Net

| Copyright 2001-2012 The Apache Software Foundation

|

| This product includes software developed by

| The Apache Software Foundation (<https://www.apache.org/>).

Apache Log4j includes the following in its NOTICE file:

| Apache log4j

| Copyright 2010 The Apache Software Foundation

|

| This product includes software developed at

| The Apache Software Foundation (<https://www.apache.org/>).

Apache Commons configuration includes the following in its NOTICE file:

| Apache Commons Configuration

| Copyright 2001-2008 The Apache Software Foundation

|

| This product includes software developed by

| The Apache Software Foundation (<https://www.apache.org/>).

Apache Commons digester includes the following in its NOTICE file:

| Apache Jakarta Commons Digester

| Copyright 2001-2006 The Apache Software Foundation

|

| This product includes software developed by

| The Apache Software Foundation (<https://www.apache.org/>).

Apache Commons beanutils includes the following in its NOTICE file:

| Apache Commons BeanUtils

| Copyright 2000-2008 The Apache Software Foundation

|

| This product includes software developed by

| The Apache Software Foundation (<https://www.apache.org/>).

Apache Directory includes the following in its NOTICE file:

| ApacheDS

| Copyright 2003-2015 The Apache Software Foundation

|

| This product includes software developed at

| The Apache Software Foundation (<https://www.apache.org/>).

Apache Zookeeper includes the following in its NOTICE file:

| Apache ZooKeeper

| Copyright 2009-2014 The Apache Software Foundation

|

| This product includes software developed at

| The Apache Software Foundation (<https://www.apache.org/>).

Apache Avro

Copyright 2010-2015 The Apache Software Foundation

This product includes software developed at

The Apache Software Foundation (<https://www.apache.org/>).

This library was original developed by Yann Kerherve with the following copyright notice:

| Copyright (C) 2010 Yann Kerherve. All rights reserved.

Apache Avro

Copyright 2010 The Apache Software Foundation

This product includes software developed at

The Apache Software Foundation (<https://www.apache.org/>).

Based upon the representations of upstream licensors, it is understood that portions of the mapreduce API included in the Java implementation are licensed from various contributors under one or more contributor license agreements to Odiago, Inc. and were then contributed by Odiago to Apache Avro, which has now made them available under the Apache 2.0 license. The original file header text is:

| Licensed to Odiago, Inc. under one or more contributor license

| agreements. See the NOTICE file distributed with this work for

| additional information regarding copyright ownership. Odiago, Inc.

| licenses this file to you under the Apache License, Version 2.0

| (the "License"); you may not use this file except in compliance

| with the License. You may obtain a copy of the License at

|

| <https://www.apache.org/licenses/LICENSE-2.0>

|

| Unless required by applicable law or agreed to in writing, software

| distributed under the License is distributed on an "AS IS" BASIS,

| WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or

| implied. See the License for the specific language governing

| permissions and limitations under the License.

The Odiago NOTICE at the time of the contribution:

| This product includes software developed by Odiago, Inc.
| (<https://www.wibidata.com>).
Apache Avro
Copyright 2010-2022 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<https://www.apache.org/>).

Based upon the representations of upstream licensors, it is understood that portions of the mapreduce API included in the Java implementation are licensed from various contributors under one or more contributor license agreements to Odiago, Inc. and were then contributed by Odiago to Apache Avro, which has now made them available under the Apache 2.0 license. The original file header text is:

| Licensed to Odiago, Inc. under one or more contributor license
| agreements. See the NOTICE file distributed with this work for
| additional information regarding copyright ownership. Odiago, Inc.
| licenses this file to you under the Apache License, Version 2.0
| (the "License"); you may not use this file except in compliance
| with the License. You may obtain a copy of the License at
|
| <https://www.apache.org/licenses/LICENSE-2.0>
|
| Unless required by applicable law or agreed to in writing, software
| distributed under the License is distributed on an "AS IS" BASIS,
| WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
| implied. See the License for the specific language governing
| permissions and limitations under the License.

|-----
| This product includes software developed by The Docsy Authors.
| (<https://www.docsy.dev/>).
|

| This product includes software developed at
| The Apache Software Foundation (<https://www.apache.org/>).
|

| See also the file LICENSE.txt
|

|-----
| The purpose of this NOTICE.txt file is to contain notices that are
| required by the copyright owner and their license. Some of the
| accompanying products have an attribution requirement, so see below.
| Other accompanying products do not require attribution, so are not listed.
|

|-----

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or

agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<https://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

License for msinttypes.h and msstdint.h used in the C implementation:

Source from:

<https://code.google.com/p/msinttypes/downloads/detail?name=msinttypes-r26.zip>

Copyright (c) 2006-2008 Alexander Chemeris

| Redistribution and use in source and binary forms, with or without
| modification, are permitted provided that the following conditions are met:

- |
- | 1. Redistributions of source code must retain the above copyright notice,
| this list of conditions and the following disclaimer.
 - |
 - | 2. Redistributions in binary form must reproduce the above copyright
| notice, this list of conditions and the following disclaimer in the
| documentation and/or other materials provided with the distribution.
 - |
 - | 3. The name of the author may be used to endorse or promote products
| derived from this software without specific prior written permission.
 - |

| THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED
| WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
| MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO
| EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
| SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
| PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS;
| OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
| WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
| OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF
| ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

License for st.c and st.h used in the C implementation:

| This is a public domain general purpose hash table package written by
| Peter Moore @ UCB.

License for Diredt API for Microsoft Visual Studio used in the C implementation:

Source from:

<http://www.softagalleria.net/download/diredt/diredt-1.11.zip>

Copyright (C) 2006 Toni Ronkko

| Permission is hereby granted, free of charge, to any person obtaining
| a copy of this software and associated documentation files (the
| ``Software"), to deal in the Software without restriction, including
| without limitation the rights to use, copy, modify, merge, publish,
| distribute, sublicense, and/or sell copies of the Software, and to
| permit persons to whom the Software is furnished to do so, subject to
| the following conditions:

|
| The above copyright notice and this permission notice shall be included
| in all copies or substantial portions of the Software.

|
| THE SOFTWARE IS PROVIDED ``AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
| OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
| MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
| IN NO EVENT SHALL TONI RONKKO BE LIABLE FOR ANY CLAIM, DAMAGES OR
| OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
| ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
| OTHER DEALINGS IN THE SOFTWARE.

Apache License

Version 2.0, January 2004

<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the

Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside

or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer,

and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<https://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

License for portions of idl.jj in the Java compiler implementation:

Portions of idl.jj were modeled after the example Java 1.5 parser included with JavaCC. For those portions:

Copyright (c) 2006, Sun Microsystems, Inc.
All rights reserved.

| Redistribution and use in source and binary forms, with or without
| modification, are permitted provided that the following conditions are met:

|

| * Redistributions of source code must retain the above copyright notice,
| this list of conditions and the following disclaimer.
| * Redistributions in binary form must reproduce the above copyright
| notice, this list of conditions and the following disclaimer in the
| documentation and/or other materials provided with the distribution.
| * Neither the name of the Sun Microsystems, Inc. nor the names of its
| contributors may be used to endorse or promote products derived from
| this software without specific prior written permission.

| THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
| AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
| IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
| ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
| LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
| CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
| SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
| INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
| CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
| ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
| THE POSSIBILITY OF SUCH DAMAGE.

1.146 asm-commons 7.2

1.146.1 Available under license :

<OWNER> = Regents of the University of California
<ORGANIZATION> = University of California, Berkeley
<YEAR> = 1998

In the original BSD license, both occurrences of the phrase "COPYRIGHT HOLDERS AND CONTRIBUTORS" in the disclaimer read "REGENTS AND CONTRIBUTORS".

Here is the license template:

Copyright (c) <YEAR>, <OWNER>

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the <ORGANIZATION> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS

"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.147 micronaut-gcp 4.6.0

1.147.1 Available under license :

Apache License

Version 2.0, January 2004

<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed

as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this

License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.148 bean-validation-api 2.0.1

1.148.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but

excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its

distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise,

unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 2013 Cognifide Limited

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.149 error_prone_annotations 2.7.1

1.149.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2014 The Error Prone Authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1288519880_1647351834.56/0/error-prone-annotations-2-7-1-sources-
jar/com/google/errorprone/annotations/NoAllocation.java
* /opt/cola/permits/1288519880_1647351834.56/0/error-prone-annotations-2-7-1-sources-
jar/com/google/errorprone/annotations/concurrent/LockMethod.java
* /opt/cola/permits/1288519880_1647351834.56/0/error-prone-annotations-2-7-1-sources-
jar/com/google/errorprone/annotations/concurrent/UnlockMethod.java
No license file was found, but licenses were detected in source scan.
```

```
/*
 * Copyright 2017 The Error Prone Authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

- * /opt/cola/permits/1288519880_1647351834.56/0/error-prone-annotations-2-7-1-sources-jar/com/google/errorprone/annotations/OverridingMethodsMustInvokeSuper.java
- * /opt/cola/permits/1288519880_1647351834.56/0/error-prone-annotations-2-7-1-sources-jar/com/google/errorprone/annotations/DoNotCall.java
- * /opt/cola/permits/1288519880_1647351834.56/0/error-prone-annotations-2-7-1-sources-jar/com/google/errorprone/annotations/concurrent/GuardedBy.java
- * /opt/cola/permits/1288519880_1647351834.56/0/error-prone-annotations-2-7-1-sources-jar/com/google/errorprone/annotations/CheckReturnValue.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2016 The Error Prone Authors.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

- * /opt/cola/permits/1288519880_1647351834.56/0/error-prone-annotations-2-7-1-sources-jar/com/google/errorprone/annotations/FormatString.java
- * /opt/cola/permits/1288519880_1647351834.56/0/error-prone-annotations-2-7-1-sources-jar/com/google/errorprone/annotations/CompatibleWith.java
- * /opt/cola/permits/1288519880_1647351834.56/0/error-prone-annotations-2-7-1-sources-jar/com/google/errorprone/annotations/RestrictedApi.java
- * /opt/cola/permits/1288519880_1647351834.56/0/error-prone-annotations-2-7-1-sources-jar/com/google/errorprone/annotations/MustBeClosed.java
- * /opt/cola/permits/1288519880_1647351834.56/0/error-prone-annotations-2-7-1-sources-jar/com/google/errorprone/annotations/DoNotMock.java
- * /opt/cola/permits/1288519880_1647351834.56/0/error-prone-annotations-2-7-1-sources-jar/com/google/errorprone/annotations/FormatMethod.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2015 The Error Prone Authors.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1288519880_1647351834.56/0/error-prone-annotations-2-7-1-sources-jar/com/google/errorprone/annotations/Var.java
* /opt/cola/permits/1288519880_1647351834.56/0/error-prone-annotations-2-7-1-sources-jar/com/google/errorprone/annotations/concurrent/LazyInit.java
* /opt/cola/permits/1288519880_1647351834.56/0/error-prone-annotations-2-7-1-sources-jar/com/google/errorprone/annotations/IncompatibleModifiers.java
* /opt/cola/permits/1288519880_1647351834.56/0/error-prone-annotations-2-7-1-sources-jar/com/google/errorprone/annotations/RequiredModifiers.java
* /opt/cola/permits/1288519880_1647351834.56/0/error-prone-annotations-2-7-1-sources-jar/com/google/errorprone/annotations/Immutable.java
* /opt/cola/permits/1288519880_1647351834.56/0/error-prone-annotations-2-7-1-sources-jar/com/google/errorprone/annotations/CompileTimeConstant.java
* /opt/cola/permits/1288519880_1647351834.56/0/error-prone-annotations-2-7-1-sources-jar/com/google/errorprone/annotations/SuppressPackageLocation.java
* /opt/cola/permits/1288519880_1647351834.56/0/error-prone-annotations-2-7-1-sources-jar/com/google/errorprone/annotations/ForOverride.java
* /opt/cola/permits/1288519880_1647351834.56/0/error-prone-annotations-2-7-1-sources-jar/com/google/errorprone/annotations/CanIgnoreReturnValue.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2021 The Error Prone Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

```
* /opt/cola/permits/1288519880_1647351834.56/0/error-prone-annotations-2-7-1-sources-
jar/com/google/errorprone/annotations/InlineMeValidationDisabled.java
* /opt/cola/permits/1288519880_1647351834.56/0/error-prone-annotations-2-7-1-sources-
jar/com/google/errorprone/annotations/InlineMe.java
```

1.150 feign-hystrix 10.12

1.150.1 Available under license :

No license file was found, but licenses were detected in source scan.

<!--

Copyright 2012-2020 The Feign Authors

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express

or implied. See the License for the specific language governing permissions and limitations under the License.

-->

Found in path(s):

```
* /opt/cola/permits/1418559522_1663298209.9027398/0/feign-hystrix-10-12-sources-1-jar/META-
INF/maven/io.github.openfeign/feign-hystrix/pom.xml
```

No license file was found, but licenses were detected in source scan.

/**

* Copyright 2012-2020 The Feign Authors

*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License

* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express

* or implied. See the License for the specific language governing permissions and limitations under

* the License.

*/

Found in path(s):

- * /opt/cola/permits/1418559522_1663298209.9027398/0/feign-hystrix-10-12-sources-1-jar/feign/hystrix/HystrixCapability.java
- * /opt/cola/permits/1418559522_1663298209.9027398/0/feign-hystrix-10-12-sources-1-jar/feign/hystrix/HystrixFeign.java
- * /opt/cola/permits/1418559522_1663298209.9027398/0/feign-hystrix-10-12-sources-1-jar/feign/hystrix/FallbackFactory.java
- * /opt/cola/permits/1418559522_1663298209.9027398/0/feign-hystrix-10-12-sources-1-jar/feign/hystrix/ObservableCompletableFuture.java
- * /opt/cola/permits/1418559522_1663298209.9027398/0/feign-hystrix-10-12-sources-1-jar/feign/hystrix/SetterFactory.java
- * /opt/cola/permits/1418559522_1663298209.9027398/0/feign-hystrix-10-12-sources-1-jar/feign/hystrix/HystrixDelegatingContract.java
- * /opt/cola/permits/1418559522_1663298209.9027398/0/feign-hystrix-10-12-sources-1-jar/feign/hystrix/HystrixInvocationHandler.java

1.151 javaserver(tm)-faces-2.3-api 3.1.0.SP01

1.151.1 Available under license :

Eclipse Public License - v 2.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial content Distributed under this Agreement, and

b) in the case of each subsequent Contributor:

- i) changes to the Program, and
- ii) additions to the Program;

where such changes and/or additions to the Program originate from and are Distributed by that particular Contributor. A Contribution "originates" from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include changes or additions to the Program that are not Modified Works.

"Contributor" means any person or entity that Distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions Distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

"Derivative Works" shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

"Modified Works" shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

"Distribute" means the acts of a) distributing or b) making available in any manner that enables the transfer of a copy.

"Source Code" means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Secondary License" means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition

of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

e) Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

3. REQUIREMENTS

3.1 If a Contributor Distributes the Program in any form, then:

a) the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and

b) the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:

i) effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and

iv) requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.

3.2 When the Program is Distributed as Source Code:

a) it must be made available under this Agreement, or if the Program (i) is combined with other material in a separate file or files made available under a Secondary License, and (ii) the initial Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available under the terms of such Secondary Licenses, and

b) a copy of this Agreement must be included with each copy of the Program.

3.3 Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability ("notices") contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in

writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the

minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.

Exhibit A - Form of Secondary Licenses Notice

"This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: {name license(s), version(s), and exceptions or additional permissions here}."

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor
Boston, MA 02110-1335
USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code.

And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it,

under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are

prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those

countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1335 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type
`show w'. This is free software, and you are welcome to redistribute
it under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

CLASSPATH EXCEPTION

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

Notices for Jakarta Server Faces

This content is produced and maintained by the Jakarta Server Faces project.

* Project home: <https://projects.eclipse.org/projects/ee4j.faces>

Trademarks

Jakarta Server Faces is a trademark of the Eclipse Foundation.

Copyright

All content is the property of the respective authors or their employers. For more information regarding authorship of content, please consult the listed source code repository logs.

Declared Project Licenses

This program and the accompanying materials are made available under the terms of the Eclipse Public License v. 2.0 which is available at <http://www.eclipse.org/legal/epl-2.0>. This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License v. 2.0 are satisfied: GNU General Public License, version 2 with the GNU Classpath Exception which is available at <https://www.gnu.org/software/classpath/license.html>.

SPDX-License-Identifier: EPL-2.0 OR GPL-2.0 WITH Classpath-exception-2.0

Source Code

The project maintains the following source code repositories:

* <https://github.com/eclipse-ee4j/faces-api>

Third-party Content

Cryptography

Content may contain encryption software. The country in which you are currently may have restrictions on the import, possession, and use, and/or re-export to another country, of encryption software. BEFORE using any encryption software, please check the country's laws, regulations and policies concerning the import, possession, or use, and re-export of encryption software, to see if this is permitted.

1.152 perfmark-perfmark-api 0.25.0

1.152.1 Available under license :

ASM: a very small and fast Java bytecode manipulation framework
Copyright (c) 2000-2011 INRIA, France Telecom
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from

this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright 2019 Google LLC

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

This product contains a modified portion of 'Catapult', an open source Trace Event viewer for Chrome, Linux, and Android applications, which can be obtained at:

* LICENSE:

* [traceviewer/src/main/resources/io/perfmark/traceviewer/third_party/catapult/LICENSE](#) (New BSD License)

* HOMEPAGE:

* <https://github.com/catapult-project/catapult>

This product contains a modified portion of 'Polymer', a library for Web Components, which can be obtained at:

* LICENSE:

* [traceviewer/src/main/resources/io/perfmark/traceviewer/third_party/polymer/LICENSE](#) (New BSD License)

* HOMEPAGE:

* <https://github.com/Polymer/polymer>

This product contains a modified portion of 'ASM', an open source

Java Bytecode library, which can be obtained at:

* LICENSE:

* agent/src/main/resources/io/perfmark/agent/third_party/asm/LICENSE (BSD style License)

* HOMEPAGE:

* <https://asm.ow2.io/>

Copyright 2015 The Chromium Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of catapult nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common

control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or

documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill,

work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

```
// Copyright (c) 2012 The Polymer Authors. All rights reserved.  
//  
// Redistribution and use in source and binary forms, with or without  
// modification, are permitted provided that the following conditions are  
// met:  
//  
// * Redistributions of source code must retain the above copyright
```

```
// notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
// distribution.
// * Neither the name of Google Inc. nor the names of its
// contributors may be used to endorse or promote products derived from
// this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
// "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
// LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
// A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
// OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
// SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
// LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
// OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
/*
 * Copyright 2019 Google LLC
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

1.153 snake-yaml 1.32

1.153.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/**
 * Copyright (c) 2008, SnakeYAML
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * 
```

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software distributed under the License

* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express

* or implied. See the License for the specific language governing permissions and limitations under

* the License.

*/

Found in path(s):

* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/constructor/ClassLoaderConstructor.java

* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/error/Mark.java

* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/tokens/KeyToken.java

* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/tokens/ValueToken.java

* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/reader/ReaderException.java

* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/tokens/AnchorToken.java

* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/env/EnvScalarConstructor.java

* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/util/EnumUtils.java

* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/constructor/SafeConstructor.java

* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/events/SequenceEndEvent.java

* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/tokens/DocumentStartToken.java

* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/resolver/ResolverTuple.java

* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/comments/CommentType.java

* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/tokens/DirectiveToken.java

* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/tokens/FlowMappingStartToken.java

* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/events/Event.java

* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/representer/Representer.java

* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/TypeDescription.java

* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/events/StreamStartEvent.java

* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/events/SequenceStartEvent.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/scanner/ScannerException.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/LoaderOptions.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/reader/StreamReader.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/util/PlatformFeatureDetector.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/events/CollectionStartEvent.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/tokens/BlockSequenceStartToken.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/scanner/SimpleKey.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/introspector/MissingProperty.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/events/CommentEvent.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/extensions/compactnotation/CompactConstructor.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/constructor/BaseConstructor.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/introspector/PropertyUtils.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/constructor/Construct.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/introspector/BeanAccess.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/reader/UnicodeReader.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/composer/ComposerException.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/emitter/EmitterState.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/introspector/Property.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/tokens/FlowEntryToken.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/scanner/Constant.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/introspector/FieldProperty.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/tokens/BlockMappingStartToken.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/resolver/Resolver.java

* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/tokens/FlowSequenceStartToken.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/nodes/CollectionNode.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/emitter/ScalarAnalysis.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/extensions/compactnotation/CompactData.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/util/ArrayUtils.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/tokens/CommentToken.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/parser/Production.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/representer/BaseRepresenter.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/emitter/EmitterException.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/error/YAMLException.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/events/DocumentEndEvent.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/tokens/TagTuple.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/nodes/AnchorNode.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/tokens/DocumentEndToken.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/events/MappingStartEvent.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/introspector/MethodProperty.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/tokens/StreamEndToken.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/tokens/Token.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/serializer/NumberAnchorGenerator.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/events/DocumentStartEvent.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/nodes/ScalarNode.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/events/AliasEvent.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/composer/Composer.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/tokens/AliasToken.java

* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/introspector/GenericProperty.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/serializer/Serializer.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/representer/Represent.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/nodes/SequenceNode.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/representer/SafeRepresenter.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/nodes/Tag.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/events/ScalarEvent.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/tokens/ScalarToken.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/error/MarkedYAMLErrorException.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/serializer/SerializerException.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/parser/ParserException.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/parser/ParserImpl.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/tokens/TagToken.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/constructor/Constructor.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/events/StreamEndEvent.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/DumperOptions.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/Yaml.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/constructor/AbstractConstruct.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/nodes/NodeTuple.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/comments/CommentLine.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/util/ArrayStack.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/events/MappingEndEvent.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/events/CollectionEndEvent.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/parser/VersionTagsTuple.java

* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/util/UriEncoder.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/emitter/Emitable.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/events/NodeEvent.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/scanner/ScannerImpl.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/tokens/StreamStartToken.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/constructor/ConstructorException.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/nodes/MappingNode.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/scanner/Scanner.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/introspector/PropertySubstitute.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/error/MissingEnvironmentVariableException.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/tokens/FlowSequenceEndToken.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/parser/Parser.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/comments/CommentEventsCollector.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/events/ImplicitTuple.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/tokens/BlockEndToken.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/nodes/NodeId.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/tokens/WhitespaceToken.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/extensions/compactnotation/PackageCompactConstructor.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/emitter/Emitter.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/tokens/FlowMappingEndToken.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/tokens/BlockEntryToken.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/nodes/Node.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/constructor/DuplicateKeyException.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/org/yaml/snakeyaml/serializer/AnchorGenerator.java

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (c) 2008 Google Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-
jar/org/yaml/snakeyaml/external/com/google/gdata/util/common/base/PercentEscaper.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-
jar/org/yaml/snakeyaml/external/com/google/gdata/util/common/base/Escaper.java
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-
jar/org/yaml/snakeyaml/external/com/google/gdata/util/common/base/UnicodeEscaper.java
```

No license file was found, but licenses were detected in source scan.

```
// This module is multi-licensed and may be used under the terms
// EPL, Eclipse Public License, V1.0 or later, http://www.eclipse.org/legal
// LGPL, GNU Lesser General Public License, V2.1 or later, http://www.gnu.org/licenses/lgpl.html
// GPL, GNU General Public License, V2 or later, http://www.gnu.org/licenses/gpl.html
// AL, Apache License, V2.0 or later, http://www.apache.org/licenses
// BSD, BSD License, http://www.opensource.org/licenses/bsd-license.php
/**
 * A Base64 encoder/decoder.
 *
 * <p>
 * This class is used to encode and decode data in Base64 format as described in RFC 1521.
 *
 * <p>
 * Project home page: <a href="http://www.source-code.biz/base64coder/java/">www.
 * source-code.biz/base64coder/java</a><br>
 * Author: Christian d'Heureuse, Inventec Informatik AG, Zurich, Switzerland<br>
 * Multi-licensed: EPL / LGPL / GPL / AL / BSD.
 */
```

Found in path(s):

```
* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-
jar/org/yaml/snakeyaml/external/biz/base64Coder/Base64Coder.java
```

No license file was found, but licenses were detected in source scan.

<name>Apache License, Version 2.0</name>
<url><http://www.apache.org/licenses/LICENSE-2.0.txt></url>

Found in path(s):

* /opt/cola/permits/1473459948_1668478097.0931652/0/snakeyaml-1-32-sources-1-jar/META-INF/maven/org.yaml/snakeyaml/pom.xml

1.154 prometheus-java-simpleclient 0.15.0

1.154.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*

* Written by Doug Lea with assistance from members of JCP JSR-166
* Expert Group and released to the public domain, as explained at
* <http://creativecommons.org/publicdomain/zero/1.0/>
*
* Source: <http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/src/jsr166e/Striped64.java?revision=1.10>
*/

Found in path(s):

* /opt/cola/permits/1341561591_1654802717.0915134/0/simpleclient-0-15-0-sources-jar/io/prometheus/client/Striped64.java

No license file was found, but licenses were detected in source scan.

/*

* Written by Doug Lea with assistance from members of JCP JSR-166
* Expert Group and released to the public domain, as explained at
* <http://creativecommons.org/publicdomain/zero/1.0/>
*
* Source: <http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/src/jsr166e/DoubleAdder.java?revision=1.12>
*/

Found in path(s):

* /opt/cola/permits/1341561591_1654802717.0915134/0/simpleclient-0-15-0-sources-jar/io/prometheus/client/DoubleAdder.java

No license file was found, but licenses were detected in source scan.

/*

Copyright 2012 Andrew Wang (andrew@umbrant.com)

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1341561591_1654802717.0915134/0/simpleclient-0-15-0-sources-jar/io/prometheus/client/CKMSQuantiles.java

No license file was found, but licenses were detected in source scan.

<url><http://www.apache.org/licenses/LICENSE-2.0.txt></url>

Found in path(s):

* /opt/cola/permits/1341561591_1654802717.0915134/0/simpleclient-0-15-0-sources-jar/META-INF/maven/io.prometheus/simpleclient/pom.xml

1.155 byte-buddy-agent 1.9.10

1.155.1 Available under license :

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form,

including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark,

and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.156 icu4j 67.1

1.156.1 Available under license :

COPYRIGHT AND PERMISSION NOTICE (ICU 58 and later)

Copyright 1991-2020 Unicode, Inc. All rights reserved.

Distributed under the Terms of Use in <https://www.unicode.org/copyright.html>.

Permission is hereby granted, free of charge, to any person obtaining a copy of the Unicode data files and any associated documentation (the "Data Files") or Unicode software and any associated documentation (the "Software") to deal in the Data Files or Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Data Files or Software, and to permit persons to whom the Data Files or Software are furnished to do so, provided that either

- (a) this copyright and permission notice appear with all copies of the Data Files or Software, or
- (b) this copyright and permission notice appear in associated Documentation.

THE DATA FILES AND SOFTWARE ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS.

IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THE DATA FILES OR SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in these Data Files or Software without prior written authorization of the copyright holder.

Third-Party Software Licenses

This section contains third-party software notices and/or additional terms for licensed third-party software components included within ICU libraries.

1. ICU License - ICU 1.8.1 to ICU 57.1

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1995-2016 International Business Machines Corporation and others
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the

"Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

All trademarks and registered trademarks mentioned herein are the property of their respective owners.

2. Chinese/Japanese Word Break Dictionary Data (cjdict.txt)

```
# The Google Chrome software developed by Google is licensed under
# the BSD license. Other software included in this distribution is
# provided under other licenses, as set forth below.
#
# The BSD License
# http://opensource.org/licenses/bsd-license.php
# Copyright (C) 2006-2008, Google Inc.
#
# All rights reserved.
#
# Redistribution and use in source and binary forms, with or without
# modification, are permitted provided that the following conditions are met:
#
# Redistributions of source code must retain the above copyright notice,
# this list of conditions and the following disclaimer.
# Redistributions in binary form must reproduce the above
# copyright notice, this list of conditions and the following
# disclaimer in the documentation and/or other materials provided with
# the distribution.
# Neither the name of Google Inc. nor the names of its
```

```
# contributors may be used to endorse or promote products derived from
# this software without specific prior written permission.
#
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
# CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES,
# INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
# MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
# DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
# LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
# CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
# SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
# BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
# LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
# NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
# SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
#
#
# The word list in cjdict.txt are generated by combining three word lists
# listed below with further processing for compound word breaking. The
# frequency is generated with an iterative training against Google web
# corpora.
#
# * Libtabe (Chinese)
# - https://sourceforge.net/project/?group\_id=1519
# - Its license terms and conditions are shown below.
#
# * IPADIC (Japanese)
# - http://chasen.aist-nara.ac.jp/chasen/distribution.html
# - Its license terms and conditions are shown below.
#
# -----COPYING.libtabe ---- BEGIN-----
#
# /*
# * Copyright (c) 1999 TaBE Project.
# * Copyright (c) 1999 Pai-Hsiang Hsiao.
# * All rights reserved.
# *
# * Redistribution and use in source and binary forms, with or without
# * modification, are permitted provided that the following conditions
# * are met:
# *
# * . Redistributions of source code must retain the above copyright
# * notice, this list of conditions and the following disclaimer.
# * . Redistributions in binary form must reproduce the above copyright
# * notice, this list of conditions and the following disclaimer in
# * the documentation and/or other materials provided with the
# * distribution.
```

```
# * . Neither the name of the TaBE Project nor the names of its
# * contributors may be used to endorse or promote products derived
# * from this software without specific prior written permission.
# *
# * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
# * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
# * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
# * FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
# * REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
# * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
# * (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
# * SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
# * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
# * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
# * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
# * OF THE POSSIBILITY OF SUCH DAMAGE.
# */
#
# /*
# * Copyright (c) 1999 Computer Systems and Communication Lab,
# *      Institute of Information Science, Academia
# *      Sinica. All rights reserved.
# *
# * Redistribution and use in source and binary forms, with or without
# * modification, are permitted provided that the following conditions
# * are met:
# *
# * . Redistributions of source code must retain the above copyright
# * notice, this list of conditions and the following disclaimer.
# * . Redistributions in binary form must reproduce the above copyright
# * notice, this list of conditions and the following disclaimer in
# * the documentation and/or other materials provided with the
# * distribution.
# * . Neither the name of the Computer Systems and Communication Lab
# * nor the names of its contributors may be used to endorse or
# * promote products derived from this software without specific
# * prior written permission.
# *
# * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
# * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
# * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
# * FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
# * REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
# * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
# * (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
# * SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
# * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
# * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
```

```
# * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
# * OF THE POSSIBILITY OF SUCH DAMAGE.
# */
#
# Copyright 1996 Chih-Hao Tsai @ Beckman Institute,
# University of Illinois
# c-tsai4@uiuc.edu http://casper.beckman.uiuc.edu/~c-tsai4
#
# -----COPYING.libtabe-----END-----
#
#
# -----COPYING.ipadic-----BEGIN-----
#
# Copyright 2000, 2001, 2002, 2003 Nara Institute of Science
# and Technology. All Rights Reserved.
#
# Use, reproduction, and distribution of this software is permitted.
# Any copy of this software, whether in its original form or modified,
# must include both the above copyright notice and the following
# paragraphs.
#
# Nara Institute of Science and Technology (NAIST),
# the copyright holders, disclaims all warranties with regard to this
# software, including all implied warranties of merchantability and
# fitness, in no event shall NAIST be liable for
# any special, indirect or consequential damages or any damages
# whatsoever resulting from loss of use, data or profits, whether in an
# action of contract, negligence or other tortuous action, arising out
# of or in connection with the use or performance of this software.
#
# A large portion of the dictionary entries
# originate from ICOT Free Software. The following conditions for ICOT
# Free Software applies to the current dictionary as well.
#
# Each User may also freely distribute the Program, whether in its
# original form or modified, to any third party or parties, PROVIDED
# that the provisions of Section 3 ("NO WARRANTY") will ALWAYS appear
# on, or be attached to, the Program, which is distributed substantially
# in the same form as set out herein and that such intended
# distribution, if actually made, will neither violate or otherwise
# contravene any of the laws and regulations of the countries having
# jurisdiction over the User or the intended distribution itself.
#
# NO WARRANTY
#
# The program was produced on an experimental basis in the course of the
# research and development conducted during the project and is provided
# to users as so produced on an experimental basis. Accordingly, the
```

program is provided without any warranty whatsoever, whether express,
implied, statutory or otherwise. The term "warranty" used herein
includes, but is not limited to, any warranty of the quality,
performance, merchantability and fitness for a particular purpose of
the program and the nonexistence of any infringement or violation of
any right of any third party.

#

Each user of the program will agree and understand, and be deemed to
have agreed and understood, that there is no warranty whatsoever for
the program and, accordingly, the entire risk arising from or
otherwise connected with the program is assumed by the user.

#

Therefore, neither ICOT, the copyright holder, or any other
organization that participated in or was otherwise related to the
development of the program and their respective officials, directors,
officers and other employees shall be held liable for any and all
damages, including, without limitation, general, special, incidental
and consequential damages, arising out of or otherwise in connection
with the use or inability to use the program or any product, material
or result produced or otherwise obtained by using the program,
regardless of whether they have been advised of, or otherwise had
knowledge of, the possibility of such damages at any time during the
project or thereafter. Each user will be deemed to have agreed to the
foregoing by his or her commencement of use of the program. The term
"use" as used herein includes, but is not limited to, the use,
modification, copying and distribution of the program and the
production of secondary products from the program.

#

In the case where the program, whether in its original form or
modified, was distributed or delivered to or received by a user from
any person, organization or entity other than ICOT, unless it makes or
grants independently of ICOT any specific warranty to the user in
writing, such person, organization or entity, will also be exempted
from and not be held liable to the user for any such damages as noted
above as far as the program is concerned.

#

-----COPYING.ipadic-----END-----

3. Lao Word Break Dictionary Data (laodict.txt)

Copyright (c) 2013 International Business Machines Corporation
and others. All Rights Reserved.

#

Project: <http://code.google.com/p/lao-dictionary/>

Dictionary: <http://lao-dictionary.googlecode.com/git/Lao-Dictionary.txt>

License: <http://lao-dictionary.googlecode.com/git/Lao-Dictionary-LICENSE.txt>

(copied below)

#

```
# This file is derived from the above dictionary, with slight
# modifications.
# -----
# Copyright (C) 2013 Brian Eugene Wilson, Robert Martin Campbell.
# All rights reserved.
#
# Redistribution and use in source and binary forms, with or without
# modification,
# are permitted provided that the following conditions are met:
#
#
# Redistributions of source code must retain the above copyright notice, this
# list of conditions and the following disclaimer. Redistributions in
# binary form must reproduce the above copyright notice, this list of
# conditions and the following disclaimer in the documentation and/or
# other materials provided with the distribution.
#
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
# "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
# LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
# FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
# COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT,
# INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
# (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
# SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
# HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
# STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
# ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
# OF THE POSSIBILITY OF SUCH DAMAGE.
# -----
```

4. Burmese Word Break Dictionary Data (burmesedict.txt)

```
# Copyright (c) 2014 International Business Machines Corporation
# and others. All Rights Reserved.
#
# This list is part of a project hosted at:
# github.com/kanyawtech/myanmar-karen-word-lists
#
# -----
# Copyright (c) 2013, LeRoy Benjamin Sharon
# All rights reserved.
#
# Redistribution and use in source and binary forms, with or without
# modification, are permitted provided that the following conditions
# are met: Redistributions of source code must retain the above
# copyright notice, this list of conditions and the following
```

```
# disclaimer. Redistributions in binary form must reproduce the
# above copyright notice, this list of conditions and the following
# disclaimer in the documentation and/or other materials provided
# with the distribution.
#
# Neither the name Myanmar Karen Word Lists, nor the names of its
# contributors may be used to endorse or promote products derived
# from this software without specific prior written permission.
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
# CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES,
# INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
# MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
# DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS
# BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
# EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED
# TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
# DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON
# ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR
# TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF
# THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
# SUCH DAMAGE.
# -----
```

5. Time Zone Database

ICU uses the public domain data and code derived from Time Zone Database for its time zone support. The ownership of the TZ database is explained in BCP 175: Procedure for Maintaining the Time Zone Database section 7.

7. Database Ownership

```
#
# The TZ database itself is not an IETF Contribution or an IETF
# document. Rather it is a pre-existing and regularly updated work
# that is in the public domain, and is intended to remain in the
# public domain. Therefore, BCPS 78 [RFC5378] and 79 [RFC3979] do
# not apply to the TZ Database or contributions that individuals make
# to it. Should any claims be made and substantiated against the TZ
# Database, the organization that is providing the IANA
# Considerations defined in this RFC, under the memorandum of
# understanding with the IETF, currently ICANN, may act in accordance
# with all competent court orders. No ownership claims will be made
# by ICANN or the IETF Trust on the database or the code. Any person
# making a contribution to the database or code waives all rights to
# future claims in that contribution or in the TZ Database.
```

6. Google double-conversion

Copyright 2006-2011, the V8 project authors. All rights reserved.
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

- * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following
disclaimer in the documentation and/or other materials provided
with the distribution.
- * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived
from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.157 generex 1.0.2

1.157.1 Available under license :

No license file was found, but licenses were detected in source scan.

Generex is licensed under the Apache License, Version 2.0.

<http://www.apache.org/licenses/LICENSE-2.0>

Found in path(s):

* /opt/cola/permits/1162863542_1620923914.63/0/mifmif-generex-1-0-2-0-ge910243-tar-gz/mifmif-Generex-e910243/README.md

No license file was found, but licenses were detected in source scan.

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.github.mifmif</groupId>
  <artifactId>generex</artifactId>
  <version>1.0.2</version>
```

```

<name>Generex</name>
<url>https://github.com/mifmif/Generex/tree/master</url>
<description>Generex A Java Library for regex to Strings generation</description>
<packaging>jar</packaging>
<licenses>
  <license>
    <name>The Apache Software License, Version 2.0</name>
    <url>http://www.apache.org/licenses/LICENSE-2.0.txt</url>
    <distribution>repo</distribution>
  </license>
</licenses>
<developers>
  <developer>
    <id>mifmif</id>
    <name>MIFRAH Youssef</name>
    <email>mifmif.com@gmail.com</email>
  </developer>
  <developer>
    <id>mkolisnyk</id>
    <name>mkolisnyk</name>
    <email>kolesnik.nickolay@gmail.com</email>
  </developer>
</developers>

<parent>
  <groupId>org.sonatype.oss</groupId>
  <artifactId>oss-parent</artifactId>
  <version>7</version>
</parent>

<properties>
  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  <maven.pmd.excludes>*/generated-sources/*</maven.pmd.excludes>
</properties>
<scm>
  <connection>scm:git:git@github.com:mifmif/Generex.git</connection>
  <developerConnection>scm:git:git@github.com:mifmif/Generex.git</developerConnection>
  <url>https://github.com/mifmif/Generex.git</url>
</scm>
<ciManagement>
  <url>https://travis-ci.org/mifmif/Generex/builds</url>
  <system>Travis</system>
<notifiers>
  <notifier>
    <address>mifmif.com@gmail.com</address>
    <type>mail</type>
  </notifier>
  <notifier>

```

```

    <address>kolesnik.nickolay@gmail.com</address>
    <type>mail</type>
  </notifier>
</notifiers>
</ciManagement>

<distributionManagement>
  <snapshotRepository>
    <id>ossrh</id>
    <url>https://oss.sonatype.org/content/repositories/snapshots</url>
  </snapshotRepository>
  <repository>
    <id>ossrh</id>
    <url>https://oss.sonatype.org/service/local/staging/deploy/maven2/</url>
  </repository>
</distributionManagement>

<build>
  <sourceDirectory>src/main/java</sourceDirectory>
  <testSourceDirectory>src/test/java</testSourceDirectory>
  <resources>
    <resource>
      <directory>src</directory>
      <excludes>
        <exclude>**/*.java</exclude>
      </excludes>
    </resource>
    <resource>
      <directory>target/dependency</directory>
      <excludes>
        <exclude>**/*.java</exclude>
      </excludes>
    </resource>
    <resource>
      <directory>src/main/resources</directory>
      <includes>
        <include>**/*.properties</include>
      </includes>
    </resource>
  </resources>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-install-plugin</artifactId>
      <version>2.5.2</version>
      <configuration>
        <file>target/${project.artifactId}-${project.version}.jar</file>
      </configuration>
    </plugin>
  </plugins>

```

```

<groupId>${project.groupId}</groupId>
<artifactId>${project.artifactId}</artifactId>
<version>${project.version}</version>
<packaging>maven-plugin</packaging>
</configuration>
</plugin>

<plugin>
<artifactId>maven-compiler-plugin</artifactId>
<version>3.5.1</version>
<configuration>
<source>1.5</source>
<target>1.5</target>
</configuration>
</plugin>
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-jar-plugin</artifactId>
<version>3.0.2</version>
<configuration>
<excludes>
<exclude>*</exclude>
<exclude>com/thoughtworks/**/*</exclude>
<exclude>freemarker/**/*</exclude>
<exclude>ftl/**/*</exclude>
<exclude>i18n/**/*</exclude>
<exclude>style/**/*</exclude>
<exclude>junit/**/*</exclude>
<exclude>licenses/**/*</exclude>
<exclude>META-INF/maven/**/*</exclude>
<exclude>org/codehaus/**/*</exclude>
<exclude>org/hamcrest/**/*</exclude>
<exclude>org/jbehave/**/*</exclude>
<exclude>org/junit/**/*</exclude>
<exclude>org/testng/**/*</exclude>
<exclude>org/xmlpull/**/*</exclude>
<exclude>stories/**/*</exclude>
<exclude>style/**/*</exclude>
<exclude>tests/**/*</exclude>
</excludes>
<archive>
<manifest>
<addClasspath>>false</addClasspath>
<addDefaultImplementationEntries>>true</addDefaultImplementationEntries>
<addDefaultSpecificationEntries>>true</addDefaultSpecificationEntries>
<addExtensions>>false</addExtensions>
<classpathLayoutType>simple</classpathLayoutType>
</manifest>

```

```

    </archive>
  </configuration>
</plugin>
<plugin>
  <artifactId>maven-assembly-plugin</artifactId>
  <version>2.6</version>
  <configuration>
    <descriptors>
      <descriptor>./sources.xml</descriptor>
    </descriptors>
  </configuration>
</plugin>
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-dependency-plugin</artifactId>
  <version>2.10</version>
</plugin>
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-javadoc-plugin</artifactId>
  <version>2.10.3</version>
  <configuration>
    <show>private</show>
    <nohelp>>false</nohelp>
    <failOnError>>false</failOnError>
    <sourcepath>src/main/java</sourcepath>
  </configuration>
</plugin>
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-release-plugin</artifactId>
  <configuration>
    <tagNameFormat>v@{project.version}</tagNameFormat>
    <preparationGoals>package install:install-file</preparationGoals>
    <scmCommentPrefix>#3</scmCommentPrefix>
  </configuration>
</plugin>
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-deploy-plugin</artifactId>
  <version>2.8.2</version>
  <configuration>
    <skip>>true</skip>
  </configuration>
</plugin>

<plugin>

```

```
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-source-plugin</artifactId>
<version>3.0.1</version>
<executions>
<execution>
<goals>
<goal>jar-no-fork</goal>
</goals>
</execution>
</executions>
</plugin>
```

```
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-javadoc-plugin</artifactId>
<executions>
<execution>
<id>attach-javadocs</id>
<goals>
<goal>jar</goal>
</goals>
</execution>
</executions>
</plugin>
```

```
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-gpg-plugin</artifactId>
<executions>
<execution>
<id>sign-artifacts</id>
<phase>deploy</phase>
<goals>
<goal>sign</goal>
</goals>
</execution>
</executions>

</plugin>
```

```
<plugin>
<groupId>org.sonatype.plugins</groupId>
<artifactId>nexus-staging-maven-plugin</artifactId>
<version>1.6.3</version>
<executions>
```



```

<configLocation>./sun_checks.xml</configLocation>
<suppressionsLocation>./checkstyle-suppressions.xml</suppressionsLocation>
<failOnViolation>>true</failOnViolation>
</configuration>
<executions>
  <execution>
    <goals>
      <goal>check</goal>
    </goals>
  </execution>
</executions>
</plugin>
<plugin>
  <groupId>org.codehaus.mojo</groupId>
  <artifactId>cobertura-maven-plugin</artifactId>
  <version>2.6</version>
  <configuration>
    <check>
      <haltOnFailure>>false</haltOnFailure>
      <totalBranchRate>80</totalBranchRate>
      <totalLineRate>80</totalLineRate>
      <packageLineRate>80</packageLineRate>
      <packageBranchRate>80</packageBranchRate>
    </check>
    <instrumentation>
      <excludes>
        <exclude>*/HelpMojo.class</exclude>
      </excludes>
    </instrumentation>
  </configuration>
</plugin>
<executions>
  <execution>
    <id>verification</id>
    <phase>verify</phase>
    <goals>
      <goal>check</goal>
      <goal>cobertura</goal>
    </goals>
  </execution>
</executions>
</plugin>
<plugin>
  <groupId>org.codehaus.mojo</groupId>
  <artifactId>javancss-maven-plugin</artifactId>
  <version>2.0</version>
  <configuration>
    <encoding>${project.build.sourceEncoding}</encoding>
    <ccnLimit>10</ccnLimit>
  </configuration>
</plugin>

```

```

<ncssLimit>100</ncssLimit>
<failOnViolation>true</failOnViolation>
<excludes>
  <exclude>**/tests/**/*.*</exclude>
</excludes>
</configuration>
<executions>
  <execution>
    <goals>
      <goal>check</goal>
    </goals>
  </execution>
</executions>
</plugin>
<plugin>
  <groupId>org.codehaus.mojo</groupId>
  <artifactId>findbugs-maven-plugin</artifactId>
  <version>3.0.3</version>
  <configuration>
    <encoding>${project.build.sourceEncoding}</encoding>
    <failOnError>true</failOnError>
    <excludeFilterFile>findBugsExclude.xml</excludeFilterFile>
  </configuration>
  <executions>
    <execution>
      <goals>
        <goal>check</goal>
      </goals>
    </execution>
  </executions>
</plugin>
<plugin>
  <groupId>org.codehaus.mojo</groupId>
  <artifactId>jdepend-maven-plugin</artifactId>
  <version>2.0</version>
</plugin>
</plugins>
</build>
<dependencies>
  <dependency>
    <groupId>dk.brics.automaton</groupId>
    <artifactId>automaton</artifactId>
    <version>1.11-8</version>
  </dependency>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.12</version>

```

```
<scope>test</scope>
</dependency>
<dependency>
  <groupId>org.hamcrest</groupId>
  <artifactId>hamcrest-junit</artifactId>
  <version>2.0.0.0</version>
  <scope>test</scope>
</dependency>
</dependencies>
</project>
```

Found in path(s):

* /opt/cola/permits/1162863542_1620923914.63/0/mifmif-generex-1-0-2-0-ge910243-tar-gz/mifmif-Generex-e910243/pom.xml

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2014 y.mifrah
 *
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *   http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

* /opt/cola/permits/1162863542_1620923914.63/0/mifmif-generex-1-0-2-0-ge910243-tar-gz/mifmif-Generex-e910243/src/main/java/com/mifmif/common/regex/util/Iterable.java
* /opt/cola/permits/1162863542_1620923914.63/0/mifmif-generex-1-0-2-0-ge910243-tar-gz/mifmif-Generex-e910243/src/main/java/com/mifmif/common/regex/GenerexIterator.java
* /opt/cola/permits/1162863542_1620923914.63/0/mifmif-generex-1-0-2-0-ge910243-tar-gz/mifmif-Generex-e910243/src/main/java/com/mifmif/common/regex/util/Iterator.java
* /opt/cola/permits/1162863542_1620923914.63/0/mifmif-generex-1-0-2-0-ge910243-tar-gz/mifmif-Generex-e910243/src/main/java/com/mifmif/common/regex/Node.java
* /opt/cola/permits/1162863542_1620923914.63/0/mifmif-generex-1-0-2-0-ge910243-tar-gz/mifmif-Generex-e910243/src/main/java/com/mifmif/common/regex/Main.java
* /opt/cola/permits/1162863542_1620923914.63/0/mifmif-generex-1-0-2-0-ge910243-tar-gz/mifmif-Generex-e910243/src/main/java/com/mifmif/common/regex/Generex.java

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2015 y.mifrah
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1162863542_1620923914.63/0/mifmif-generex-1-0-2-0-ge910243-tar-gz/mifmif-Generex-
e910243/src/test/java/com/mifmif/common/regex/GenerexUnitTest.java
No license file was found, but licenses were detected in source scan.
```

```
/*
 * Copyright 2015 y.mifrah
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1162863542_1620923914.63/0/mifmif-generex-1-0-2-0-ge910243-tar-gz/mifmif-Generex-
e910243/src/test/java/com/mifmif/common/regex/GenerexIteratorUnitTest.java
```

1.158 apache-http-client 4.5.13

1.158.1 Available under license :

Apache HttpComponents Client

Copyright 1999-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications

represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without

modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. **Trademarks.** This License does not grant permission to use the trade

names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

=====

This project includes Public Suffix List copied from
<https://publicsuffix.org/list/effective_tld_names.dat>
licensed under the terms of the Mozilla Public License, v. 2.0

Full license text: <<http://mozilla.org/MPL/2.0/>>

Mozilla Public License Version 2.0

=====

1. Definitions

1.1. "Contributor"

means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

1.2. "Contributor Version"

means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution.

1.3. "Contribution"

means Covered Software of a particular Contributor.

1.4. "Covered Software"

means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

1.5. "Incompatible With Secondary Licenses"

means

(a) that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or

(b) that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

1.6. "Executable Form"

means any form of the work other than Source Code Form.

1.7. "Larger Work"

means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

1.8. "License"

means this document.

1.9. "Licensable"

means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

1.10. "Modifications"

means any of the following:

- (a) any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or
- (b) any new file in Source Code Form that contains any Covered Software.

1.11. "Patent Claims" of a Contributor

means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

1.12. "Secondary License"

means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

1.13. "Source Code Form"

means the form of the work preferred for making modifications.

1.14. "You" (or "Your")

means an individual or a legal entity exercising rights under this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

- (a) under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or

as part of a Larger Work; and

- (b) under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

- (a) for any code that a Contributor has removed from Covered Software; or
- (b) for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or
- (c) under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

- (a) such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and
- (b) You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of

the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have

come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

```
*****
*
*
* 6. Disclaimer of Warranty
* -----
*
* Covered Software is provided under this License on an "as is"
* basis, without warranty of any kind, either expressed, implied, or
* statutory, including, without limitation, warranties that the
* Covered Software is free of defects, merchantable, fit for a
* particular purpose or non-infringing. The entire risk as to the
* quality and performance of the Covered Software is with You.
* Should any Covered Software prove defective in any respect, You
* (not any Contributor) assume the cost of any necessary servicing,
* repair, or correction. This disclaimer of warranty constitutes an
* essential part of this License. No use of any Covered Software is
* authorized under this License except under this disclaimer.
*
*
*****
```

```
*****
*
*
* 7. Limitation of Liability
* -----
*
* Under no circumstances and under no legal theory, whether tort
* (including negligence), contract, or otherwise, shall any
* Contributor, or anyone who distributes Covered Software as
* permitted above, be liable to You for any direct, indirect,
* special, incidental, or consequential damages of any character
*
*****
```

* including, without limitation, damages for lost profits, loss of *
 * goodwill, work stoppage, computer failure or malfunction, or any *
 * and all other commercial damages or losses, even if such party *
 * shall have been informed of the possibility of such damages. This *
 * limitation of liability shall not apply to liability for death or *
 * personal injury resulting from such party's negligence to the *
 * extent applicable law prohibits such limitation. Some *
 * jurisdictions do not allow the exclusion or limitation of *
 * incidental or consequential damages, so this exclusion and *
 * limitation may not apply to You. *
 * *

8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license

steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - "Incompatible With Secondary Licenses" Notice

This Source Code Form is "Incompatible With Secondary Licenses", as defined by the Mozilla Public License, v. 2.0.

1.159 jetbrains-kotlin-kotlin-stdlib-jdk7 1.6.21

1.159.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2010-2017 JetBrains s.r.o.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

- * You may obtain a copy of the License at
- *
- * <http://www.apache.org/licenses/LICENSE-2.0>
- *
- * Unless required by applicable law or agreed to in writing, software
- * distributed under the License is distributed on an "AS IS" BASIS,
- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.
- */

Found in path(s):

* /opt/cola/permits/1343419565_1655661636.7344434/0/kotlin-stdlib-jdk7-1-6-21-sources-jar/kotlin/AutoCloseable.kt

1.160 commons-codec 1.14

1.160.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made,

use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability

incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Apache Commons Codec

Copyright 2002-2019 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<https://www.apache.org/>).

`src/test/org/apache/commons/codec/language/DoubleMetaphoneTest.java`

contains test data from <http://aspell.net/test/orig/batch0.tab>.

Copyright (C) 2002 Kevin Atkinson (kevina@gnu.org)

=====
The content of package `org.apache.commons.codec.language.bm` has been translated from the original php source code available at <http://stevemorse.org/phoneticinfo.htm> with permission from the original authors.

Original source copyright:

Copyright (c) 2008 Alexander Beider & Stephen P. Morse.

1.161 fabric8-:::-kubernetes-model-::- admission-registration,-authentication-and- authorization 4.13.3

1.161.1 Available under license :

No license file was found, but licenses were detected in source scan.

*

* Copyright (C) 2015 Red Hat, Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*

Found in path(s):

* /opt/cola/permits/1288519860_1647861596.41/0/kubernetes-model-admissionregistration-4-13-3-jar/manifest.vm

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0

Bnd-LastModified: 1619068462582

Build-Jdk-Spec: 1.8

Bundle-Description: Java client for Kubernetes and OpenShift

Bundle-DocURL: <http://redhat.com>

Bundle-License: <http://www.apache.org/licenses/LICENSE-2.0.txt>

Bundle-ManifestVersion: 2

Bundle-Name: Fabric8 :: Kubernetes Model :: Admission Registration, Au
thentication and Authorization

Bundle-SymbolicName: io.fabric8.kubernetes-model-admissionregistration

Bundle-Vendor: Red Hat

Bundle-Version: 4.13.3

Created-By: Apache Maven Bundle Plugin

Export-Package: io.fabric8.kubernetes.api.model.admission;uses:="com.f
asterxml.jackson.annotation,com.fasterxml.jackson.databind,com.faster
xml.jackson.databind.annotation,io.fabric8.kubernetes.api.builder,io.
fabric8.kubernetes.api.model,io.fabric8.kubernetes.api.model.authenti
cation";version="4.13.3",io.fabric8.kubernetes.api.model.admissionreg

istration.v1;uses:="com.fasterxml.jackson.annotation,com.fasterxml.ja
ckson.databind,com.fasterxml.jackson.databind.annotation,io.fabric8.k
ubernetes.api.builder,io.fabric8.kubernetes.api.model,io.fabric8.kube
rnetes.model.annotation";version="4.13.3",io.fabric8.kubernetes.api.m
odel.admissionregistration.v1beta1;uses:="com.fasterxml.jackson.annot
ation,com.fasterxml.jackson.databind,com.fasterxml.jackson.databind.a
nnotation,io.fabric8.kubernetes.api.builder,io.fabric8.kubernetes.api
.model,io.fabric8.kubernetes.model.annotation";version="4.13.3",io.f
abric8.kubernetes.api.model.authentication;uses:="com.fasterxml.jackso
n.annotation,com.fasterxml.jackson.databind,com.fasterxml.jackson.dat
abind.annotation,io.fabric8.kubernetes.api.builder,io.fabric8.kuberne
tes.api.model";version="4.13.3",io.fabric8.kubernetes.api.model.autho
rization.v1;uses:="com.fasterxml.jackson.annotation,com.fasterxml.jac
kson.databind,com.fasterxml.jackson.databind.annotation,io.fabric8.ku
bernetes.api.builder,io.fabric8.kubernetes.api.model";version="4.13.3
",io.fabric8.kubernetes.api.model.authorization.v1beta1;uses:="com.f
asterxml.jackson.annotation,com.fasterxml.jackson.databind,com.fasterx
ml.jackson.databind.annotation,io.fabric8.kubernetes.api.builder,io.f
abric8.kubernetes.api.model";version="4.13.3"

Implementation-Title: Fabric8 :: Kubernetes Model :: Admission Registr
ation, Authentication and Authorization

Implementation-Vendor: Red Hat

Implementation-Version: 4.13.3

Import-Package: com.fasterxml.jackson.annotation;version="[2.11,3)",co
m.fasterxml.jackson.databind;version="[2.11,3)",com.fasterxml.jackson
.databind.annotation;version="[2.11,3)",io.fabric8.kubernetes.api.bui
lder;version="[4.13,5)",io.fabric8.kubernetes.api.model,io.fabric8.ku
bernetes.api.model.authentication,io.fabric8.kubernetes.model.annotat
ion;version="[4.13,5)"

Require-Capability: osgi.ee;filter:="(&(osgi.ee=JavaSE)(version=1.8))"

Specification-Title: Fabric8 :: Kubernetes Model :: Admission Registra
tion, Authentication and Authorization

Specification-Vendor: Red Hat

Specification-Version: 4.13

Tool: Bnd-5.1.1.202006162103

Found in path(s):

* /opt/cola/permits/1288519860_1647861596.41/0/kubernetes-model-admissionregistration-4-13-3-jar/META-
INF/MANIFEST.MF

No license file was found, but licenses were detected in source scan.

Copyright (C) 2015 Red Hat, Inc.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE>

2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Found in path(s):

* /opt/cola/permits/1288519860_1647861596.41/0/kubernetes-model-admissionregistration-4-13-3-jar/META-INF/maven/io.fabric8/kubernetes-model-admissionregistration/pom.xml

1.162 wire-protocol-buffer-runtime 3.6.0

1.162.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2016 Square Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
*/
```

Found in path(s):

* /opt/cola/permits/1340031473_1654666144.907512/0/wire-runtime-3-6-0-sources-jar/jvmMain/com/squareup/wire/EnumAdapter.kt
* /opt/cola/permits/1340031473_1654666144.907512/0/wire-runtime-3-6-0-sources-jar/jvmMain/com/squareup/wire/AndroidMessage.kt
* /opt/cola/permits/1340031473_1654666144.907512/0/wire-runtime-3-6-0-sources-jar/commonMain/com/squareup/wire/EnumAdapter.kt

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2019 Square, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
*/
```

* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1340031473_1654666144.907512/0/wire-runtime-3-6-0-sources-
jar/commonMain/com/squareup/wire/internal/Util.kt

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2020 Square Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1340031473_1654666144.907512/0/wire-runtime-3-6-0-sources-
jar/commonMain/com/squareup/wire/OneOf.kt

* /opt/cola/permits/1340031473_1654666144.907512/0/wire-runtime-3-6-0-sources-
jar/commonMain/com/squareup/wire/Instant.kt

* /opt/cola/permits/1340031473_1654666144.907512/0/wire-runtime-3-6-0-sources-
jar/jvmMain/com/squareup/wire/Duration.kt

* /opt/cola/permits/1340031473_1654666144.907512/0/wire-runtime-3-6-0-sources-
jar/jvmMain/com/squareup/wire/internal/EnumJsonFormatter.kt

* /opt/cola/permits/1340031473_1654666144.907512/0/wire-runtime-3-6-0-sources-
jar/commonMain/com/squareup/wire/WireEnumConstant.kt

* /opt/cola/permits/1340031473_1654666144.907512/0/wire-runtime-3-6-0-sources-
jar/commonMain/com/squareup/wire/Duration.kt

* /opt/cola/permits/1340031473_1654666144.907512/0/wire-runtime-3-6-0-sources-
jar/jvmMain/com/squareup/wire/Instant.kt

* /opt/cola/permits/1340031473_1654666144.907512/0/wire-runtime-3-6-0-sources-

```
jar/jvmMain/com/squareup/wire/internal/DurationJsonFormatter.kt
* /opt/cola/permits/1340031473_1654666144.907512/0/wire-runtime-3-6-0-sources-
jar/jvmMain/com/squareup/wire/internal/JsonFormatter.kt
* /opt/cola/permits/1340031473_1654666144.907512/0/wire-runtime-3-6-0-sources-
jar/commonMain/com/squareup/wire/Syntax.kt
* /opt/cola/permits/1340031473_1654666144.907512/0/wire-runtime-3-6-0-sources-
jar/jvmMain/com/squareup/wire/internal/InstantJsonFormatter.kt
* /opt/cola/permits/1340031473_1654666144.907512/0/wire-runtime-3-6-0-sources-
jar/commonMain/com/squareup/wire/AnyMessage.kt
* /opt/cola/permits/1340031473_1654666144.907512/0/wire-runtime-3-6-0-sources-
jar/jvmMain/com/squareup/wire/internal/JsonIntegration.kt
No license file was found, but licenses were detected in source scan.
```

```
/*
* Copyright (c) 2016, the R8 project authors.
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions are met:
*
* * Redistributions of source code must retain the above copyright notice, this
* list of conditions and the following disclaimer.
*
* * Redistributions in binary form must reproduce the above copyright notice,
* this list of conditions and the following disclaimer in the documentation
* and/or other materials provided with the distribution.
*
* * Neither the name of Google Inc. nor the names of its
* contributors may be used to endorse or promote products derived from
* this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
* DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
* SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
* CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
* OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
* OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/
```

Found in path(s):

```
* /opt/cola/permits/1340031473_1654666144.907512/0/wire-runtime-3-6-0-sources-
jar/commonMain/com/squareup/wire/internal/MathMethods.kt
No license file was found, but licenses were detected in source scan.
```

```
/*
 * Copyright 2013 Square Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

```
// Copyright 2013 Google Inc. All rights reserved.
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
// * Neither the name of Google Inc. nor the names of its
// this software without specific prior written permission.
```

Found in path(s):

```
* /opt/cola/permits/1340031473_1654666144.907512/0/wire-runtime-3-6-0-sources-
jar/commonMain/com/squareup/wire/ProtoWriter.kt
```

No license file was found, but licenses were detected in source scan.

```
// Copyright 2013 Google Inc. All rights reserved.
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
// * Neither the name of Google Inc. nor the names of its
// this software without specific prior written permission.
```

Found in path(s):

```
* /opt/cola/permits/1340031473_1654666144.907512/0/wire-runtime-3-6-0-sources-
jar/commonMain/com/squareup/wire/ProtoReader.kt
```

No license file was found, but licenses were detected in source scan.

```
/*
```

* Copyright 2015 Square Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1340031473_1654666144.907512/0/wire-runtime-3-6-0-sources-jar/commonMain/com/squareup/wire/internal/MutableOnWriteList.kt
* /opt/cola/permits/1340031473_1654666144.907512/0/wire-runtime-3-6-0-sources-jar/jvmMain/com/squareup/wire/internal/FieldBinding.kt
* /opt/cola/permits/1340031473_1654666144.907512/0/wire-runtime-3-6-0-sources-jar/commonMain/com/squareup/wire/ProtoAdapter.kt
* /opt/cola/permits/1340031473_1654666144.907512/0/wire-runtime-3-6-0-sources-jar/jvmMain/com/squareup/wire/ProtoAdapter.kt
* /opt/cola/permits/1340031473_1654666144.907512/0/wire-runtime-3-6-0-sources-jar/jvmMain/com/squareup/wire/MessageSerializedForm.kt

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2019 Square Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1340031473_1654666144.907512/0/wire-runtime-3-6-0-sources-jar/commonMain/com/squareup/wire/internal/Internal.kt
* /opt/cola/permits/1340031473_1654666144.907512/0/wire-runtime-3-6-0-sources-

jar/commonMain/com/squareup/wire/MessageSink.kt
* /opt/cola/permits/1340031473_1654666144.907512/0/wire-runtime-3-6-0-sources-jar/jvmMain/com/squareup/wire/internal/-Platform.kt
* /opt/cola/permits/1340031473_1654666144.907512/0/wire-runtime-3-6-0-sources-jar/commonMain/com/squareup/wire/WireRpc.kt
* /opt/cola/permits/1340031473_1654666144.907512/0/wire-runtime-3-6-0-sources-jar/commonMain/com/squareup/wire/Service.kt
* /opt/cola/permits/1340031473_1654666144.907512/0/wire-runtime-3-6-0-sources-jar/jvmMain/com/squareup/wire/MessageSink.kt
* /opt/cola/permits/1340031473_1654666144.907512/0/wire-runtime-3-6-0-sources-jar/commonMain/com/squareup/wire/internal/-Platform.kt
* /opt/cola/permits/1340031473_1654666144.907512/0/wire-runtime-3-6-0-sources-jar/jvmMain/com/squareup/wire/MessageSource.kt
* /opt/cola/permits/1340031473_1654666144.907512/0/wire-runtime-3-6-0-sources-jar/commonMain/com/squareup/wire/MessageSource.kt

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2013 Square Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1340031473_1654666144.907512/0/wire-runtime-3-6-0-sources-jar/jvmMain/com/squareup/wire/RuntimeEnumAdapter.kt

* /opt/cola/permits/1340031473_1654666144.907512/0/wire-runtime-3-6-0-sources-jar/jvmMain/com/squareup/wire/internal/RuntimeMessageAdapter.kt

* /opt/cola/permits/1340031473_1654666144.907512/0/wire-runtime-3-6-0-sources-jar/commonMain/com/squareup/wire/FieldEncoding.kt

* /opt/cola/permits/1340031473_1654666144.907512/0/wire-runtime-3-6-0-sources-jar/jvmMain/com/squareup/wire/Message.kt

* /opt/cola/permits/1340031473_1654666144.907512/0/wire-runtime-3-6-0-sources-jar/commonMain/com/squareup/wire/internal/ImmutableList.kt

* /opt/cola/permits/1340031473_1654666144.907512/0/wire-runtime-3-6-0-sources-jar/jvmMain/com/squareup/wire/internal/InternalJvm.kt

* /opt/cola/permits/1340031473_1654666144.907512/0/wire-runtime-3-6-0-sources-jar/commonMain/com/squareup/wire/WireEnum.kt

```
* /opt/cola/permits/1340031473_1654666144.907512/0/wire-runtime-3-6-0-sources-
jar/commonMain/com/squareup/wire/Message.kt
* /opt/cola/permits/1340031473_1654666144.907512/0/wire-runtime-3-6-0-sources-
jar/commonMain/com/squareup/wire/WireField.kt
* /opt/cola/permits/1340031473_1654666144.907512/0/wire-runtime-3-6-0-sources-
jar/jvmMain/com/squareup/wire/Wire.kt
No license file was found, but licenses were detected in source scan.
```

```
/*
* Copyright 2021 Square Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):

```
* /opt/cola/permits/1340031473_1654666144.907512/0/wire-runtime-3-6-0-sources-
jar/jvmMain/com/squareup/wire/internal/OneOfBinding.kt
* /opt/cola/permits/1340031473_1654666144.907512/0/wire-runtime-3-6-0-sources-
jar/jvmMain/com/squareup/wire/internal/FieldOrOneOfBinding.kt
```

1.163 bean-validation-api 2.0.2

1.163.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity

on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one

of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a

result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 2013 Cognifide Limited

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.164 apache-commons-lang 2.6

1.164.1 Available under license :

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or

agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Apache Commons Lang

Copyright 2001-2011 The Apache Software Foundation

This product includes software developed by

The Apache Software Foundation (<http://www.apache.org/>).

1.165 org.apache.groovy:groovy-xml 4.0.1

1.165.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2020 the original author or authors.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/ResponseParserRegistrar.java

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/mapping/JsonbMapper.groovy

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2019 the original author or authors.

*

- * Licensed under the Apache License, Version 2.0 (the "License");
- * you may not use this file except in compliance with the License.
- * You may obtain a copy of the License at
- *
- * <http://www.apache.org/licenses/LICENSE-2.0>
- *
- * Unless required by applicable law or agreed to in writing, software
- * distributed under the License is distributed on an "AS IS" BASIS,
- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.
- */

Found in path(s):

- * /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/specification/LogSpecification.java
- * /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/log/LogRepository.java
- * /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/TestSpecificationImpl.groovy
- * /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/http/ContentTypeExtractor.groovy
- * /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/mapping/Jackson1Mapper.groovy
- * /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/RestAssuredResponseOptionsGroovyImpl.groovy
- * /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/mapper/ObjectMapperSerializationContext.java
- * /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/RedirectSpecificationImpl.groovy
- * /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/mapping/JaxbMapper.groovy
- * /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/filter/SendRequestFilter.groovy
- * /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/builder/MultiPartSpecBuilder.java
- * /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/response/ResponseBody.java
- * /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/filter/Filter.java
- * /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/assertion/HeaderMatcher.groovy
- * /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/http/ContentType.java
- * /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/authentication/PreemptiveOAuth2HeaderScheme.groovy
- * /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/UriValidator.java

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/http/HttpResponseContentTypeFinder.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/http/HttpResponseException.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/multipart/RestAssuredMultiPartEntity.groovy
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/specification/RequestSender.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/http/GZIPEncoding.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/filter/session/SessionFilter.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/config/LogConfig.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/matcher/RestAssuredMatchers.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/builder/ResponseSpecBuilder.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/filter/FilterContext.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/mapper/ObjectMapperDeserializationContext.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/authentication/NoAuthScheme.groovy
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/mapping/ObjectMapperDeserializationContextImpl.groovy
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/MultiValueEntity.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/ValidatableResponseImpl.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/util/IOUtils.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/response/ExtractableResponse.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/util/MatcherErrorMessageBuilder.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/util/SafeExceptionRethrower.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/response/ResponseBodyExtractionOptions.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/authentication/OAuth2Scheme.groovy
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/mapping/JohnzonMapper.groovy
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/config/HeaderConfig.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/TrustAndKeystoreSpec.java

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/TrustAndKeystoreSpecImpl.groovy

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/PreemptiveAuthSpecImpl.groovy

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/MapCreator.groovy

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/spi/AuthFilter.java

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/specification/RequestSenderOptions.java

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/http/StringHashMap.java

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/http/HttpResponseDecorator.java

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/filter/FormAuthFilter.groovy

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/SpecificationMerger.groovy

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/http/HttpRequestFactory.java

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/support/Prettifier.groovy

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/authentication/OAuthSignature.java

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/specification/AuthenticationSpecification.java

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/specification/ProxySpecification.java

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/support/ParameterUpdater.groovy

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/response/ValidatableResponse.java

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/config/RedirectConfig.java

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/LogSpecificationImpl.groovy

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/filter/log/StatusCodeBasedLoggingFilter.java

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/RequestSpecificationImpl.groovy

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/authentication/AuthenticationScheme.groovy

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/http/URIBuilder.java

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/NoParameterValue.groovy

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/response/ResponseOptions.java

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/matcher/ResponseAwareMatcher.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/config/DecoderConfig.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/mapper/ObjectMapperType.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/config/XmlConfig.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/filter/time/TimingFilter.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/specification/FilterableRequestSpecification.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/filter/log/LogDetail.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/http/HTTPBuilder.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/response/Response.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/builder/RequestSpecBuilder.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/mapping/JakartaEEMapper.groovy
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/http/Cookies.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/authentication/CertAuthScheme.groovy
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/authentication/PreemptiveBasicAuthScheme.groovy
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/http/Headers.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/print/RequestPrinter.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/filter/cookie/CookieFilter.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/http/Header.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/serialization/SerializationSupport.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/specification/SpecificationQuerier.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/specification/PreemptiveAuthSpec.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/specification/ResponseLogSpecification.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/mapping/Jackson2Mapper.groovy
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/http/BasicNameValuePairWithNoValueSupport.java

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/support/PathSupport.groovy
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/config/FailureConfig.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/config/RestAssuredConfig.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/matcher/ResponseAwareMatcherComposer.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/mapper/ObjectMapper.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/multipart/MultiPartSpecificationImpl.groovy
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/authentication/ExplicitNoAuthScheme.groovy
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/matcher/xml/XmlXsdMatcher.groovy
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/matcher/DetailedCookieMatcher.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/specification/MultiPartSpecification.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/http/ContentEncodingRegistry.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/authentication/PreemptiveAuthProvider.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/http/Status.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/specification/Argument.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/ContentParser.groovy
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/specification/FilterableResponseSpecification.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/authentication/FormAuthScheme.groovy
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/http/EncoderRegistry.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/mapping/ObjectMapping.groovy
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/specification/RequestSpecification.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/config/ConnectionConfig.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/filter/log/ErrorLoggingFilter.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/http/ContentEncoding.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/ValidatableResponseOptionsImpl.java

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/RestAssuredResponseImpl.java

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/http/ContentTypeSubTypeExtractor.groovy

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/print/ResponsePrinter.java

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/specification/QueryableRequestSpecification.java

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/response/ValidatableResponseLogSpec.java

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/assertion/ResponseTimeMatcher.groovy

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/filter/FilterContextImpl.groovy

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/http/DeflateEncoding.java

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/authentication/FormAuthConfig.java

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/config/HttpClientConfig.java

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/mapping/ObjectMapperSerializationContextImpl.groovy

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/response/ValidatableResponseOptions.java

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/filter/log/ResponseLoggingFilter.java

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/matcher/xml/LoadFromClasspathSupport.groovy

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/multipart/MultiPartInternal.groovy

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/config/EncoderConfig.java

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/response/Validatable.java

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/assertion/StreamVerifier.groovy

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/authentication/BasicAuthScheme.groovy

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/config/SessionConfig.java

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/ResponseSpecificationImpl.groovy

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/support/CloseHttpClientConnectionInputStreamWrapper.java

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/config/ObjectMapperConfig.java

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/filter/log/UrlDecoder.java

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/AuthenticationSpecificationImpl.groovy
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/LogRequestAndResponseOnFailListener.groovy
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/parsing/Parser.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/response/ResponseBodyData.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/config/OAuthConfig.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/support/FileReader.groovy
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/http/Method.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/RestAssuredResponseOptionsImpl.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/RestAssured.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/config/Config.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/config/MultiPartConfig.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/http/ResponseParseException.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/matcher/xml/XmlDtdMatcher.groovy
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/filter/log/RequestLoggingFilter.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/authentication/CertificateAuthSettings.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/http/CharsetExtractor.groovy
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/authentication/OAuthScheme.groovy
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/listener/ResponseValidationFailureListener.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/specification/RedirectSpecification.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/specification/RequestLogSpecification.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/authentication/NTLMAuthScheme.groovy
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/RequestLogSpecificationImpl.groovy
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/config/SSLConfig.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/http/HttpContextDecorator.java

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/proxy/RestAssuredProxySelector.groovy
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/builder/ResponseBuilder.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/filter/OrderedFilter.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/ResponseLogSpecificationImpl.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/config/MatcherConfig.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/mapping/GsonMapper.groovy
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/config/ParamConfig.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/http/AuthConfig.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/config/JsonConfig.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/http/CustomHttpMethod.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/proxy/RestAssuredProxySelectorRoutePlanner.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/NameAndValue.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/specification/ResponseSpecification.java
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/http/BoundaryExtractor.groovy
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/assertion/DetailedCookieAssertion.groovy
* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/http/Cookie.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2022 the original author or authors.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1473459900_1668570861.4992757/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/RestAssuredHttpBuilder.java

1.166 vavr-match 0.9.2

1.166.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*  _ _ _ _ _ _ _ _
 *  \ \ / / \ \ / / _/
 *  \ \ / / ^ \ \ / /
 *  \ \ / / \ \ / /
 *
 * Copyright 2014-2017 Vavr, http://vavr.io
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

* /opt/cola/permits/1274701994_1645234893.05/0/vavr-match-0-9-2-sources-jar/io/vavr/match/UnapplyChecker.java
* /opt/cola/permits/1274701994_1645234893.05/0/vavr-match-0-9-2-sources-jar/io/vavr/match/annotation/Patterns.java
* /opt/cola/permits/1274701994_1645234893.05/0/vavr-match-0-9-2-sources-jar/io/vavr/match/model/MethodModel.java
* /opt/cola/permits/1274701994_1645234893.05/0/vavr-match-0-9-2-sources-jar/io/vavr/match/generator/ImportManager.java
* /opt/cola/permits/1274701994_1645234893.05/0/vavr-match-0-9-2-sources-jar/io/vavr/match/model/TypeParameterModel.java
* /opt/cola/permits/1274701994_1645234893.05/0/vavr-match-0-9-2-sources-jar/io/vavr/match/PatternsProcessor.java
* /opt/cola/permits/1274701994_1645234893.05/0/vavr-match-0-9-2-sources-jar/io/vavr/match/annotation/Unapply.java
* /opt/cola/permits/1274701994_1645234893.05/0/vavr-match-0-9-2-sources-

jar/io/vavr/match/generator/Generator.java
* /opt/cola/permits/1274701994_1645234893.05/0/vavr-match-0-9-2-sources-
jar/io/vavr/match/model/ParameterModel.java
* /opt/cola/permits/1274701994_1645234893.05/0/vavr-match-0-9-2-sources-
jar/io/vavr/match/model/ClassModel.java

1.167 fabric8-:::kubernetes-model-:::- extensions 4.13.3

1.167.1 Available under license :

No license file was found, but licenses were detected in source scan.

Copyright (C) 2015 Red Hat, Inc.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE>

2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Found in path(s):

* /opt/cola/permits/1288519834_1647861786.95/0/kubernetes-model-extensions-4-13-3-jar/META-
INF/maven/io.fabric8/kubernetes-model-extensions/pom.xml

No license file was found, but licenses were detected in source scan.

*

* Copyright (C) 2015 Red Hat, Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.

*

Found in path(s):

* /opt/cola/permits/1288519834_1647861786.95/0/kubernetes-model-extensions-4-13-3-jar/manifest.vm

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0

Bnd-LastModified: 1619068600781

Build-Jdk-Spec: 1.8

Bundle-Description: Java client for Kubernetes and OpenShift

Bundle-DocURL: http://redhat.com

Bundle-License: http://www.apache.org/licenses/LICENSE-2.0.txt

Bundle-ManifestVersion: 2

Bundle-Name: Fabric8 :: Kubernetes Model :: Extensions

Bundle-SymbolicName: io.fabric8.kubernetes-model-extensions

Bundle-Vendor: Red Hat

Bundle-Version: 4.13.3

Created-By: Apache Maven Bundle Plugin

Export-Package: io.fabric8.kubernetes.api.model.extensions;uses:="com.fasterxml.jackson.annotation,com.fasterxml.jackson.databind,com.fasterxml.jackson.databind.annotation,io.fabric8.kubernetes.api.builder,io.fabric8.kubernetes.api.model,io.fabric8.kubernetes.model.annotation";version="4.13.3"

Implementation-Title: Fabric8 :: Kubernetes Model :: Extensions

Implementation-Vendor: Red Hat

Implementation-Version: 4.13.3

Import-Package: com.fasterxml.jackson.annotation;version="[2.11,3)",com.fasterxml.jackson.databind;version="[2.11,3)",com.fasterxml.jackson.databind.annotation;version="[2.11,3)",io.fabric8.kubernetes.api.builder;version="[4.13,5)",io.fabric8.kubernetes.api.model,io.fabric8.kubernetes.model.annotation;version="[4.13,5)"

Require-Capability: osgi.ee;filter="(&(osgi.ee=JavaSE)(version=1.8))"

Specification-Title: Fabric8 :: Kubernetes Model :: Extensions

Specification-Vendor: Red Hat

Specification-Version: 4.13

Tool: Bnd-5.1.1.202006162103

Found in path(s):

* /opt/cola/permits/1288519834_1647861786.95/0/kubernetes-model-extensions-4-13-3-jar/META-INF/MANIFEST.MF

1.168 jsr311-api 1.1.1

1.168.1 Available under license :

/*

- * The contents of this file are subject to the terms
- * of the Common Development and Distribution License

```
* (the "License"). You may not use this file except
* in compliance with the License.
*
* You can obtain a copy of the license at
* http://www.opensource.org/licenses/cddl1.php
* See the License for the specific language governing
* permissions and limitations under the License.
*/
```

1.169 antlr 4.8-1

1.169.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* [The "BSD license"]
* Copyright (c) 2012-2016 Terence Parr
* Copyright (c) 2012-2016 Sam Harwell
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products
* derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/
```

```
tree grammar BlockSetTransformer;
options {
language = Java;
```

```

tokenVocab = ANTLRParser;
ASTLabelType = GrammarAST;
output = AST;
filter = true;
}

@header {
package org.antlr.v4.parse;
import org.antlr.v4.misc.Utils;
import org.antlr.v4.misc.*;
import org.antlr.v4.tool.*;
import org.antlr.v4.tool.ast.*;
import java.util.List;
import java.util.Set;
import java.util.HashSet;
import java.util.ArrayList;
import org.antlr.v4.runtime.misc.IntervalSet;
}

@members {
public String currentRuleName;
public GrammarAST currentAlt;
public Grammar g;
public BlockSetTransformer(TreeNodeStream input, Grammar g) {
    this(input, new RecognizerSharedState());
    this.g = g;
}
}

topdown
: ^(RULE (id=TOKEN_REF|id=RULE_REF) {currentRuleName=$id.text;} .+)
| setAlt
| ebnfBlockSet
| blockSet
;

setAlt
: {inContext("RULE BLOCK")}?
ALT {currentAlt = $start;}
;

// (BLOCK (ALT (+ (BLOCK (ALT INT) (ALT ID))))))
ebnfBlockSet
@after {
GrammarTransformPipeline.setGrammarPtr(g, $tree);
}
: ^(ebnfSuffix blockSet) -> ^(ebnfSuffix ^(BLOCK<BlockAST> ^(ALT<AltAST> blockSet)))
;

```

```

ebnfSuffix
@after {$tree = (GrammarAST)adaptor.dupNode($start);}
: OPTIONAL
| CLOSURE
| POSITIVE_CLOSURE
;

blockSet
@init {
boolean inLexer = Grammar.isTokenName(currentRuleName);
}
@after {
GrammarTransformPipeline.setGrammarPtr(g, $tree);
}
: {inContext("RULE")}? // top-level: rule block and > 1 alt
^(BLOCK ^(alt=ALT elementOptions? {((AltAST)$alt).altLabel==null}? setElement[inLexer]) (^(ALT
elementOptions? setElement[inLexer])))+
-> ^(BLOCK<BlockAST>[$BLOCK.token] ^(ALT<AltAST>[$BLOCK.token,"ALT"] ^(SET[$BLOCK.token,
"SET"] setElement+)))
| {!inContext("RULE")}? // if not rule block and > 1 alt
^(BLOCK ^(ALT elementOptions? setElement[inLexer]) (^(ALT elementOptions? setElement[inLexer])))+
-> ^(SET[$BLOCK.token, "SET"] setElement+)
;

setElement[boolean inLexer]
@after {
GrammarTransformPipeline.setGrammarPtr(g, $tree);
}
: (^(a=STRING_LITERAL elementOptions) {!inLexer ||
CharSupport.getCharValueFromGrammarCharLiteral($a.getText())!=-1}?
| a=STRING_LITERAL {!inLexer || CharSupport.getCharValueFromGrammarCharLiteral($a.getText())!=-1}?
| {!inLexer}?=> ^(TOKEN_REF elementOptions)
| {!inLexer}?=> TOKEN_REF
| {!inLexer}?=> ^(RANGE a=STRING_LITERAL b=STRING_LITERAL)
{CharSupport.getCharValueFromGrammarCharLiteral($a.getText())!=-1 &&
CharSupport.getCharValueFromGrammarCharLiteral($b.getText())!=-1}?
)
;

elementOptions
: ^(ELEMENT_OPTIONS elementOption*)
;

elementOption
: ID
| ^(ASSIGN id=ID v=ID)
| ^(ASSIGN ID v=STRING_LITERAL)

```

| ^(ASSIGN ID v=ACTION)

| ^(ASSIGN ID v=INT)

;

Found in path(s):

* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/parse/BlockSetTransformer.g

No license file was found, but licenses were detected in source scan.

/*

[The "BSD license"]

Copyright (c) 2010 Terence Parr

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

Found in path(s):

* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/parse/ATNBuilder.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (c) 2012-2017 The ANTLR Project. All rights reserved.

* Use of this file is governed by the BSD 3-clause license that

* can be found in the LICENSE.txt file in the project root.

*/

Found in path(s):

* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/tool/ast/GrammarRootAST.java

* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/OutputModelWalker.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/SerializedATN.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/gui/PostScriptDocument.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/tool/ast/GrammarASTWithOptions.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/tool/ToolMessage.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/LL1AltBlock.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/misc/EscapeSequenceParsing.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/Recognizer.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/MatchNotSet.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/tool/LexerGrammar.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/Wildcard.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/chunk/ActionText.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/automata/LexerATNFactory.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/decl/TokenTypeDecl.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/analysis/LeftRecursiveRuleAltInfo.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/Target.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/tool/ast/GrammarASTVisitor.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/parse/GrammarToken.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/OutputModelController.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/tool/ErrorSeverity.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/gui/TestRig.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/VisitorDispatchMethod.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/DispatchMethod.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/tool/ast/RuleRefAST.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/StarBlock.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/tool/GrammarSyntaxMessage.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/chunk/TokenRef.java

* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/tool/ast/RangeAST.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/ArgAction.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/tool/ANTLRToolListener.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/tool/LabelElementPair.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/target/Python3Target.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/ExceptionClause.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/automata/TailEpsilonRemover.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/chunk/TokenPropertyRef_line.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/tool/ErrorType.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/gui/Trees.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/tool/DOTGenerator.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/misc/OrderedHashMap.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/Action.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/ListenerFile.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/LL1PlusBlockSingleAlt.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/dbg.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/chunk/SetNonLocalAttr.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/chunk/ThisRulePropertyRef_ctx.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/misc/CharSupport.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/analysis/AnalysisPipeline.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/decl/ContextTokenGetterDecl.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/tool/ANTLRMessage.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/misc/MutableInt.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/chunk/ActionTemplate.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/BaseListenerFile.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/tool/ast/ActionAST.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/ModelElement.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/semantics/ActionSniffer.java

- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/semantics/RuleCollector.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/decl/ContextGetterDecl.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/chunk/QRetValueRef.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/tool/AttributeDict.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/analysis/LeftRecursiveRuleTransformer.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/OutputModelObject.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/tool/ast/BlockAST.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/tool/AttributeResolver.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/chunk/LabelRef.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/parse/ScopeParser.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/CodeBlockForAlt.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/tool/ast/OptionalBlockAST.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/tool/ErrorMessageManager.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/chunk/TokenPropertyRef_index.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/Parser.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/CodeGenerator.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/misc/Utils.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/chunk/TokenPropertyRef.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/gui/JFileChooserConfirmOverwrite.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/automata/ATNVisitor.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/decl/ContextTokenListIndexedGetterDecl.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/LL1OptionalBlock.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/decl/ContextRuleListIndexedGetterDecl.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/LL1StarBlockSingleAlt.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/tool/ast/PredAST.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/decl/RuleContextListDecl.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/semantics/BasicSemanticChecks.java

- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/target/SwiftTarget.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/chunk/TokenPropertyRef_text.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/decl/AttributeDecl.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/parse/ResyncToEndOfRuleBlock.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/VisitorFile.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/chunk/RulePropertyRef_parser.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/tool/Grammar.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/tool/ast/AltAST.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/CodeGenPipeline.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/parse/v3TreeGrammarException.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/tool/ast/SetAST.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/chunk/RulePropertyRef_stop.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/decl/StructDecl.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/AltBlock.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/Sync.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/tool/ast/TerminalAST.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/gui/BasicFontMetrics.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/tool/ast/RuleAST.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/RuleElement.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/OptionalBlock.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/LL1Choice.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/RuleFunction.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/chunk/RetValRef.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/analysis/LeftRecursionDetector.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/CodeGeneratorExtension.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/gui/TreePostScriptGenerator.java
- * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-

jar/org/antlr/v4/codegen/target/PHPTarget.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
 jar/org/antlr/v4/semantics/UseDefAnalyzer.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
 jar/org/antlr/v4/codegen/model/AddToLabelList.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
 jar/org/antlr/v4/codegen/model/LL1OptionalBlockSingleAlt.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
 jar/org/antlr/v4/codegen/model/BaseVisitorFile.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
 jar/org/antlr/v4/codegen/model/ListenerDispatchMethod.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
 jar/org/antlr/v4/parse/v4ParserException.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
 jar/org/antlr/v4/codegen/model/chunk/LocalRef.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
 jar/org/antlr/v4/tool/GrammarSemanticsMessage.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
 jar/org/antlr/v4/parse/GrammarASTAdaptor.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
 jar/org/antlr/v4/codegen/model/decl/TokenListDecl.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
 jar/org/antlr/v4/tool/ast/RuleElementAST.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
 jar/org/antlr/v4/tool/ast/GrammarASTErrorNode.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
 jar/org/antlr/v4/codegen/model/chunk/NonLocalAttrRef.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
 jar/org/antlr/v4/codegen/model/decl/ElementListDecl.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
 jar/org/antlr/v4/codegen/OutputModelFactory.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
 jar/org/antlr/v4/codegen/model/chunk/RulePropertyRef_start.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
 jar/org/antlr/v4/codegen/model/chunk/ThisRulePropertyRef_start.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/misc/Graph.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
 jar/org/antlr/v4/codegen/model/InvokeRule.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
 jar/org/antlr/v4/codegen/model/chunk/TokenPropertyRef_int.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/Tool.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
 jar/org/antlr/v4/automata/ATNOptimizer.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
 jar/org/antlr/v4/codegen/model/Lexer.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
 jar/org/antlr/v4/codegen/model/PlusBlock.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-

jar/org/antlr/v4/codegen/target/Python2Target.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
 jar/org/antlr/v4/codegen/target/JavaTarget.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
 jar/org/antlr/v4/codegen/model/RuleSempredFunction.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
 jar/org/antlr/v4/codegen/model/chunk/ThisRulePropertyRef_text.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
 jar/org/antlr/v4/codegen/LexerFactory.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
 jar/org/antlr/v4/semantics/SymbolCollector.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
 jar/org/antlr/v4/tool/GrammarParserInterpreter.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
 jar/org/antlr/v4/codegen/model/decl/ContextRuleGetterDecl.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
 jar/org/antlr/v4/parse/ToolANTLRLexer.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
 jar/org/antlr/v4/codegen/model/LexerFile.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
 jar/org/antlr/v4/codegen/model/LabeledOp.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
 jar/org/antlr/v4/codegen/target/CSharpTarget.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
 jar/org/antlr/v4/codegen/model/TestSetInline.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
 jar/org/antlr/v4/codegen/model/decl/ContextTokenListGetterDecl.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/gui/TreeViewer.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/tool/LabelType.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/misc/FrequencySet.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
 jar/org/antlr/v4/tool/ast/GrammarAST.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
 jar/org/antlr/v4/codegen/UnicodeEscapes.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
 jar/org/antlr/v4/automata/ATNPrinter.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
 jar/org/antlr/v4/tool/DefaultToolListener.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
 jar/org/antlr/v4/codegen/model/decl/ContextRuleListGetterDecl.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/tool/Attribute.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
 jar/org/antlr/v4/codegen/model/CodeBlockForOuterMostAlt.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
 jar/org/antlr/v4/tool/ast/QuantifierAST.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
 jar/org/antlr/v4/semantics/SymbolChecks.java
 * /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-

jar/org/antlr/v4/codegen/model/MatchToken.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/codegen/model/ThrowRecognitionException.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/codegen/ParserFactory.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/codegen/DefaultOutputModelFactory.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/codegen/model/ThrowNoViableAlt.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/gui/SystemFontMetrics.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/codegen/model/CaptureNextTokenType.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/tool/LeftRecursionCyclesMessage.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/codegen/model/chunk/SetAttr.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/codegen/model/chunk/TokenPropertyRef_channel.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/automata/ATNFactory.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/unicode/UnicodeDataTemplateController.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/semantics/BlankActionSplitterListener.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/codegen/model/chunk/ThisRulePropertyRef_parser.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/codegen/model/decl/AltLabelStructDecl.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/analysis/LeftRecursiveRuleAnalyzer.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/codegen/model/chunk/ArgRef.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/tool/BuildDependencyGenerator.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/tool/GrammarTransformPipeline.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/gui/TreeLayoutAdaptor.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/tool/GrammarInterpreterRuleContext.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/semantics/SemanticPipeline.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/gui/TreeTextProvider.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/parse/ToolANTLRParser.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-

jar/org/antlr/v4/codegen/model/decl/CodeBlock.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/codegen/target/CppTarget.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/codegen/model/chunk/RulePropertyRef.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/codegen/model/chunk/TokenPropertyRef_type.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/codegen/model/decl/Decl.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/parse/TokenVocabParser.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/codegen/model/decl/TokenDecl.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/semantics/AttributeChecks.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/codegen/target/JavaScriptTarget.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/automata/ParserATNFactory.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/parse/ActionSplitterListener.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/codegen/model/Choice.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/Loop.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/codegen/ActionTranslator.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/tool/ast/NotAST.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/tool/Rule.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/tool/ast/StarBlockAST.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/codegen/model/ParserFile.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/codegen/model/CaptureNextToken.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/codegen/model/chunk/TokenPropertyRef_pos.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/codegen/model/chunk/ListLabelRef.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/codegen/model/ElementFrequenciesVisitor.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/codegen/model/RuleActionFunction.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/codegen/model/LeftRecursiveRuleFunction.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/codegen/model/OutputFile.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/codegen/model/LL1Loop.java

* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/MatchSet.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/tool/Alternative.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/BlankOutputModelFactory.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/target/GoTarget.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/chunk/ThisRulePropertyRef_stop.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/ThrowEarlyExitException.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/decl/RuleContextDecl.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/SemPred.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/tool/LeftRecursiveRule.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/chunk/RulePropertyRef_ctx.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/chunk/ActionChunk.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/SrcOp.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/codegen/model/chunk/RulePropertyRef_text.java
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/tool/ast/PlusBlockAST.java
No license file was found, but licenses were detected in source scan.

/*
* [The "BSD license"]
* Copyright (c) 2012-2016 Terence Parr
* Copyright (c) 2012-2016 Sam Harwell
* Copyright (c) 2015 Janyou
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products
* derived from this software without specific prior written permission.
*
*/

* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
 * IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
 * IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
 * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
 * DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
 * THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
 * THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */

```
SwiftTypeInitMap ::= [
  "Int":"0",
  "Int64":"0",
  "Float":"0.0",
  "Double":"0.0",
  "Bool":"false",
  default:"nil" // anything other than a primitive type is an object
]
```

```
SwiftTypeMap ::= [
  "int":"Int",
  "float":"Float",
  "long":"Int64",
  "double":"Double",
  "bool":"Bool",
  "boolean":"Bool",
  default : key
]
```

// args must be <object-model-object>, <fields-resulting-in-STs>

```
accessLevelOpenOK(obj) ::= "<obj.accessLevel; null=\\"open\\">"
accessLevelNotOpen(obj) ::= "<obj.accessLevel; null=\\"public\\">"
```

```
ParserFile(file, parser, namedActions, contextSuperClass) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
<if(file.genPackage)>
<!package <file.genPackage>;!>
<endif>
<namedActions.header>
import Antlr4

<parser>
>>
```

```
ListenerFile(file, header, namedActions) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
<if(file.genPackage)>
```

```

<!package <file.genPackage>!>
<endif>
<header>
import Antlr4

/**
 * This interface defines a complete listener for a parse tree produced by
 * { @link <file.parserName>}.
 */
<accessLevelNotOpen(file)> protocol <file.grammarName>Listener: ParseTreeListener {
  <file.listenerNames:{lname |
/**
<if(file.listenerLabelRuleNames.(lname))>
 * Enter a parse tree produced by the { @code <lname>}
 * labeled alternative in { @link <file.parserName>#<file.listenerLabelRuleNames.(lname)>}}.
<else>
 * Enter a parse tree produced by { @link <file.parserName>#<lname>}}.
<endif>
- Parameters:
  - ctx: the parse tree
*/
func enter<lname; format="cap">(_ ctx: <file.parserName>.<lname; format="cap">Context)
/**
<if(file.listenerLabelRuleNames.(lname))>
 * Exit a parse tree produced by the { @code <lname>}
 * labeled alternative in { @link <file.parserName>#<file.listenerLabelRuleNames.(lname)>}}.
<else>
 * Exit a parse tree produced by { @link <file.parserName>#<lname>}}.
<endif>
- Parameters:
  - ctx: the parse tree
*/
func exit<lname; format="cap">(_ ctx: <file.parserName>.<lname; format="cap">Context); separator="\n"
}
>>

```

```

BaseListenerFile(file, header, namedActions) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
<if(file.genPackage)>
<!package <file.genPackage>!>
<endif>
<header>

```

```

import Antlr4

```

```

/**
 * This class provides an empty implementation of { @link <file.grammarName>Listener},

```

```

* which can be extended to create a listener which only needs to handle a subset
* of the available methods.
*/
<accessLevelOpenOK(file)> class <file.grammarName>BaseListener: <file.grammarName>Listener {
    <accessLevelNotOpen(file)> init() { \}
    <file.listenerNames: {lname |

/**
 * { @inheritDoc\}
 *
 * \<p>The default implementation does nothing.\</p>
 */
<accessLevelOpenOK(file)> func enter<lname; format="cap">(_ ctx: <file.parserName>.<lname;
format="cap">Context) { \}
/**
 * { @inheritDoc\}
 *
 * \<p>The default implementation does nothing.\</p>
 */
<accessLevelOpenOK(file)> func exit<lname; format="cap">(_ ctx: <file.parserName>.<lname;
format="cap">Context) { \} }; separator="\n">

/**
 * { @inheritDoc\}
 *
 * \<p>The default implementation does nothing.\</p>
 */
<accessLevelOpenOK(file)> func enterEveryRule(_ ctx: ParserRuleContext) { \}
/**
 * { @inheritDoc\}
 *
 * \<p>The default implementation does nothing.\</p>
 */
<accessLevelOpenOK(file)> func exitEveryRule(_ ctx: ParserRuleContext) { \}
/**
 * { @inheritDoc\}
 *
 * \<p>The default implementation does nothing.\</p>
 */
<accessLevelOpenOK(file)> func visitTerminal(_ node: TerminalNode) { \}
/**
 * { @inheritDoc\}
 *
 * \<p>The default implementation does nothing.\</p>
 */
<accessLevelOpenOK(file)> func visitErrorNode(_ node: ErrorNode) { \}
}
>>

```

```

VisitorFile(file, header, namedActions) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
<if(file.genPackage)>
<!package <file.genPackage>;!>
<endif>
<header>
import Antlr4

/**
 * This interface defines a complete generic visitor for a parse tree produced
 * by { @link <file.parserName> }.
 *
 * @param \<T> The return type of the visit operation. Use { @link Void } for
 * operations with no return type.
 */
<accessLevelOpenOK(file)> class <file.grammarName>Visitor\<T>: ParseTreeVisitor\<T> {
  <file.visitorNames:{ lname |
/**
<if(file.visitorLabelRuleNames.(lname))>
 * Visit a parse tree produced by the { @code <lname>\}
 * labeled alternative in { @link <file.parserName>#<file.visitorLabelRuleNames.(lname)>\}.
<else>
 * Visit a parse tree produced by { @link <file.parserName>#<lname>\}.
<endif>
- Parameters:
- ctx: the parse tree
- returns: the visitor result
*/
<accessLevelOpenOK(file)> func visit<lname; format="cap">(_ ctx: <file.parserName>.<lname;
format="cap">Context) -> T {
  fatalError(#function + " must be overridden")
\}
}; separator="\n">
}
>>

```

```

BaseVisitorFile(file, header, namedActions) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
<if(file.genPackage)>
<!package <file.genPackage>;!>
<endif>
<header>
import Antlr4

```

```

/**
 * This class provides an empty implementation of { @link <file.grammarName>Visitor },
 * which can be extended to create a visitor which only needs to handle a subset

```

```

* of the available methods.
*
* @param <T> The return type of the visit operation. Use { @link Void} for
* operations with no return type.
*/
<accessLevelOpenOK(file)> class <file.grammarName>BaseVisitor<T>: AbstractParseTreeVisitor<T> {
  <file.visitorNames:{ lname |
/**
* { @inheritDoc\}
*
* <p>The default implementation returns the result of calling
* { @link #visitChildren\} on { @code ctx\}.\</p>
*/
<accessLevelOpenOK(file)> func visit<lname; format="cap">(_ ctx: <file.parserName>.<lname;
format="cap">Context) -> T? { return visitChildren(ctx) \}}; separator="\n">
}
>>

fileHeader(grammarFileName, ANTLRVersion) ::= <<
// Generated from <grammarFileName; format="java-escape"> by ANTLR <ANTLRVersion>
>>

Parser(parser, funcs, atn, sempredFuncs, superClass) ::= <<
<Parser_(ctor="parser_ctor", ...)>
>>

Parser_(parser, funcs, atn, sempredFuncs, ctor, superClass) ::= <<
</*!@SuppressWarnings({"all", "warnings", "unchecked", "unused", "cast"})!>
<accessLevelOpenOK(parser)> class <parser.name>: <superClass; null="Parser"> {

internal static var _decisionToDFA: [DFA] = {
  var decisionToDFA = [DFA]()
  let length = <parser.name>._ATN.getNumberOfDecisions()
  for i in 0..\<length {
    </*! decisionToDFA[i] = DFA(<parser.name>._ATN.getDecisionState(i)!, i)!>
    decisionToDFA.append(DFA(<parser.name>._ATN.getDecisionState(i)!, i))
  }
  return decisionToDFA
}()

internal static let _sharedContextCache = PredictionContextCache()

<if(parser.tokens)>
<accessLevelNotOpen(parser)>
enum Tokens: Int {
  case EOF = -1, <parser.tokens:{ k | <k> = <parser.tokens.(k)>}; separator=", ", wrap, anchor>
}
<endif>

```

```

<accessLevelNotOpen(parser)>
static let <parser.rules:{r | RULE_<r.name> = <r.index>}; separator=", ", wrap, anchor>

<accessLevelNotOpen(parser)>
static let ruleNames: [String] = [
  <parser.ruleNames:{r | "<r>"}; separator=", ", wrap, anchor>
]

<vocabulary(parser.literalNames, parser.symbolicNames,
  accessLevelNotOpen(parser))>

override <accessLevelOpenOK(parser)>
func getGrammarFileName() -> String { return "<parser.grammarFileName; format="java-escape">" }

override <accessLevelOpenOK(parser)>
func getRuleNames() -> [String] { return <parser.name>.ruleNames }

override <accessLevelOpenOK(parser)>
func getSerializedATN() -> String { return <parser.name>._serializedATN }

override <accessLevelOpenOK(parser)>
func getATN() -> ATN { return <parser.name>._ATN }

<namedActions.members>
<parser:(ctor)()>
<funcs; separator="\n">

<if(sempredFuncs)>
override <accessLevelOpenOK(parser)>
func sempred(_ localctx: RuleContext?, _ ruleIndex: Int, _ predIndex: Int)throws -> Bool {
  switch (ruleIndex) {
    <parser.sempredFuncs.values:{f}
  case <f.ruleIndex>:
    return try <f.name>_sempred(_ localctx?.castdown(<f.ctxType>.self), predIndex)); separator="\n">
    default: return true
  }
  <!return true;!>
}
<sempredFuncs.values; separator="\n">
<endif>

<atn>

<accessLevelNotOpen(parser)>
static let _serializedATN = <parser.name>ATN().jsonString

<accessLevelNotOpen(parser)>

```

```

static let _ATN = ATNDeserializer().deserializeFromJson(_serializedATN)
}
>>

vocabulary(literalNames, symbolicNames, accessLevel) ::= <<
private static let _LITERAL_NAMES: [String?] = [
<literalNames:{t | <t>}; null="nil", separator=", ", wrap, anchor>
]
private static let _SYMBOLIC_NAMES: [String?] = [
<symbolicNames:{t | <t>}; null="nil", separator=", ", wrap, anchor>
]
<accessLevel>
static let VOCABULARY = Vocabulary(_LITERAL_NAMES, _SYMBOLIC_NAMES)
>>

dumpActions(recog, argFuncs, actionFuncs, sempredFuncs) ::= <<
<if(actionFuncs)>
override <accessLevelOpenOK(parser)>
func action(_ _localctx: RuleContext?, _ ruleIndex: Int, _ actionIndex: Int) throws {
switch (ruleIndex) {
<recog.actionFuncs.values:{f}
case <f.ruleIndex>:
<f.name>_action((_localctx as <f.ctxType>?), actionIndex)
}; separator="\n">
default: break
}
}
<actionFuncs.values; separator="\n">
<endif>
<if(sempredFuncs)>
override <accessLevelOpenOK(parser)>
func sempred(_ _localctx: RuleContext?, _ ruleIndex: Int, _ predIndex: Int) throws -> Bool {
switch (ruleIndex) {
<recog.sempredFuncs.values:{f}
case <f.ruleIndex>:
return try <f.name>_sempred(_localctx?.castdown(<f.ctxType>.self), predIndex); separator="\n">
default: return true
}
<!return true;!>
}
<sempredFuncs.values; separator="\n">
<endif>
>>

parser_ctor(p) ::= <<

override <accessLevelOpenOK(parser)>
func getVocabulary() -> Vocabulary {

```

```

    return <p.name>.VOCABULARY
}

override <accessLevelNotOpen(parser)>
init(_ input:TokenStream) throws {
    RuntimeMetaData.checkVersion("4.8", RuntimeMetaData.VERSION)
    try super.init(input)
    _interp = ParserATNSimulator(self,<p.name>._ATN,<p.name>._decisionToDFA,
<parser.name>._sharedContextCache)
}

>>

/* This generates a private method since the actionIndex is generated, making an
* overriding implementation impossible to maintain.
*/
RuleActionFunction(r, actions) ::= <<
private func <r.name>_action(_ _localctx: <r.ctxType>?, _ actionIndex: Int) {
    switch (actionIndex) {
    <actions:{index|
case <index>:
    <actions.(index)>
    }; separator="\n">
    default: break
    }
}
}
>>

/* This generates a private method since the predIndex is generated, making an
* overriding implementation impossible to maintain.
*/
RuleSempredFunction(r, actions) ::= <<
private func <r.name>_sempred(_ _localctx: <r.ctxType>!, _ predIndex: Int) throws -> Bool {
    switch (predIndex) {
    <actions:{index|
    case <index>:return <actions.(index)>}; separator="\n">
    default: return true
    }
    <!return true;!>
}
}
>>

RuleFunction(currentRule,args,code,locals,ruleCtx,altLabelCtxs,namedActions,finallyAction,postamble,exceptions)
::= <<
<ruleCtx>
<altLabelCtxs:{l | <altLabelCtxs.(l)>}; separator="\n">
@discardableResult
<if(currentRule.modifiers)><currentRule.modifiers:{f | <f> }><else> <accessLevelOpenOK(parser)> func

```

```

<endif><currentRule.name>(<if(first(args))>_ <endif><args; separator=", _">) throws -> <currentRule.ctxType> {
  var _localctx: <currentRule.ctxType> = <currentRule.ctxType>(_ctx, getState()<currentRule.args:{a | ,
<a.name>}>)
  try enterRule(_localctx, <currentRule.startState>, <parser.name>.RULE_<currentRule.name>)
  <namedActions.init>
  <locals; separator="\n">
  defer {
    <finallyAction>
    try! exitRule()
  }
  do {
<if(currentRule.hasLookaheadBlock)>
  var _alt:Int
<endif>
  <code>
  <postamble; separator="\n">
  <namedActions.after>
  }
  <if(exceptions)>
  <exceptions; separator="\n">
  <else>
  catch ANTLRException.recognition(let re) {
    _localctx.exception = re
    _errHandler.reportError(self, re)
    try _errHandler.recover(self, re)
  }
  <endif>

  return _localctx
}
>>

```

```

LeftRecursiveRuleFunction(currentRule,args,code,locals,ruleCtx,altLabelCtxs,
  namedActions,finallyAction,postamble) ::=
<<

```

```

<ruleCtx>
<altLabelCtxs:{l | <altLabelCtxs.(l)>}; separator="\n">

<if(currentRule.modifiers)><currentRule.modifiers:{f | <f> }><else> <accessLevelNotOpen(parser)> final <endif>
func <currentRule.name>(<if(first(args))>_ <endif><args; separator=", _">) throws -> <currentRule.ctxType> {
  return try <currentRule.name>(0<currentRule.args:{a | , <a.name>}>)
}
@discardableResult
private func <currentRule.name>(_ p<args:{a | , <a>}>: Int) throws -> <currentRule.ctxType> {
  let _parentctx: ParserRuleContext? = _ctx
  var _parentState: Int = getState()
  var _localctx: <currentRule.ctxType> = <currentRule.ctxType>(_ctx, _parentState<currentRule.args:{a | ,

```

```

<a.name>}>)
var _prevctx: <currentRule.ctxType> = _localctx
var _startState: Int = <currentRule.startState>
try enterRecursionRule(_localctx, <currentRule.startState>, <parser.name>.RULE_<currentRule.name>, _p)
<namedActions.init>
<locals; separator="\n">
defer {
  <finallyAction>
  try! unrollRecursionContexts(_parentctx)
}
do {
<if(currentRule.hasLookaheadBlock)>
  var _alt: Int
<endif>
<code>
  <postamble; separator="\n">
  <namedActions.after>
}
catch ANTLRException.recognition(let re) {
  _localctx.exception = re
  _errHandler.reportError(self, re)
  try _errHandler.recover(self, re)
}

return _localctx;
}
>>

```

```

CodeBlockForOuterMostAlt(currentOuterMostAltCodeBlock, locals, preamble, ops) ::= <<
<if(currentOuterMostAltCodeBlock.altLabel)>_localctx = <currentOuterMostAltCodeBlock.altLabel;
format="cap">Context(_localctx);<endif>
try enterOuterAlt(_localctx, <currentOuterMostAltCodeBlock.alt.altNum>)
<CodeBlockForAlt(currentAltCodeBlock=currentOuterMostAltCodeBlock, ...)>
>>

```

```

CodeBlockForAlt(currentAltCodeBlock, locals, preamble, ops) ::= <<
<!/{!>
<locals; separator="\n">
<preamble; separator="\n">
<ops; separator="\n">
<!/{}!>
>>

```

```

LL1AltBlock(choice, preamble, alts, error) ::= <<
setState(<choice.stateNumber>)
try _errHandler.sync(self)
<if(choice.label)><labelref(choice.label)> = try _input.LT(1)<endif>
<preamble; separator="\n">

```

```

switch (<parser.name>.Tokens(rawValue: try _input.LA(1))!) {
<choice.altLook,alts:{look,alt | <cases(ttypes=look)>
<alt>
break}; separator="\n">
default:
<error>
}
>>

```

```

LL1OptionalBlock(choice, alts, error) ::= <<
setState(<choice.stateNumber>)
try _errHandler.sync(self)
switch (<parser.name>.Tokens(rawValue: try _input.LA(1))!) {
<choice.altLook,alts:{look,alt| <cases(ttypes=look)>
<alt>
break}; separator="\n">
default:
break
}
>>

```

```

LL1OptionalBlockSingleAlt(choice, expr, alts, preamble, error, followExpr) ::= <<
setState(<choice.stateNumber>)
try _errHandler.sync(self)
<preamble; separator="\n">
if (<expr>) {
<alts; separator="\n">
}
<!else if ( !(<followExpr> ) <error>!>
>>

```

```

LL1StarBlockSingleAlt(choice, loopExpr, alts, preamble, iteration) ::= <<
setState(<choice.stateNumber>)
try _errHandler.sync(self)
<preamble; separator="\n">
while (<loopExpr>) {
<alts; separator="\n">
setState(<choice.loopBackStateNumber>)
try _errHandler.sync(self)
<iteration>
}
>>

```

```

LL1PlusBlockSingleAlt(choice, loopExpr, alts, preamble, iteration) ::= <<
setState(<choice.blockStartStateNumber>) <! alt block decision !>
try _errHandler.sync(self)
<preamble; separator="\n">
repeat {

```

```

<alts; separator="\n">
  setState(<choice.stateNumber>); <! loopback/exit decision !>
  try _errHandler.sync(self)
  <iteration>
} while (<loopExpr>)
>>

// LL(*) stuff

AltBlock(choice, preamble, alts, error) ::= <<
  setState(<choice.stateNumber>)
  try _errHandler.sync(self)
  <if(choice.label)><labelref(choice.label)> = try _input.LT(1)<endif>
  <preamble; separator="\n">
  switch(try getInterpreter().adaptivePredict(_input,<choice.decision>, _ctx)) {
  <alts:{alt |
  case <i>:
    <alt>
    break }; separator="\n">
  default: break
  }
  >>

OptionalBlock(choice, alts, error) ::= <<
  setState(<choice.stateNumber>)
  try _errHandler.sync(self)
  switch (try getInterpreter().adaptivePredict(_input,<choice.decision>, _ctx)) {
  <alts:{alt |
  case <i><if(!choice.ast.greedy)>+1<endif>:
    <alt>
    break }; separator="\n">
  default: break
  }
  >>

StarBlock(choice, alts, sync, iteration) ::= <<
  setState(<choice.stateNumber>)
  try _errHandler.sync(self)
  _alt = try getInterpreter().adaptivePredict(_input,<choice.decision>, _ctx)
  while (_alt != <choice.exitAlt> && _alt != ATN.INVALID_ALT_NUMBER) {
  if ( _alt==1<if(!choice.ast.greedy)>+1<endif> ) {
    <iteration>
    <alts> <! should only be one !>
  }
  setState(<choice.loopBackStateNumber>)
  try _errHandler.sync(self)
  _alt = try getInterpreter().adaptivePredict(_input,<choice.decision>, _ctx)
  }

```

>>

```
PlusBlock(choice, alts, error) ::= <<
setState(<choice.blockStartStateNumber>); <! alt block decision !>
try _errHandler.sync(self)
_alt = 1<if(!choice.ast.greedy)>+1<endif>;
repeat {
  switch (_alt) {
    <alts:{alt|
case <i><if(!choice.ast.greedy)>+1<endif>:
    <alt>
break}; separator="\n">
  default:
    <error>
  }
  setState(<choice.loopBackStateNumber>); <! loopback/exit decision !>
  try _errHandler.sync(self)
  _alt = try getInterpreter().adaptivePredict(_input,<choice.decision>,_ctx)
} while (_alt != <choice.exitAlt> && _alt != ATN.INVALID_ALT_NUMBER)
>>
```

```
Sync(s) ::= "sync(<s.expecting.name>);"
```

```
ThrowNoViableAlt(t) ::= "throw ANTLRException.recognition(e: NoViableAltException(self))"
```

```
TestSetInline(s) ::= <<
<!<s.bitsets:{bits | <if(rest(rest(bits.ttypes)))><bitsetBitfieldComparison(s, bits)><else><bitsetInlineComparison(s,
bits)><endif>; separator=" || ">!>
//closure
{ () -> Bool in
  <if(rest(s.bitsets)>var<else>let<endif> testSet: Bool = <first(s.bitsets):{bits |
<if(rest(rest(bits.ttypes)))><bitsetBitfieldComparison(s, bits)><else><bitsetInlineComparison(s, bits)><endif>}>
  <rest(s.bitsets):{bits | testSet = testSet || <if(rest(rest(bits.ttypes)))><bitsetBitfieldComparison(s,
bits)><else><bitsetInlineComparison(s, bits)><endif>; separator="\n">
  return testSet
}()
>>
```

```
// Java language spec 15.19 - shift operators mask operands rather than overflow to 0... need range test
testShiftInRange(shiftAmount) ::= <<
((<shiftAmount>) & ~0x3f) == 0
>>
```

```
// produces smaller bytecode only when bits.ttypes contains more than two items
bitsetBitfieldComparison(s, bits) ::= <<
<!(<testShiftInRange({<offsetShift(s.varName, bits.shift)>})> && ((1 \<< <offsetShift(s.varName, bits.shift)>) &
(<bits.ttypes:{ttype | (1 \<< <offsetShift(ttype, bits.shift)>}); separator=" | ">)) != 0)!>
{ () -> Bool in
```

```

    <! let test: Bool = (<testShiftInRange({<offsetShift(s.varName, bits.shift)>}>>)>!>
    <!var temp: Int64 = Int64(<offsetShift(s.varName, bits.shift)>)>!>
    <!temp = (temp < 0) ? (64 + (temp % 64)) : (temp % 64)>!>
    <!let test1: Int64 = (Int64(1) <<< temp)>!>
    <!var test2: Int64 = (<first(bits.ttypes):{ ttype | Utils.bitLeftShift(<offsetShift(parserName(ttype), bits.shift)>)}>>)>!>
    <!<rest(bits.ttypes):{ ttype | test2 = test2 | Utils.bitLeftShift(<offsetShift(parserName(ttype), bits.shift)>)}>>
separator="\n">!>
    let testArray: [Int] = [<s.varName>, <bits.ttypes:{ ttype |<parserName(ttype)>}; separator=",">]
    <!var test2: Int64 = Utils.testBitLeftShiftArray(testArray)>!>
    return Utils.testBitLeftShiftArray(testArray, <bits.shift>)
  }()
>>

```

```

isZero ::= [
  "0": true,
  default: false
]
parserName(ttype) ::= <%
<parser.name>.Tokens.<ttype>.rawValue
%>
offsetShift(shiftAmount, offset) ::= <%
<if(!isZero.(offset))><shiftAmount> - <offset><else><shiftAmount><endif>
%>

```

```

// produces more efficient bytecode when bits.ttypes contains at most two items
bitsetInlineComparison(s, bits) ::= <%
<bits.ttypes:{ ttype | <s.varName> == <parser.name>.Tokens.<ttype>.rawValue }; separator=" || ">
%>

```

```

cases(ttypes) ::= <<
<trunc(ttypes): { t | case .<t>:fallthrough } ; separator="\n">
<last(ttypes): { t | case .<t>: } ; separator="\n">
>>

```

```

InvokeRule(r, argExprsChunks) ::= <<
  setState(<r.stateNumber>)
  <if(r.labels)>
  try {
    let assignmentValue = try
    <r.name>(<if(r.ast.options.p)><r.ast.options.p><if(argExprsChunks)>,<endif><endif><argExprsChunks>)
    <r.labels:{l | <labelref(l)> = assignmentValue } ; separator="\n">
    }()
  <else>try
  <r.name>(<if(r.ast.options.p)><r.ast.options.p><if(argExprsChunks)>,<endif><endif><argExprsChunks>)<endif>
>>

```

```

MatchToken(m) ::= <<
  setState(<m.stateNumber>)

```

```

<if(m.labels)>
try {
  let assignmentValue = try match(<parser.name>.Tokens.<m.name>.rawValue)
  <m.labels:{1 | <labelref(1)> = assignmentValue} ; separator="\n">
  }()
<else>try match(<parser.name>.Tokens.<m.name>.rawValue)<endif>
>>

MatchSet(m, expr, capture) ::= "<CommonSetStuff(m, expr, capture, false)>"

MatchNotSet(m, expr, capture) ::= "<CommonSetStuff(m, expr, capture, true)>"

CommonSetStuff(m, expr, capture, invert) ::= <<
setState(<m.stateNumber>)
<if(m.labels)><m.labels:{1 | <labelref(1)> = }>try _input.LT(1)<endif>
<capture>
if (<if(invert)><m.varName> \<= 0 || <else>!<endif>(<expr>)) {
  <if(m.labels)><m.labels:{1 | <labelref(1)> = }><endif>try _errHandler.recoverInline(self)<if(m.labels)> as
  Token<endif>
}
else {
  _errHandler.reportMatch(self)
  try consume()
}
>>

Wildcard(w) ::= <<
setState(<w.stateNumber>)
<if(w.labels)><w.labels:{1 | <labelref(1)> = }><endif>try matchWildcard();
>>

// ACTION STUFF

Action(a, foo, chunks) ::= "<chunks>"

ArgAction(a, chunks) ::= "<chunks>"

SemPred(p, chunks, failChunks) ::= <<
setState(<p.stateNumber>)
if (!(<chunks>)) {
  throw ANTLRException.recognition(e:FailedPredicateException(self, <p.predicate><if(failChunks)>,
  <failChunks><elseif(p.msg)>, <p.msg><endif>))
}
>>

ExceptionClause(e, catchArg, catchAction) ::= <<
catch (<catchArg>) {
  <catchAction>
}

```

```

}
>>

// lexer actions are not associated with model objects

LexerSkipCommand() ::= "skip()"
LexerMoreCommand() ::= "more()"
LexerPopModeCommand() ::= "popMode()"

LexerTypeCommand(arg) ::= "_type = <arg>"
LexerChannelCommand(arg) ::= "_channel = <arg>"
LexerModeCommand(arg) ::= "_mode = <arg>"
LexerPushModeCommand(arg) ::= "pushMode(<arg>)"

ActionText(t) ::= "<t.text>"
ActionTemplate(t) ::= "<t.st>"
ArgRef(a) ::= "_localctx.<a.name>"
LocalRef(a) ::= "_localctx.<a.name>"
RetValRef(a) ::= "_localctx.<a.name>"
QRetValRef(a) ::= "<ctx(a)>.<a.dict>.<a.name>"
/** How to translate $tokenLabel */
TokenRef(t) ::= "<ctx(t)>.<t.name>"
LabelRef(t) ::= "<ctx(t)>.<t.name>"
ListLabelRef(t) ::= "<ctx(t)>.<ListLabelName(t.name)>"
SetAttr(s,rhsChunks) ::= "<ctx(s)>.<s.name> = <rhsChunks>"

TokenLabelType() ::= "<file.TokenLabelType; null={Token}>"
InputSymbolType() ::= "<file.InputSymbolType; null={Token}>"

TokenPropertyRef_text(t) ::= "<ctx(t)>.<t.label> != nil ? <ctx(t)>.<t.label>!.getText()! : \"\""
TokenPropertyRef_type(t) ::= "<ctx(t)>.<t.label> != nil ? <ctx(t)>.<t.label>!.getType() : 0)"
TokenPropertyRef_line(t) ::= "<ctx(t)>.<t.label> != nil ? <ctx(t)>.<t.label>!.getLine() : 0)"
TokenPropertyRef_pos(t) ::= "<ctx(t)>.<t.label> != nil ? <ctx(t)>.<t.label>!.getCharPositionInLine() : 0)"
TokenPropertyRef_channel(t) ::= "<ctx(t)>.<t.label> != nil ? <ctx(t)>.<t.label>!.getChannel() : 0)"
TokenPropertyRef_index(t) ::= "<ctx(t)>.<t.label> != nil ? <ctx(t)>.<t.label>!.getTokenIndex() : 0)"
TokenPropertyRef_int(t) ::= "<ctx(t)>.<t.label> != nil ? Int(<ctx(t)>.<t.label>!.getText()!) : 0)"

RulePropertyRef_start(r) ::= "<ctx(r)>.<r.label> != nil ? (<ctx(r)>.<r.label>!.start?.description ?? \"\") : \"\""
RulePropertyRef_stop(r) ::= "<ctx(r)>.<r.label> != nil ? (<ctx(r)>.<r.label>!.stop?.description ?? \"\") : \"\""
RulePropertyRef_text(r) ::= "<ctx(r)>.<r.label> != nil ? try
_input.getText(<ctx(r)>.<r.label>!.start,<ctx(r)>.<r.label>!.stop) : \"\""
RulePropertyRef_ctx(r) ::= "<ctx(r)>.<r.label>"
RulePropertyRef_parser(r) ::= "self"

ThisRulePropertyRef_start(r) ::= "_localctx.start"
ThisRulePropertyRef_stop(r) ::= "_localctx.stop"
ThisRulePropertyRef_text(r) ::= "(try _input.getText(_localctx.start, try _input.LT(-1)))"
ThisRulePropertyRef_ctx(r) ::= "_localctx"

```

```

ThisRulePropertyRef_parser(r) ::= "self"

NonLocalAttrRef(s) ::= "((<s.ruleName; format=\"cap\">Context)getInvokingContext(<s.ruleIndex>)).<s.name>"
SetNonLocalAttr(s, rhsChunks) ::=
"((<s.ruleName; format=\"cap\">Context)getInvokingContext(<s.ruleIndex>)).<s.name> = <rhsChunks>"

AddToLabelList(a) ::= "<ctx(a.label)>.<a.listName>.append(<labelref(a.label)>)"

TokenDecl(t) ::= "<t.name>: <SwiftTypeMap.(TokenLabelType())>!"
TokenTypeDecl(t) ::= "var <t.name>: Int = 0"
TokenListDecl(t) ::= "<t.name>: [Token] = [Token]()"
RuleContextDecl(r) ::= "<r.name>: <r.ctxName>!"
RuleContextListDecl(rdecl) ::= "<rdecl.name>: [<rdecl.ctxName>] = [<rdecl.ctxName>]()"

ContextTokenGetterDecl(t) ::= <<
<accessLevelOpenOK(parser)>
func <t.name>() -> TerminalNode? {
    return getToken(<parser.name>.Tokens.<t.name>.rawValue, 0)
}
>>
ContextTokenListGetterDecl(t) ::= <<
<accessLevelOpenOK(parser)>
func <t.name>() -> [TerminalNode] {
    return getTokens(<parser.name>.Tokens.<t.name>.rawValue)
}
>>
ContextTokenListIndexedGetterDecl(t) ::= <<
<accessLevelOpenOK(parser)>
func <t.name>(_ i: Int) -> TerminalNode? {
    return getToken(<parser.name>.Tokens.<t.name>.rawValue, i)
}
>>
ContextRuleGetterDecl(r) ::= <<
<accessLevelOpenOK(parser)>
func <r.name>() -> <r.ctxName>? {
    return getRuleContext(<r.ctxName>.self, 0)
}
>>
ContextRuleListGetterDecl(r) ::= <<
<accessLevelOpenOK(parser)>
func <r.name>() -> [<r.ctxName>] {
    return getRuleContexts(<r.ctxName>.self)
}
>>
ContextRuleListIndexedGetterDecl(r) ::= <<
<accessLevelOpenOK(parser)>
func <r.name>(_ i: Int) -> <r.ctxName>? {
    return getRuleContext(<r.ctxName>.self, i)
}

```

```
}  
>>
```

```
LexerRuleContext() ::= "RuleContext"
```

```
/** The rule context name is the rule followed by a suffix; e.g.,  
 * r becomes rContext.  
 */
```

```
RuleContextNameSuffix() ::= "Context"
```

```
ImplicitTokenLabel(tokenName) ::= "_<tokenName>"
```

```
ImplicitRuleLabel(ruleName) ::= "_<ruleName>"
```

```
ImplicitSetLabel(id) ::= "_tset<id>"
```

```
ListLabelName(label) ::= "<label>"
```

```
CaptureNextToken(d) ::= "<d.varName> = try _input.LT(1)"
```

```
CaptureNextTokenType(d) ::= "<d.varName> = try _input.LA(1)"
```

```
StructDecl(struct,ctorAttrs,attrs,getters,dispatchMethods,interfaces,extensionMembers,  
    superClass={ParserRuleContext}) ::= <<
```

```
<accessLevelNotOpen(parser)> class <struct.name>:
```

```
<if(contextSuperClass)><contextSuperClass><else>ParserRuleContext<endif><if(interfaces)>, <interfaces;  
separator=", "><endif> {
```

```
<attrs:{a | <accessLevelOpenOK(parser)> var <a>}; separator="\n">
```

```
<getters:{g | <g>}; separator="\n">
```

```
<! <if(ctorAttrs)> <accessLevelNotOpen(parser)> init(_ parent: ParserRuleContext,_ invokingState: Int) {  
super.init(parent, invokingState) }<endif> !>
```

```
<if(ctorAttrs)>
```

```
<accessLevelNotOpen(parser)> convenience init(_ parent: ParserRuleContext?, _ invokingState: Int<ctorAttrs:{a |,  
_ <a>}>) {
```

```
self.init(parent, invokingState)
```

```
<struct.ctorAttrs:{a | self.<a.name> = <a.name>;}; separator="\n">
```

```
}
```

```
<endif>
```

```
override <accessLevelOpenOK(parser)>
```

```
func getRuleIndex() -> Int {
```

```
return <parser.name>.RULE_<struct.derivedFromName>
```

```
}
```

```
<if(struct.provideCopyFrom && struct.attrs)> <! don't need copy unless we have subclasses !>
```

```
<accessLevelOpenOK(parser)>
```

```
override func copyFrom(_ ctx_: ParserRuleContext) {
```

```
super.copyFrom(ctx_)
```

```
let ctx = ctx_ as! <struct.name>
```

```
<struct.attrs:{a | self.<a.name> = ctx.<a.name>;}; separator="\n">
```

```
}
```

```
<endif>
```

```

<dispatchMethods; separator="\n">
<extensionMembers; separator="\n">
}
>>

```

```

AltLabelStructDecl(struct,attrs, getters,dispatchMethods) ::= <<
<accessLevelNotOpen(parser)> class <struct.name>: <currentRule.name; format="cap">Context {
<attrs:{a | <accessLevelNotOpen(parser)> var <a>}; separator="\n">
<getters:{g | <g>}; separator="\n">

```

```

<accessLevelNotOpen(parser)>
init(_ ctx: <currentRule.name; format="cap">Context) {
super.init()
copyFrom(ctx)
}
<dispatchMethods; separator="\n">
}
>>

```

```

ListenerDispatchMethod(method) ::= <<
override <accessLevelOpenOK(parser)>
func <if(method.isEnter)>enter<else>exit<endif>Rule(_ listener: ParseTreeListener) {
if let listener = listener as? <parser.grammarName>Listener {
listener.<if(method.isEnter)>enter<else>exit<endif><<struct.derivedFromName; format="cap">(self)
}
}
>>

```

```

VisitorDispatchMethod(method) ::= <<
override <accessLevelOpenOK(parser)>
func accept<T>(<_ visitor: ParseTreeVisitor<T>>) -> T? {
if let visitor = visitor as? <parser.grammarName>Visitor {
return visitor.visit<struct.derivedFromName; format="cap">(self)
}
else if let visitor = visitor as? <parser.grammarName>BaseVisitor {
return visitor.visit<struct.derivedFromName; format="cap">(self)
}
else {
return visitor.visitChildren(self)
}
}
>>

```

```

AttributeDecl(d) ::= "<d.name>: <SwiftTypeMap.(d.type)><if(d.initValue)> = <d.initValue><else>!<endif>"

```

```

/** If we don't know location of label def x, use this template (_localctx as! <x.ctx.name> */
labelref(x) ::= "<if(!x.isLocal)>_localctx.castdown(<x.ctx.name>.self).<endif><x.name>"

```

```

/** For any action chunk, what is correctly-typed context struct ptr? */
ctx(actionChunk) ::= "_localctx.castdown(<actionChunk.ctx.name>.self)"

// used for left-recursive rules
recRuleAltPredicate(ruleName,opPrec) ::= "precpred(_ctx, <opPrec>)"
recRuleSetReturnAction(src,name) ::= "$<name>=<src>.<name>"
recRuleSetStopToken() ::= "_ctx!.stop = try _input.LT(-1)"

recRuleAltStartAction(ruleName, ctxName, label, isListLabel) ::= <<
  _localctx = <ctxName>Context(_parentctx, _parentState);
  <if(label)>
  <if(isListLabel)>
  _localctx.<label>.append(_prevctx)
  <else>
  _localctx.<label> = _prevctx
  <endif>
  <endif>
  <if(label)>_localctx.<label> = _prevctx;<endif>
  try pushNewRecursionContext(_localctx, _startState, <parser.name>.RULE_<ruleName>)
  >>

recRuleLabeledAltStartAction(ruleName, currentAltLabel, label, isListLabel) ::= <<
  _localctx = <currentAltLabel; format="cap">Context( <ruleName; format="cap">Context(_parentctx,
  _parentState))
  <if(label)>
  <if(isListLabel)>
  (_localctx as! <currentAltLabel; format="cap">Context).<label>.append(_prevctx)
  <else>
  (_localctx as! <currentAltLabel; format="cap">Context).<label> = _prevctx
  <endif>
  <endif>
  try pushNewRecursionContext(_localctx, _startState, <parser.name>.RULE_<ruleName>)
  >>

recRuleReplaceContext(ctxName) ::= <<
  _localctx = <ctxName>Context(_localctx)
  _ctx = _localctx
  _prevctx = _localctx
  >>

recRuleSetPrevCtx() ::= <<
  if _parseListeners != nil {
  try triggerExitRuleEvent()
  }
  _prevctx = _localctx
  >>

```

```

LexerFile(lexerFile, lexer, namedActions) ::= <<
<fileHeader(lexerFile.grammarFileName, lexerFile.ANTLRVersion)>
<if(lexerFile.genPackage)>
<!package <lexerFile.genPackage>;!>
<endif>
<namedActions.header>
import Antlr4

<lexer>
>>
Lexer(lexer, atn, actionFuncs, sempredFuncs, superClass) ::= <<
<accessLevelOpenOK(lexer)> class <lexer.name>: <superClass; null="Lexer"> {

internal static var _decisionToDFA: [DFA] = {
    var decisionToDFA = [DFA]()
    let length = <lexer.name>._ATN.getNumberOfDecisions()
    for i in 0..<length {
        <! decisionToDFA[i] = DFA(<lexer.name>._ATN.getDecisionState(i)!, i);!>
        decisionToDFA.append(DFA(<lexer.name>._ATN.getDecisionState(i)!, i))
    }
    return decisionToDFA
}()

internal static let _sharedContextCache = PredictionContextCache()

<accessLevelNotOpen(lexer)>
static let <lexer.tokens>:{k | <k>=<lexer.tokens.(k)>}; separator=", ", wrap, anchor>

<if(lexer.channels)>
<accessLevelNotOpen(lexer)>
static let <lexer.channels>:{k | <k>=<lexer.channels.(k)>}; separator=", ", wrap, anchor>
<endif>
<if(rest(lexer.modes))>
<accessLevelNotOpen(lexer)>
static let <rest(lexer.modes)>:{m | <m>=<i>}; separator=", ", wrap, anchor>
<endif>
<accessLevelNotOpen(lexer)>
static let channelNames: [String] = [
    "DEFAULT_TOKEN_CHANNEL", "HIDDEN"<if (lexer.channels)>, <lexer.channels:{c | "<c">"}; separator=", ",
wrap, anchor><endif>
]

<accessLevelNotOpen(lexer)>
static let modeNames: [String] = [
    <lexer.modes:{m | "<m">"}; separator=", ", wrap, anchor>
]

<accessLevelNotOpen(lexer)>

```

```

static let ruleNames: [String] = [
  <lexer.ruleNames:{r | "<r>"}; separator=", ", wrap, anchor>
]

<vocabulary(lexer.literalNames, lexer.symbolicNames,
  accessLevelNotOpen(lexer))>

<namedActions.members>

override <accessLevelOpenOK(lexer)>
func getVocabulary() -> Vocabulary {
  return <lexer.name>.VOCABULARY
}

<accessLevelNotOpen(lexer)>
required init(_ input: CharStream) {
  RuntimeMetaData.checkVersion("<lexerFile.ANTLRVersion>", RuntimeMetaData.VERSION)
  super.init(input)
  _interp = LexerATNSimulator(self, <lexer.name>._ATN, <lexer.name>._decisionToDFA,
<lexer.name>._sharedContextCache)
}

override <accessLevelOpenOK(lexer)>
func getGrammarFileName() -> String { return "<lexer.grammarFileName>" }

override <accessLevelOpenOK(lexer)>
func getRuleNames() -> [String] { return <lexer.name>.ruleNames }

override <accessLevelOpenOK(lexer)>
func getSerializedATN() -> String { return <lexer.name>._serializedATN }

override <accessLevelOpenOK(lexer)>
func getChannelNames() -> [String] { return <lexer.name>.channelNames }

override <accessLevelOpenOK(lexer)>
func getModeNames() -> [String] { return <lexer.name>.modeNames }

override <accessLevelOpenOK(lexer)>
func getATN() -> ATN { return <lexer.name>._ATN }

<dumpActions(lexer, "", actionFuncs, sempredFuncs)>
<atn>

<accessLevelNotOpen(lexer)>
static let _serializedATN: String = <lexer.name>ATN().jsonString

<accessLevelNotOpen(lexer)>
static let _ATN: ATN = ATNDeserializer().deserializeFromJson(_serializedATN)

```

```

}
>>

/** Don't need to define anything. The tool generates a XParserATN.swift file (and same for lexer)
 * which is referenced from static field _serializedATN. This json string is passed to
 * deserializeFromJson(). Note this is not the "serialization as array of ints" that other targets
 * do. It is more or less the output of ATNPrinter which gets read back in.
 */
SerializedATN(model) ::= <<
>>

/** Using a type to init value map, try to init a type; if not in table
 * must be an object, default value is "null".
 */
initValue(typeName) ::= <<
<SwiftTypeInitMap.(typeName)>
>>

codeFileExtension() ::= ".swift"

```

Found in path(s):

```

* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/tool/templates/codegen/Swift/Swift.stg

```

No license file was found, but licenses were detected in source scan.

```

/*

```

[The "BSD licence"]

Copyright (c) 2006 Kay Roepke

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY

THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

/*

This file contains the actual layout of the messages emitted by ANTLR.

The text itself is coming out of the languages/*stg files, according to the chosen locale.

This file contains the default format ANTLR uses.

*/

location(file, line, column) ::= "<file>(<line>,<column>)"

message(id, text) ::= "error <id> : <text>"

report(location, message, type) ::= "<location> : <type> <message.id> : <message.text>"

wantsSingleLineMessage() ::= "true"

Found in path(s):

* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/tool/templates/messages/formats/vs2005.stg

No license file was found, but licenses were detected in source scan.

/*

* [The "BSD license"]

* Copyright (c) 2012-2016 Terence Parr

* Copyright (c) 2012-2016 Sam Harwell

* All rights reserved.

*

* Redistribution and use in source and binary forms, with or without

* modification, are permitted provided that the following conditions

* are met:

*

* 1. Redistributions of source code must retain the above copyright

* notice, this list of conditions and the following disclaimer.

* 2. Redistributions in binary form must reproduce the above copyright

* notice, this list of conditions and the following disclaimer in the

* documentation and/or other materials provided with the distribution.

* 3. The name of the author may not be used to endorse or promote products

* derived from this software without specific prior written permission.

*

* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR

* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES

* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.

* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,

* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT

* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,

* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY

* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
 * THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */

```
tree grammar SourceGenTriggers;
options {
language    = Java;
tokenVocab = ANTLRParser;
ASTLabelType = GrammarAST;
}
```

```
@header {
package organtlr.v4.codegen;
import organtlr.v4.misc.Utils;
import organtlr.v4.codegen.model.*;
import organtlr.v4.codegen.model.decl.*;
import organtlr.v4.tool.*;
import organtlr.v4.tool.ast.*;
import java.util.Collections;
import java.util.Map;
import java.util.HashMap;
}
```

```
@members {
public OutputModelController controller;
public boolean hasLookaheadBlock;
public SourceGenTriggers(TreeNodeStream input, OutputModelController controller) {
this(input);
this.controller = controller;
}
}
```

```
dummy : block[null, null] ;
```

```
block[GrammarAST label, GrammarAST ebfnRoot] returns [List<? extends SrcOp> omos]
: ^( blk=BLOCK (^ (OPTIONS .+))?
{List<CodeBlockForAlt> alts = new ArrayList<CodeBlockForAlt>();
( alternative {alts.add($alternative.altCodeBlock);} )+
)
{
if ( alts.size()==1 && ebfnRoot==null) return alts;
if ( ebfnRoot==null ) {
Somos = DefaultOutputModelFactory.list(controller.getChoiceBlock((BlockAST)$blk, alts, $label));
}
else {
Choice choice = controller.getEBNFBlock($ebfnRoot, alts);
hasLookaheadBlock |= choice instanceof PlusBlock || choice instanceof StarBlock;
```

```

        Somos = DefaultOutputModelFactory.list(choice);
    }
}
;

alternative returns [CodeBlockForAlt altCodeBlock, List<SrcOp> ops]
@init {
    boolean outerMost = inContext("RULE BLOCK");
}
@after {
    controller.finishAlternative($altCodeBlock, $ops, outerMost);
}
: a=alt[outerMost] {$altCodeBlock=$a.altCodeBlock; $ops=$a.ops;}
;

alt[boolean outerMost] returns [CodeBlockForAlt altCodeBlock, List<SrcOp> ops]
@init {
    // set alt if outer ALT only (the only ones with alt field set to Alternative object)
    AltAST altAST = (AltAST)retval.start;
    if ( outerMost ) controller.setCurrentOuterMostAlt(altAST.alt);
}
: {
    List<SrcOp> elems = new ArrayList<SrcOp>();
    // TODO: shouldn't we pass $start to controller.alternative()?
    $altCodeBlock = controller.alternative(controller.getCurrentOuterMostAlt(), outerMost);
    $altCodeBlock.ops = $ops = elems;
    controller.setCurrentBlock($altCodeBlock);
}
^( ALT elementOptions? ( element {if ($element.omos!=null) elems.addAll($element.omos);} )+ )

| ^(ALT elementOptions? EPSILON)
    {$altCodeBlock = controller.epsilon(controller.getCurrentOuterMostAlt(), outerMost);}
;

element returns [List<? extends SrcOp> omos]
: labeledElement    {$omos = $labeledElement.omos;}
| atom[null,false] {$omos = $atom.omos;}
| subrule    {$omos = $subrule.omos;}
| ACTION    {$omos = controller.action((ActionAST)$ACTION);}
| SEMPRED    {$omos = controller.sempred((ActionAST)$SEMPRED);}
| ^(ACTION elementOptions) {$omos = controller.action((ActionAST)$ACTION);}
| ^(SEMPRED elementOptions) {$omos = controller.sempred((ActionAST)$SEMPRED);}
;

labeledElement returns [List<? extends SrcOp> omos]
: ^(ASSIGN ID atom[$ID,false] ) {$omos = $atom.omos;}
| ^(PLUS_ASSIGN ID atom[$ID,false]) {$omos = $atom.omos;}
| ^(ASSIGN ID block[$ID,null] ) {$omos = $block.omos;}

```

```

| ^(PLUS_ASSIGN ID block[$ID,null]) {$omos = $block.omos;}
;

subrule returns [List<? extends SrcOp> omos]
: ^(OPTIONAL b=block[null,$OPTIONAL])
{
  $omos = $block.omos;
}
| ( ^(op=CLOSURE b=block[null,null])
| ^(op=POSITIVE_CLOSURE b=block[null,null])
)
{
  List<CodeBlockForAlt> alts = new ArrayList<CodeBlockForAlt>();
  SrcOp blk = $b.omos.get(0);
  CodeBlockForAlt alt = new CodeBlockForAlt(controller.delegate);
  alt.addOp(blk);
  alts.add(alt);
  SrcOp loop = controller.getEBNFBlock($op, alts); // "star it"
  hasLookaheadBlock |= loop instanceof PlusBlock || loop instanceof StarBlock;
  $omos = DefaultOutputModelFactory.list(loop);
}
| block[null, null] {$omos = $block.omos;}
;

blockSet[GrammarAST label, boolean invert] returns [List<SrcOp> omos]
: ^(SET atom[label,invert]+) {$omos = controller.set($SET, $label, invert);}
;

/*
setElement
: STRING_LITERAL
| TOKEN_REF
| ^(RANGE STRING_LITERAL STRING_LITERAL)
;
*/

// TODO: combine ROOT/BANG into one then just make new op ref'ing return value of atom/terminal...
// TODO: same for NOT
atom[GrammarAST label, boolean invert] returns [List<SrcOp> omos]
: ^(NOT a=atom[$label, true]) {$omos = $a.omos;}
| range[label] {$omos = $range.omos;}
| ^(DOT ID terminal[$label])
| ^(DOT ID ruleref[$label])
| ^(WILDCARD .) {$omos = controller.wildcard($WILDCARD, $label);}
| WILDCARD {$omos = controller.wildcard($WILDCARD, $label);}
| terminal[label] {$omos = $terminal.omos;}
| ruleref[label] {$omos = $ruleref.omos;}
| blockSet[$label, invert] {$omos = $blockSet.omos;}

```

```

;

ruleref[GrammarAST label] returns [List<SrcOp> omos]
  : ^(RULE_REF ARG_ACTION? elementOptions?) {$mos = controller.ruleRef($RULE_REF, $label,
$ARG_ACTION);}
  ;

range[GrammarAST label] returns [List<SrcOp> omos]
  : ^(RANGE a=STRING_LITERAL b=STRING_LITERAL)
  ;

terminal[GrammarAST label] returns [List<SrcOp> omos]
  : ^(STRING_LITERAL .) {$mos = controller.stringRef($STRING_LITERAL, $label);}
  | STRING_LITERAL {$mos = controller.stringRef($STRING_LITERAL, $label);}
  | ^(TOKEN_REF ARG_ACTION .) {$mos = controller.tokenRef($TOKEN_REF, $label, $ARG_ACTION);}
  | ^(TOKEN_REF .) {$mos = controller.tokenRef($TOKEN_REF, $label, null);}
  | TOKEN_REF {$mos = controller.tokenRef($TOKEN_REF, $label, null);}
  ;

elementOptions
  : ^(ELEMENT_OPTIONS elementOption+)
  ;

elementOption
  : ID
  | ^(ASSIGN ID ID)
  | ^(ASSIGN ID STRING_LITERAL)
  | ^(ASSIGN ID ACTION)
  | ^(ASSIGN ID INT)
  ;

```

Found in path(s):

```
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/codegen/SourceGenTriggers.g
```

No license file was found, but licenses were detected in source scan.

~ Copyright (c) 2012-2017 The ANTLR Project. All rights reserved.

Found in path(s):

```
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/META-
INF/maven/org/antlr/antlr4/pom.xml
```

No license file was found, but licenses were detected in source scan.

/*

```
* [The "BSD license"]
* Copyright (c) 2012-2016 Terence Parr
* Copyright (c) 2012-2016 Sam Harwell
* Copyright (c) 2014 Eric Vergnaud
```

```

* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products
* derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

```

```

/** ANTLR tool checks output templates are compatible with tool code generation.
* For now, a simple string match used on x.y of x.y.z scheme.
* Must match Tool.VERSION during load to templates.
*
* REQUIRED.
*/

```

```

pythonTypeInitMap ::= [
  "bool":"False",
  "int":"0",
  "float":"0.0",
  "str": "",
  default:"None" // anything other than a primitive type is an object
]

```

```

// args must be <object-model-object>, <fields-resulting-in-STs>

```

```

ParserFile(file, parser, namedActions, contextSuperClass) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
# encoding: utf-8
from __future__ import print_function
from antlr4 import *

```

```

from io import StringIO
import sys

<namedActions.header>
<parser>

>>

ListenerFile(file, header, namedActions) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
from antlr4 import *
<header>

# This class defines a complete listener for a parse tree produced by <file.parserName>.
class <file.grammarName>Listener(ParseTreeListener):

    <file.listenerNames:{lname |
# Enter a parse tree produced by <file.parserName>#<lname>.
def enter<lname; format="cap">(self, ctx):
    pass

# Exit a parse tree produced by <file.parserName>#<lname>.
def exit<lname; format="cap">(self, ctx):
    pass

}; separator="\n">

>>

VisitorFile(file, header, namedActions) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
from antlr4 import *
<header>

# This class defines a complete generic visitor for a parse tree produced by <file.parserName>.

class <file.grammarName>Visitor(ParseTreeVisitor):

    <file.visitorNames:{lname |
# Visit a parse tree produced by <file.parserName>#<lname>.
def visit<lname; format="cap">(self, ctx):
    return self.visitChildren(ctx)

}; separator="\n">

>>

```

```

fileHeader(grammarFileName, ANTLRVersion) ::= <<
# Generated from <grammarFileName> by ANTLR <ANTLRVersion>
>>

Parser(parser, funcs, atn, sempredFuncs, superClass) ::= <<
<Parser_(ctor="parser_ctor", ...)>
>>

Parser_(parser, funcs, atn, sempredFuncs, ctor, superClass) ::= <<
<if(superClass)>
if __name__ is not None and "." in __name__:
    from .<superClass> import <superClass>
else:
    from <superClass> import <superClass>

<endif>
<atn>

class <parser.name> ( <if(superClass)><superClass><else>Parser<endif> ):

    grammarFileName = "<parser.grammarFileName>"

    atn = ATNDeserializer().deserialize(serializedATN())

    decisionsToDFA = [ DFA(ds, i) for i, ds in enumerate(atn.decisionToState) ]

    sharedContextCache = PredictionContextCache()

    literalNames = [ <parser.literalNames:{t | u<t>}; null="u\"<INVALID>\"", separator=", ", wrap, anchor ]

    symbolicNames = [ <parser.symbolicNames:{t | u<t>}; null="u\"<INVALID>\"", separator=", ", wrap, anchor ]

    <parser.rules:{r | RULE_<r.name> = <r.index>}; separator="\n", wrap, anchor>

    ruleNames = [ <parser.ruleNames:{r | u"<r>"}; separator=", ", wrap, anchor ]

    EOF = <TokenLabelType().EOF

    <if(parser.tokens)>
    <parser.tokens:{k | <k>=<parser.tokens.(k)>}; separator="\n", wrap, anchor>
    <endif>

    <parser:(ctor)()>

    <namedActions.members>

    <funcs; separator="\n">

```

```

<if(sempredFuncs)>
  def sempred(self, localctx, ruleIndex, predIndex):
    if self._predicates == None:
      self._predicates = dict()
<parser.sempredFuncs.values: { f |
  self._predicates[<f.ruleIndex>] = self.<f.name>_sempred}; separator="\n
  "
  pred = self._predicates.get(ruleIndex, None)
  if pred is None:
    raise Exception("No predicate with index:" + str(ruleIndex))
  else:
    return pred(localctx, predIndex)

  <sempredFuncs.values; separator="\n">
<endif>

```

```
>>
```

```

dumpActions(recog, argFuncs, actionFuncs, sempredFuncs) ::= <<
<if(actionFuncs)>
def action(self, localctx, ruleIndex, actionIndex):
  if self._actions is None:
    actions = dict()
<recog.actionFuncs.values: { f |
  actions[<f.ruleIndex>] = self.<f.name>_action }; separator="\n">
  self._actions = actions
  action = self._actions.get(ruleIndex, None)
  if action is not None:
    action(localctx, actionIndex)
  else:
    raise Exception("No registered action for:" + str(ruleIndex))

  <actionFuncs.values; separator="\n">

```

```

<endif>
<if(sempredFuncs)>
def sempred(self, localctx, ruleIndex, predIndex):
  if self._predicates is None:
    preds = dict()
<recog.sempredFuncs.values: { f |
  preds[<f.ruleIndex>] = self.<f.name>_sempred}; separator="\n">
  self._predicates = preds
  pred = self._predicates.get(ruleIndex, None)
  if pred is not None:
    return pred(localctx, predIndex)
  else:

```

```

        raise Exception("No registered predicate for:" + str(ruleIndex))

<sempredFuncs.values; separator="\n">
<endif>
>>

parser_ctor(p) ::= <<
def __init__(self, input, output=sys.stdout):
    super(<parser.name>, self).__init__(input, output=output)
    self.checkVersion("<file.ANTLRVersion>")
    self._interp = ParserATNSimulator(self, self.atn, self.decisionsToDFA, self.sharedContextCache)
    self._predicates = None

>>

/* This generates a private method since the actionIndex is generated, making an
* overriding implementation impossible to maintain.
*/
RuleActionFunction(r, actions) ::= <<

def <r.name>_action(self, localctx , actionIndex):
<actions:{index|
<if(first(actions))>
    if actionIndex == <index>:
        <actions.(index)>
<elseif(rest(actions))>
    elif actionIndex == <index>:
        <actions.(index)>
<endif> }; separator="\n">
>>

/* This generates a private method since the predIndex is generated, making an
* overriding implementation impossible to maintain.
*/
RuleSempredFunction(r, actions) ::= <<
def <r.name>_sempred(self, localctx, predIndex):
    <actions:{index|
<if(first(actions))>
    if predIndex == <index>:
        return <actions.(index)>
<elseif(rest(actions))>
    elif predIndex == <index>:
        return <actions.(index)>
<endif> }; separator="\n">

>>

RuleFunction(currentRule,args,code,locals,ruleCtx,altLabelCtxs,namedActions,finallyAction,postamble,exceptions)

```

```

::= <<

<ruleCtx>

<altLabelCtxs:{l | <altLabelCtxs.(l)>}; separator="\n">

def <currentRule.name>(self<currentRule.args:{a | , <a.name>}>):

    localctx = <parser.name>.<currentRule.ctxType>(self, self._ctx, self.state<currentRule.args:{a | , <a.name>}>)
    self.enterRule(localctx, <currentRule.startState>, self.RULE_<currentRule.name>)
    <namedActions.init>
    <locals; separator="\n">
    try:
        <code>
        <postamble; separator="\n">
        <namedActions.after>
    <if(exceptions)>
    <exceptions; separator="\n">
    <else>
    except RecognitionException as re:
        localctx.exception = re
        self._errHandler.reportError(self, re)
        self._errHandler.recover(self, re)
    <endif>
    finally:
        <finallyAction>
        self.exitRule()
    return localctx

>>

```

```

LeftRecursiveRuleFunction(currentRule,args,code,locals,ruleCtx,altLabelCtxs,
    namedActions,finallyAction,postamble) ::=

```

```

<<

```

```

<ruleCtx>
<altLabelCtxs:{l | <altLabelCtxs.(l)>}; separator="\n">

def <currentRule.name>(self, _p=0<if(currentRule.args)>, <args:{a | , <a>}><endif>):
    _parentctx = self._ctx
    _parentState = self.state
    localctx = <parser.name>.<currentRule.ctxType>(self, self._ctx, _parentState<args:{a | , <a.name>}>)
    _prevctx = localctx
    _startState = <currentRule.startState>
    self.enterRecursionRule(localctx, <currentRule.startState>, self.RULE_<currentRule.name>, _p)
    <namedActions.init>
    <locals; separator="\n">
    try:

```

```

    <code>
    <postamble; separator="\n">
    <namedActions.after>
except RecognitionException as re:
    localctx.exception = re
    self._errHandler.reportError(self, re)
    self._errHandler.recover(self, re)
finally:
    <finallyAction>
    self.unrollRecursionContexts(_parentctx)
return localctx

>>

```

```

CodeBlockForOuterMostAlt(currentOuterMostAltCodeBlock, locals, preamble, ops) ::= <<
<if(currentOuterMostAltCodeBlock.altLabel)>localctx = <parser.name>.<currentOuterMostAltCodeBlock.altLabel;
format="cap">Context(self, localctx)<endif>
self.enterOuterAlt(localctx, <currentOuterMostAltCodeBlock.alt.altNum>)
<CodeBlockForAlt(currentAltCodeBlock=currentOuterMostAltCodeBlock, ...)>
>>

```

```

CodeBlockForAlt(currentAltCodeBlock, locals, preamble, ops) ::= <<
<locals; separator="\n">
<preamble; separator="\n">
<ops; separator="\n">
>>

```

```

LL1AltBlock(choice, preamble, alts, error) ::= <<
self.state = <choice.stateNumber>
self._errHandler.sync(self)
<if(choice.label)><labelref(choice.label)> = _input.LT(1)<endif>
<preamble; separator="\n">
token = self._input.LA(1)
<choice.altLook,alts:{look,alt| <cases(ttypes=look)>
    <alt>
    pass}; separator="\nel">
else:
    <error>

>>

```

```

LL1OptionalBlock(choice, alts, error) ::= <<
self.state = <choice.stateNumber>
self._errHandler.sync(self)
token = self._input.LA(1)
<choice.altLook,alts:{look,alt| <cases(ttypes=look)>
    <alt>
    pass}; separator="\nel">

```

else:

pass

>>

LL1OptionalBlockSingleAlt(choice, expr, alts, preamble, error, followExpr) ::= <<

self.state = <choice.stateNumber>

self._errHandler.sync(self)

<preamble; separator="\n">

if <expr>:

 <alts; separator="\n">

<!else if (!(<followExpr>)) <error>!>

>>

LL1StarBlockSingleAlt(choice, loopExpr, alts, preamble, iteration) ::= <<

self.state = <choice.stateNumber>

self._errHandler.sync(self)

<preamble; separator="\n">

while <loopExpr>:

 <alts; separator="\n">

 self.state = <choice.loopBackStateNumber>

 self._errHandler.sync(self)

 <iteration>

>>

LL1PlusBlockSingleAlt(choice, loopExpr, alts, preamble, iteration) ::= <<

self.state = <choice.blockStartStateNumber> <! alt block decision !>

self._errHandler.sync(self)

<preamble; separator="\n">

while True:

 <alts; separator="\n">

 self.state = <choice.stateNumber> <! loopback/exit decision !>

 self._errHandler.sync(self)

 <iteration>

 if not (<loopExpr>):

 break

>>

// LL(*) stuff

AltBlock(choice, preamble, alts, error) ::= <<

self.state = <choice.stateNumber>

self._errHandler.sync(self)

<if(choice.label)><labelref(choice.label)> = _input.LT(1)<endif>

<preamble; separator="\n">

```

la_ = self._interp.adaptivePredict(self._input,<choice.decision>,self._ctx)
<alts:{alt |
if la_ == <i>:
    <alt>
    pass
}; separator="\nел">

>>

```

```

OptionalBlock(choice, alts, error) ::= <<
self.state = <choice.stateNumber>
self._errHandler.sync(self)
la_ = self._interp.adaptivePredict(self._input,<choice.decision>,self._ctx)
<alts:{alt |
if la_ == <i><if(!choice.ast.greedy)>+1<endif>:
    <alt>
}; separator="\nел">

>>

```

```

StarBlock(choice, alts, sync, iteration) ::= <<
self.state = <choice.stateNumber>
self._errHandler.sync(self)
_alt = self._interp.adaptivePredict(self._input,<choice.decision>,self._ctx)
while _alt!=<choice.exitAlt> and _alt!=ATN.INVALID_ALT_NUMBER:
    if _alt==1<if(!choice.ast.greedy)>+1<endif>:
        <iteration>
        <alts> <! should only be one !>
    self.state = <choice.loopBackStateNumber>
    self._errHandler.sync(self)
    _alt = self._interp.adaptivePredict(self._input,<choice.decision>,self._ctx)

>>

```

```

PlusBlock(choice, alts, error) ::= <<
self.state = <choice.blockStartStateNumber> <! alt block decision !>
self._errHandler.sync(self)
_alt = 1<if(!choice.ast.greedy)>+1<endif>
while _alt!=<choice.exitAlt> and _alt!=ATN.INVALID_ALT_NUMBER:
    <alts:{alt |
if _alt == <i><if(!choice.ast.greedy)>+1<endif>:
    <alt>
}; separator="\nел">
    else:
        <error>
    self.state = <choice.loopBackStateNumber> <! loopback/exit decision !>
    self._errHandler.sync(self)
    _alt = self._interp.adaptivePredict(self._input,<choice.decision>,self._ctx)

```

>>

Sync(s) ::= "sync(<s.expecting.name>)"

ThrowNoViableAlt(t) ::= "raise NoViableAltException(self)"

TestSetInline(s) ::= <<

<s.bitsets:{bits | <if(rest(rest(bits.ttypes)))><bitsetBitfieldComparison(s, bits)><else><bitsetInlineComparison(s, bits)><endif>}; separator=" or ">

>>

// Java language spec 15.19 - shift operators mask operands rather than overflow to 0... need range test

testShiftInRange(shiftAmount) ::= <<

((<shiftAmount>) & ~0x3f) == 0

>>

// produces smaller bytecode only when bits.ttypes contains more than two items

bitsetBitfieldComparison(s, bits) ::= <%

(<testShiftInRange({<offsetShiftVar(s.varName, bits.shift)>})> and ((1 << <offsetShiftVar(s.varName, bits.shift)>) & (<bits.ttypes:{ttype | (1 << <offsetShiftType(ttype, bits.shift)>}); separator=" | ">)) != 0
%>

isZero ::= [

"0":true,

default:false

]

offsetShiftVar(shiftAmount, offset) ::= <%

<if(!isZero.(offset))><shiftAmount> - <offset><else><shiftAmount><endif>

%>

offsetShiftType(shiftAmount, offset) ::= <%

<if(!isZero.(offset))><parser.name>.<shiftAmount> - <offset><else><parser.name>.<shiftAmount><endif>

%>

// produces more efficient bytecode when bits.ttypes contains at most two items

bitsetInlineComparison(s, bits) ::= <%

<bits.ttypes:{ttype | <s.varName>==<parser.name>.<ttype>}; separator=" or ">

%>

cases(ttypes) ::= <<

if token in [<ttypes:{t | <parser.name>.<t>}; separator=", ">]:

>>

InvokeRule(r, argExprsChunks) ::= <<

self.state = <r.stateNumber>

<if(r.labels)><r.labels:{l | <labelref(l)> =

```

}><endif>self.<r.name><(if(r.ast.options.p)><r.ast.options.p><if(argExprsChunks)><endif><endif><argExprsChunks>)
>>

MatchToken(m) ::= <<
self.state = <m.stateNumber>
<if(m.labels)><m.labels:{1 | <labelref(l)> = }><endif>self.match(<parser.name>.<m.name>)
>>

MatchSet(m, expr, capture) ::= "<CommonSetStuff(m, expr, capture, false)>"

MatchNotSet(m, expr, capture) ::= "<CommonSetStuff(m, expr, capture, true)>"

CommonSetStuff(m, expr, capture, invert) ::= <<
self.state = <m.stateNumber>
<if(m.labels)><m.labels:{1 | <labelref(l)> = }>self._input.LT(1)<endif>
<capture>
<if(invert)>if <m.varName> \<= 0 or <expr><else>if not(<expr><endif>:
<if(m.labels)><m.labels:{1 | <labelref(l)> = }><else> <endif>self._errHandler.recoverInline(self)
else:
    self._errHandler.reportMatch(self)
    self.consume()
>>

Wildcard(w) ::= <<
self.state = <w.stateNumber>
<if(w.labels)><w.labels:{1 | <labelref(l)> = }><endif>self.matchWildcard()
>>

// ACTION STUFF

Action(a, foo, chunks) ::= "<chunks>"

ArgAction(a, chunks) ::= "<chunks>"

SemPred(p, chunks, failChunks) ::= <<
self.state = <p.stateNumber>
if not <chunks>:
    from antlr4.error.Errors import FailedPredicateException
    raise FailedPredicateException(self, <p.predicate><if(failChunks)>, <failChunks><elseif(p.msg)>,
<p.msg><endif>)
>>

ExceptionClause(e, catchArg, catchAction) ::= <<
catch (<catchArg>) {
    <catchAction>
}
>>

```

```

// lexer actions are not associated with model objects

LexerSkipCommand() ::= "skip()"
LexerMoreCommand() ::= "more()"
LexerPopModeCommand() ::= "popMode()"

LexerTypeCommand(arg, grammar) ::= "_type = <arg>"
LexerChannelCommand(arg, grammar) ::= "_channel = <arg>"
LexerModeCommand(arg, grammar) ::= "_mode = <arg>"
LexerPushModeCommand(arg, grammar) ::= "pushMode(<arg>)"

ActionText(t) ::= "<t.text>"
ActionTemplate(t) ::= "<t.st>"
ArgRef(a) ::= "localctx.<a.name>"
LocalRef(a) ::= "localctx.<a.name>"
RetValRef(a) ::= "localctx.<a.name>"
QRetValRef(a) ::= "<ctx(a)>.<a.dict>.<a.name>"
/** How to translate $tokenLabel */
TokenRef(t) ::= "<ctx(t)>.<t.name>"
LabelRef(t) ::= "<ctx(t)>.<t.name>"
ListLabelRef(t) ::= "<ctx(t)>.<ListLabelName(t.name)>"
SetAttr(s,rhsChunks) ::= "<ctx(s)>.<s.name> = <rhsChunks>"

TokenLabelType() ::= "<file.TokenLabelType; null={Token}>"
InputSymbolType() ::= "<file.InputSymbolType; null={Token}>"

TokenPropertyRef_text(t) ::= "(None if <ctx(t)>.<t.label> is None else <ctx(t)>.<t.label>.text)"
TokenPropertyRef_type(t) ::= "(0 if <ctx(t)>.<t.label> is None else <ctx(t)>.<t.label>.type)"
TokenPropertyRef_line(t) ::= "(0 if <ctx(t)>.<t.label> is None else <ctx(t)>.<t.label>.line)"
TokenPropertyRef_pos(t) ::= "(0 if <ctx(t)>.<t.label> is None else <ctx(t)>.<t.label>.column)"
TokenPropertyRef_channel(t) ::= "(0 if (<ctx(t)>.<t.label> is None else <ctx(t)>.<t.label>.channel)"
TokenPropertyRef_index(t) ::= "(0 if <ctx(t)>.<t.label> is None else <ctx(t)>.<t.label>.tokenIndex)"
TokenPropertyRef_int(t) ::= "(0 if <ctx(t)>.<t.label> is None else int(<ctx(t)>.<t.label>.text))"

RulePropertyRef_start(r) ::= "(None if <ctx(r)>.<r.label> is None else <ctx(r)>.<r.label>.start)"
RulePropertyRef_stop(r) ::= "(None if <ctx(r)>.<r.label> is None else <ctx(r)>.<r.label>.stop)"
RulePropertyRef_text(r) ::= "(None if <ctx(r)>.<r.label> is None else
self._input.getText(<ctx(r)>.<r.label>.start,<ctx(r)>.<r.label>.stop))"
RulePropertyRef_ctx(r) ::= "<ctx(r)>.<r.label>"
RulePropertyRef_parser(r) ::= "self"

ThisRulePropertyRef_start(r) ::= "localctx.start"
ThisRulePropertyRef_stop(r) ::= "localctx.stop"
ThisRulePropertyRef_text(r) ::= "self._input.getText(localctx.start, self._input.LT(-1))"
ThisRulePropertyRef_ctx(r) ::= "localctx"
ThisRulePropertyRef_parser(r) ::= "self"

```

```

NonLocalAttrRef(s) ::= "getInvokingContext(<s.ruleIndex>).<s.name>"
SetNonLocalAttr(s, rhsChunks) ::= "getInvokingContext(<s.ruleIndex>).<s.name> = <rhsChunks>"

AddToLabelList(a) ::= "<ctx(a.label)>.<a.listName>.append(<labelref(a.label)>)"

TokenDecl(t) ::= "self.<t.name> = None # <TokenLabelType()>"
TokenTypeDecl(t) ::= "self.<t.name> = 0 # <TokenLabelType()> type"
TokenListDecl(t) ::= "self.<t.name> = list() # of <TokenLabelType()>s"
RuleContextDecl(r) ::= "self.<r.name> = None # <r.ctxName>"
RuleContextListDecl(rdecl) ::= "self.<rdecl.name> = list() # of <rdecl.ctxName>s"

ContextTokenGetterDecl(t) ::= <<
def <t.name>(self):
    return self.getToken(<parser.name>.<t.name>, 0)
>>

// should never be called
ContextTokenListGetterDecl(t) ::= <<
def <t.name>_list(self):
    return self.getTokens(<parser.name>.<t.name>)
>>

ContextTokenListIndexedGetterDecl(t) ::= <<
def <t.name>(self, i=None):
    if i is None:
        return self.getTokens(<parser.name>.<t.name>)
    else:
        return self.getToken(<parser.name>.<t.name>, i)
>>

ContextRuleGetterDecl(r) ::= <<
def <r.name>(self):
    return self.getTypedRuleContext(<parser.name>.<r.ctxName>,0)

>>

// should never be called
ContextRuleListGetterDecl(r) ::= <<
def <r.name>_list(self):
    return self.getTypedRuleContexts(<parser.name>.<r.ctxName>)

>>

ContextRuleListIndexedGetterDecl(r) ::= <<
def <r.name>(self, i=None):
    if i is None:
        return self.getTypedRuleContexts(<parser.name>.<r.ctxName>)
    else:

```

```

    return self.getTypedRuleContext(<parser.name>.<r.ctxName>,i)

>>

LexerRuleContext() ::= "RuleContext"

/** The rule context name is the rule followed by a suffix; e.g.,
 * r becomes rContext.
 */
RuleContextNameSuffix() ::= "Context"

ImplicitTokenLabel(tokenName) ::= "_<tokenName>"
ImplicitRuleLabel(ruleName) ::= "_<ruleName>"
ImplicitSetLabel(id) ::= "_tset<id>"
ListLabelName(label) ::= "<label>"

CaptureNextToken(d) ::= "<d.varName> = self._input.LT(1)"
CaptureNextTokenType(d) ::= "<d.varName> = self._input.LA(1)"

StructDecl(struct,ctorAttrs,attrs, getters,dispatchMethods,interfaces,extensionMembers) ::= <<
class <struct.name>(<if(contextSuperClass)><contextSuperClass><else>ParserRuleContext<endif>):

    def __init__(self, parser, parent=None, invokingState=-1<struct.ctorAttrs:{a | , <a.name>=None}>>):
        super(<parser.name>.<struct.name>, self).__init__(parent, invokingState)
        self.parser = parser
        <attrs:{a | <a>}; separator="\n">
        <struct.ctorAttrs:{a | self.<a.name> = <a.name>}; separator="\n">

    <getters:{g | <g>}; separator="\n\n">

    def getRuleIndex(self):
        return <parser.name>.RULE_<struct.derivedFromName>

<if(struct.provideCopyFrom)> <! don't need copy unless we have subclasses !>
    def copyFrom(self, ctx):
        super(<parser.name>.<struct.name>, self).copyFrom(ctx)
        <struct.attrs:{a | self.<a.name> = ctx.<a.name>}; separator="\n">

<endif>
    <dispatchMethods; separator="\n">
    <extensionMembers; separator="\n">

>>

AltLabelStructDecl(struct,attrs, getters,dispatchMethods) ::= <<
class <struct.name>(<currentRule.name; format="cap">Context):

    def __init__(self, parser, ctx): # actually a <parser.name>.<currentRule.name; format="cap">Context)

```

```

    super(<parser.name>.<struct.name>, self).__init__(parser)
    <attrs:{ a | <a>}; separator="\n">
    self.copyFrom(ctx)

<getters:{ g | <g>}; separator="\n">

<dispatchMethods; separator="\n">

>>

ListenerDispatchMethod(method) ::= <<
def <if(method.isEnter)>enter<else>exit<endif>Rule(self, listener):
    if hasattr(listener, "<if(method.isEnter)>enter<else>exit<endif><struct.derivedFromName; format="cap">"):
        listener.<if(method.isEnter)>enter<else>exit<endif><struct.derivedFromName; format="cap">(self)

>>

VisitorDispatchMethod(method) ::= <<
def accept(self, visitor):
    if hasattr(visitor, "visit<struct.derivedFromName; format="cap">"):
        return visitor.visit<struct.derivedFromName; format="cap">(self)
    else:
        return visitor.visitChildren(self)

>>

AttributeDecl(d) ::= "self.<d.name> = <if(d.initValue)><d.initValue><else>None<endif>"

/** If we don't know location of label def x, use this template */
labelref(x) ::= "<if(!x.isLocal)>localctx.<endif><x.name>"

/** For any action chunk, what is correctly-typed context struct ptr? */
ctx(actionChunk) ::= "localctx"

// used for left-recursive rules
recRuleAltPredicate(ruleName,opPrec) ::= "self.precpred(self._ctx, <opPrec>)"
recRuleSetReturnAction(src,name) ::= "$<name>=<src>.<name>"
recRuleSetStopToken() ::= "self._ctx.stop = self._input.LT(-1)"

recRuleAltStartAction(ruleName, ctxName, label) ::= <<
localctx = <parser.name>.<ctxName>Context(self, _parentctx, _parentState)
<if(label)>localctx.<label> = _prevctx<endif>
self.pushNewRecursionContext(localctx, _startState, self.RULE_<ruleName>)
>>

recRuleLabeledAltStartAction(ruleName, currentAltLabel, label, isListLabel) ::= <<
localctx = <parser.name>.<currentAltLabel; format="cap">Context(self, <parser.name>.<ruleName;
format="cap">Context(self, _parentctx, _parentState))

```

```

<if(label)>
<if(isListLabel)>
localctx.<label>.append(_prevctx)
<else>
localctx.<label> = _prevctx
<endif>
<endif>
self.pushNewRecursionContext(localctx, _startState, self.RULE_<ruleName>)
>>

```

```

recRuleReplaceContext(ctxName) ::= <<
localctx = <parser.name>.<ctxName>Context(self, localctx)
self._ctx = localctx
_prevctx = localctx
>>

```

```

recRuleSetPrevCtx() ::= <<
if self._parseListeners is not None:
    self.triggerExitRuleEvent()
_prevctx = localctx
>>

```

```

LexerFile(lexerFile, lexer, namedActions) ::= <<
<fileHeader(lexerFile.grammarFileName, lexerFile.ANTLRVersion)>
# encoding: utf-8
from __future__ import print_function
from antlr4 import *
from io import StringIO
import sys

```

```

<namedActions.header>

```

```

<lexer>
>>

```

```

Lexer(lexer, atn, actionFuncs, sempredFuncs, superClass) ::= <<
<if(superClass)>
if __name__ is not None and "." in __name__:
    from .<superClass> import <superClass>
else:
    from <superClass> import <superClass>

```

```

<endif>

```

```

<atn>

```

```

class <lexer.name>(<if(superClass)><superClass><else>Lexer<endif>):

    atn = ATNDeserializer().deserialize(serializedATN())

    decisionsToDFA = [ DFA(ds, i) for i, ds in enumerate(atn.decisionToState) ]

<if(lexer.channels)>
    <lexer.channels:{c| <c> = <lexer.channels.(c)>}; separator="\n">

<endif>
<if(rest(lexer.modes)>
    <rest(lexer.modes):{m| <m> = <i>}; separator="\n">

<endif>
<lexer.tokens:{k | <k> = <lexer.tokens.(k)>}; separator="\n", wrap, anchor>

    channelNames = [ u"DEFAULT_TOKEN_CHANNEL", u"HIDDEN"<if (lexer.channels)>, <lexer.channels:{c|
u"<c>"}; separator=", ", wrap, anchor><endif> ]

    modeNames = [ <lexer.modes:{m| u"<m>"}; separator=", ", wrap, anchor> ]

    literalNames = [ u"\<INVALID>",
        <lexer.literalNames:{t | u<t>}; separator=", ", wrap, anchor> ]

    symbolicNames = [ u"\<INVALID>",
        <lexer.symbolicNames:{t | u<t>}; separator=", ", wrap, anchor> ]

    ruleNames = [ <lexer.ruleNames:{r | u"<r>"}; separator=", ", wrap, anchor> ]

    grammarFileName = u"<lexer.grammarFileName>"

    def __init__(self, input=None, output=sys.stdout):
        super(<lexer.name>, self).__init__(input, output=output)
        self.checkVersion("<lexerFile.ANTLRVersion>")
        self._interp = LexerATNSimulator(self, self.atn, self.decisionsToDFA, PredictionContextCache())
        self._actions = None
        self._predicates = None

<namedActions.members>

<dumpActions(lexer, "", actionFuncs, sempredFuncs)>

>>

SerializedATN(model) ::= <<
<! only one segment, can be inlined !>

def serializedATN():

```

```
with StringIO() as buf:
    buf.write(u"<model.serialized; wrap={ }<\n>    buf.write(u"}>")
    return buf.getvalue()
```

```
>>
```

```
/** Using a type to init value map, try to init a type; if not in table
 * must be an object, default value is "null".
 */
initValue(typeName) ::= <<
<pythonTypeInitMap.(typeName)>
>>
```

```
codeFileExtension() ::= ".py"
```

Found in path(s):

```
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/tool/templates/codegen/Python2/Python2.stg
```

No license file was found, but licenses were detected in source scan.

```
/*
 * [The "BSD license"]
 * Copyright (c) 2012-2016 Terence Parr
 * Copyright (c) 2012-2016 Sam Harwell
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * 3. The name of the author may not be used to endorse or promote products
 * derived from this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
 * IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
 * IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
 * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
 * DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
 * THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
 * THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

```

*/

/** The definitive ANTLR v3 tree grammar to walk/visit ANTLR v4 grammars.
* Parses trees created by ANTLRParser.g.
*
* Rather than have multiple tree grammars, one for each visit, I'm
* creating this generic visitor that knows about context. All of the
* boilerplate pattern recognition is done here. Then, subclasses can
* override the methods they care about. This prevents a lot of the same
* context tracking stuff like "set current alternative for current
* rule node" that is repeated in lots of tree filters.
*/

```

```

tree grammar GrammarTreeVisitor;
options {
language    = Java;
tokenVocab = ANTLRParser;
ASTLabelType = GrammarAST;
}

```

```

// Include the copyright in this source and also the generated source

```

```

@header {

```

```

/*

```

```

[The "BSD license"]

```

```

Copyright (c) 2011 Terence Parr

```

```

All rights reserved.

```

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```

*/
package org.antlr.v4.parse;
import org.antlr.v4.Tool;
import org.antlr.v4.tool.*;
import org.antlr.v4.tool.ast.*;
import java.lang.reflect.InvocationTargetException;
import java.lang.reflect.Method;
}

@members {
public String grammarName;
public GrammarAST currentRuleAST;
public String currentModeName = LexerGrammar.DEFAULT_MODE_NAME;
public String currentRuleName;
public GrammarAST currentOuterAltRoot;
public int currentOuterAltNumber = 1; // 1..n
public int rewriteEBNFLevel = 0;

public GrammarTreeVisitor() { this(null); }

// Should be abstract but can't make gen'd parser abstract;
// subclasses should implement else everything goes to stderr!
public ErrorManager getErrorManager() { return null; }

public void visitGrammar(GrammarAST t) { visit(t, "grammarSpec"); }
public void visit(GrammarAST t, String ruleName) {
    CommonTreeNodeStream nodes = new CommonTreeNodeStream(new GrammarASTAdaptor(), t);
    setTreeNodeStream(nodes);
    try {
        Method m = getClass().getMethod(ruleName);
        m.invoke(this);
    }
    catch (Throwable e) {
        ErrorManager errMgr = getErrorManager();
        if ( e instanceof InvocationTargetException ) {
            e = e.getCause();
        }
        //e.printStackTrace(System.err);
        if ( errMgr==null ) {
            System.err.println("can't find rule "+ruleName+
                " or tree structure error: "+t.toStringTree()
            );
            e.printStackTrace(System.err);
        }
        else errMgr.toolError(ErrorType.INTERNAL_ERROR, e);
    }
}
}

```

```

public void discoverGrammar(GrammarRootAST root, GrammarAST ID) { }
public void finishPrequels(GrammarAST firstPrequel) { }
public void finishGrammar(GrammarRootAST root, GrammarAST ID) { }

public void grammarOption(GrammarAST ID, GrammarAST valueAST) { }
public void ruleOption(GrammarAST ID, GrammarAST valueAST) { }
public void blockOption(GrammarAST ID, GrammarAST valueAST) { }
public void defineToken(GrammarAST ID) { }
public void defineChannel(GrammarAST ID) { }
public void globalNamedAction(GrammarAST scope, GrammarAST ID, ActionAST action) { }
public void importGrammar(GrammarAST label, GrammarAST ID) { }

public void modeDef(GrammarAST m, GrammarAST ID) { }

public void discoverRules(GrammarAST rules) { }
public void finishRules(GrammarAST rule) { }
public void discoverRule(RuleAST rule, GrammarAST ID, List<GrammarAST> modifiers,
    ActionAST arg, ActionAST returns, GrammarAST thrws,
    GrammarAST options, ActionAST locals,
    List<GrammarAST> actions,
    GrammarAST block) { }
public void finishRule(RuleAST rule, GrammarAST ID, GrammarAST block) { }
public void discoverLexerRule(RuleAST rule, GrammarAST ID, List<GrammarAST> modifiers,
    GrammarAST block) { }
public void finishLexerRule(RuleAST rule, GrammarAST ID, GrammarAST block) { }
public void ruleCatch(GrammarAST arg, ActionAST action) { }
public void finallyAction(ActionAST action) { }
public void discoverOuterAlt(AltAST alt) { }
public void finishOuterAlt(AltAST alt) { }
public void discoverAlt(AltAST alt) { }
public void finishAlt(AltAST alt) { }

public void ruleRef(GrammarAST ref, ActionAST arg) { }
public void tokenRef(TerminalAST ref) { }
public void elementOption(GrammarASTWithOptions t, GrammarAST ID, GrammarAST valueAST) { }
public void stringRef(TerminalAST ref) { }
public void wildcardRef(GrammarAST ref) { }
public void actionInAlt(ActionAST action) { }
public void sempredInAlt(PredAST pred) { }
public void label(GrammarAST op, GrammarAST ID, GrammarAST element) { }
public void lexerCallCommand(int outerAltNumber, GrammarAST ID, GrammarAST arg) { }
public void lexerCommand(int outerAltNumber, GrammarAST ID) { }

protected void enterGrammarSpec(GrammarAST tree) { }
protected void exitGrammarSpec(GrammarAST tree) { }

protected void enterPrequelConstructs(GrammarAST tree) { }
protected void exitPrequelConstructs(GrammarAST tree) { }

```

protected void enterPrequelConstruct(GrammarAST tree) { }

protected void exitPrequelConstruct(GrammarAST tree) { }

protected void enterOptionsSpec(GrammarAST tree) { }

protected void exitOptionsSpec(GrammarAST tree) { }

protected void enterOption(GrammarAST tree) { }

protected void exitOption(GrammarAST tree) { }

protected void enterOptionValue(GrammarAST tree) { }

protected void exitOptionValue(GrammarAST tree) { }

protected void enterDelegateGrammars(GrammarAST tree) { }

protected void exitDelegateGrammars(GrammarAST tree) { }

protected void enterDelegateGrammar(GrammarAST tree) { }

protected void exitDelegateGrammar(GrammarAST tree) { }

protected void enterTokensSpec(GrammarAST tree) { }

protected void exitTokensSpec(GrammarAST tree) { }

protected void enterTokenSpec(GrammarAST tree) { }

protected void exitTokenSpec(GrammarAST tree) { }

protected void enterChannelsSpec(GrammarAST tree) { }

protected void exitChannelsSpec(GrammarAST tree) { }

protected void enterChannelSpec(GrammarAST tree) { }

protected void exitChannelSpec(GrammarAST tree) { }

protected void enterAction(GrammarAST tree) { }

protected void exitAction(GrammarAST tree) { }

protected void enterRules(GrammarAST tree) { }

protected void exitRules(GrammarAST tree) { }

protected void enterMode(GrammarAST tree) { }

protected void exitMode(GrammarAST tree) { }

protected void enterLexerRule(GrammarAST tree) { }

protected void exitLexerRule(GrammarAST tree) { }

protected void enterRule(GrammarAST tree) { }

protected void exitRule(GrammarAST tree) { }

protected void enterExceptionGroup(GrammarAST tree) { }

protected void exitExceptionGroup(GrammarAST tree) { }

protected void enterExceptionHandler(GrammarAST tree) { }

protected void exitExceptionHandler(GrammarAST tree) { }

protected void enterFinallyClause(GrammarAST tree) { }

protected void exitFinallyClause(GrammarAST tree) { }

protected void enterLocals(GrammarAST tree) { }

protected void exitLocals(GrammarAST tree) { }

protected void enterRuleReturns(GrammarAST tree) { }

protected void exitRuleReturns(GrammarAST tree) { }

protected void enterThrowsSpec(GrammarAST tree) { }

protected void exitThrowsSpec(GrammarAST tree) { }

protected void enterRuleAction(GrammarAST tree) { }

protected void exitRuleAction(GrammarAST tree) { }

protected void enterRuleModifier(GrammarAST tree) { }

protected void exitRuleModifier(GrammarAST tree) { }

protected void enterLexerRuleBlock(GrammarAST tree) { }

protected void exitLexerRuleBlock(GrammarAST tree) { }

protected void enterRuleBlock(GrammarAST tree) { }

protected void exitRuleBlock(GrammarAST tree) { }

protected void enterLexerOuterAlternative(AltAST tree) { }

protected void exitLexerOuterAlternative(AltAST tree) { }

protected void enterOuterAlternative(AltAST tree) { }

protected void exitOuterAlternative(AltAST tree) { }

protected void enterLexerAlternative(GrammarAST tree) { }

protected void exitLexerAlternative(GrammarAST tree) { }

protected void enterLexerElements(GrammarAST tree) { }

protected void exitLexerElements(GrammarAST tree) { }

protected void enterLexerElement(GrammarAST tree) { }

protected void exitLexerElement(GrammarAST tree) { }

protected void enterLabeledLexerElement(GrammarAST tree) { }

protected void exitLabeledLexerElement(GrammarAST tree) { }

protected void enterLexerBlock(GrammarAST tree) { }

protected void exitLexerBlock(GrammarAST tree) { }

protected void enterLexerAtom(GrammarAST tree) { }
protected void exitLexerAtom(GrammarAST tree) { }

protected void enterActionElement(GrammarAST tree) { }
protected void exitActionElement(GrammarAST tree) { }

protected void enterAlternative(AltAST tree) { }
protected void exitAlternative(AltAST tree) { }

protected void enterLexerCommand(GrammarAST tree) { }
protected void exitLexerCommand(GrammarAST tree) { }

protected void enterLexerCommandExpr(GrammarAST tree) { }
protected void exitLexerCommandExpr(GrammarAST tree) { }

protected void enterElement(GrammarAST tree) { }
protected void exitElement(GrammarAST tree) { }

protected void enterAstOperand(GrammarAST tree) { }
protected void exitAstOperand(GrammarAST tree) { }

protected void enterLabeledElement(GrammarAST tree) { }
protected void exitLabeledElement(GrammarAST tree) { }

protected void enterSubrule(GrammarAST tree) { }
protected void exitSubrule(GrammarAST tree) { }

protected void enterLexerSubrule(GrammarAST tree) { }
protected void exitLexerSubrule(GrammarAST tree) { }

protected void enterBlockSuffix(GrammarAST tree) { }
protected void exitBlockSuffix(GrammarAST tree) { }

protected void enterEbnfSuffix(GrammarAST tree) { }
protected void exitEbnfSuffix(GrammarAST tree) { }

protected void enterAtom(GrammarAST tree) { }
protected void exitAtom(GrammarAST tree) { }

protected void enterBlockSet(GrammarAST tree) { }
protected void exitBlockSet(GrammarAST tree) { }

protected void enterSetElement(GrammarAST tree) { }
protected void exitSetElement(GrammarAST tree) { }

protected void enterBlock(GrammarAST tree) { }
protected void exitBlock(GrammarAST tree) { }

```

protected void enterRuleref(GrammarAST tree) { }
protected void exitRuleref(GrammarAST tree) { }

protected void enterRange(GrammarAST tree) { }
protected void exitRange(GrammarAST tree) { }

protected void enterTerminal(GrammarAST tree) { }
protected void exitTerminal(GrammarAST tree) { }

protected void enterElementOptions(GrammarAST tree) { }
protected void exitElementOptions(GrammarAST tree) { }

protected void enterElementOption(GrammarAST tree) { }
protected void exitElementOption(GrammarAST tree) { }

@Override
public void traceIn(String ruleName, int ruleIndex) {
    System.err.println("enter "+ruleName+": "+input.LT(1));
}

@Override
public void traceOut(String ruleName, int ruleIndex) {
    System.err.println("exit "+ruleName+": "+input.LT(1));
}

grammarSpec
@init {
    enterGrammarSpec($start);
}
@after {
    exitGrammarSpec($start);
}
: ^( GRAMMAR ID {grammarName=$ID.text;}
    {discoverGrammar((GrammarRootAST)$GRAMMAR, $ID);}
    prequelConstructs
    {finishPrequels($prequelConstructs.firstOne);}
    rules mode*
    {finishGrammar((GrammarRootAST)$GRAMMAR, $ID);}
)
;

prequelConstructs returns [GrammarAST firstOne=null]
@init {
    enterPrequelConstructs($start);
}
@after {

```

```

exitPrequelConstructs($start);
}
: {$firstOne=$start;} prequelConstruct+
|
;

```

```

prequelConstruct
@init {
enterPrequelConstructs($start);
}
@after {
exitPrequelConstructs($start);
}
: optionsSpec
| delegateGrammars
| tokensSpec
| channelsSpec
| action
;

```

```

optionsSpec
@init {
enterOptionsSpec($start);
}
@after {
exitOptionsSpec($start);
}
: ^(OPTIONS option*)
;

```

```

option
@init {
enterOption($start);
boolean rule = inContext("RULE ...");
boolean block = inContext("BLOCK ...");
}
@after {
exitOption($start);
}
: ^(a=ASSIGN ID v=optionValue)
{
if ( block ) blockOption($ID, $v.start); // most specific first
else if ( rule ) ruleOption($ID, $v.start);
else grammarOption($ID, $v.start);
}
;

```

optionValue returns [String v]

```

@init {
  enterOptionValue($start);
  $v = $start.token.getText();
}
@after {
  exitOptionValue($start);
}
: ID
| STRING_LITERAL
| INT
;

```

delegateGrammars

```

@init {
  enterDelegateGrammars($start);
}
@after {
  exitDelegateGrammars($start);
}
: ^(IMPORT delegateGrammar+)
;

```

delegateGrammar

```

@init {
  enterDelegateGrammar($start);
}
@after {
  exitDelegateGrammar($start);
}
: ^(ASSIGN label=ID id=ID) {importGrammar($label, $id);}
| id=ID {importGrammar(null, $id);}
;

```

tokensSpec

```

@init {
  enterTokensSpec($start);
}
@after {
  exitTokensSpec($start);
}
: ^(TOKENS_SPEC tokenSpec+)
;

```

tokenSpec

```

@init {
  enterTokenSpec($start);
}
@after {

```

```

exitTokenSpec($start);
}
: ID {defineToken($ID);}
;

channelsSpec
@init {
enterChannelsSpec($start);
}
@after {
exitChannelsSpec($start);
}
: ^(CHANNELS channelSpec+)
;

channelSpec
@init {
enterChannelSpec($start);
}
@after {
exitChannelSpec($start);
}
: ID {defineChannel($ID);}
;

action
@init {
enterAction($start);
}
@after {
exitAction($start);
}
: ^(AT sc=ID? name=ID ACTION) {globalNamedAction($sc, $name, (ActionAST)$ACTION);}
;

rules
@init {
enterRules($start);
}
@after {
exitRules($start);
}
: ^(RULES {discoverRules($RULES);} (rule|lexerRule)* {finishRules($RULES);})
;

mode
@init {
enterMode($start);

```

```

}
@after {
    exitMode($start);
}
: ^( MODE ID {currentModeName=$ID.text; modeDef($MODE, $ID);} lexerRule* )
;

lexerRule
@init {
    enterLexerRule($start);
    List<GrammarAST> mods = new ArrayList<GrammarAST>();
    currentOuterAltNumber=0;
}
@after {
    exitLexerRule($start);
}
: ^( RULE TOKEN_REF
    {currentRuleName=$TOKEN_REF.text; currentRuleAST=$RULE;}
    (^ (RULEMODIFIERS m=FRAGMENT {mods.add($m);}))?
    {discoverLexerRule((RuleAST)$RULE, $TOKEN_REF, mods, (GrammarAST)input.LT(1));}
    lexerRuleBlock
    {
        finishLexerRule((RuleAST)$RULE, $TOKEN_REF, $lexerRuleBlock.start);
        currentRuleName=null; currentRuleAST=null;
    }
    )
;

rule
@init {
    enterRule($start);
    List<GrammarAST> mods = new ArrayList<GrammarAST>();
    List<GrammarAST> actions = new ArrayList<GrammarAST>(); // track roots
    currentOuterAltNumber=0;
}
@after {
    exitRule($start);
}
: ^( RULE RULE_REF {currentRuleName=$RULE_REF.text; currentRuleAST=$RULE;}
    (^ (RULEMODIFIERS (m=ruleModifier {mods.add($m.start);}))+)?
    ARG_ACTION?
    ret=ruleReturns?
    thr=throwsSpec?
    loc=locals?
    ( opts=optionsSpec
    | a=ruleAction {actions.add($a.start);}
    )*
    {discoverRule((RuleAST)$RULE, $RULE_REF, mods, (ActionAST)$ARG_ACTION,

```

```

    $ret.start!=null?(ActionAST)$ret.start.getChild(0):null,
    $thr.start, $opts.start,
    $loc.start!=null?(ActionAST)$loc.start.getChild(0):null,
    actions, (GrammarAST)input.LT(1));}
ruleBlock exceptionGroup
{finishRule((RuleAST)$RULE, $RULE_REF, $ruleBlock.start); currentRuleName=null; currentRuleAST=null;}
)
;

exceptionGroup
@init {
    enterExceptionGroup($start);
}
@after {
    exitExceptionGroup($start);
}
: exceptionHandler* finallyClause?
;

exceptionHandler
@init {
    enterExceptionHandler($start);
}
@after {
    exitExceptionHandler($start);
}
: ^(CATCH ARG_ACTION ACTION) {ruleCatch($ARG_ACTION, (ActionAST)$ACTION);}
;

finallyClause
@init {
    enterFinallyClause($start);
}
@after {
    exitFinallyClause($start);
}
: ^(FINALLY ACTION) {finallyAction((ActionAST)$ACTION);}
;

locals
@init {
    enterLocals($start);
}
@after {
    exitLocals($start);
}
: ^(LOCALS ARG_ACTION)
;

```

```
ruleReturns
@init {
    enterRuleReturns($start);
}
@after {
    exitRuleReturns($start);
}
: ^(RETURNS ARG_ACTION)
;
```

```
throwsSpec
@init {
    enterThrowsSpec($start);
}
@after {
    exitThrowsSpec($start);
}
: ^(THROWS ID+)
;
```

```
ruleAction
@init {
    enterRuleAction($start);
}
@after {
    exitRuleAction($start);
}
: ^(AT ID ACTION)
;
```

```
ruleModifier
@init {
    enterRuleModifier($start);
}
@after {
    exitRuleModifier($start);
}
: PUBLIC
| PRIVATE
| PROTECTED
| FRAGMENT
;
```

```
lexerRuleBlock
@init {
    enterLexerRuleBlock($start);
}
}
```

```

@after {
  exitLexerRuleBlock($start);
}
: ^( BLOCK
  ( {
    currentOuterAltRoot = (GrammarAST)input.LT(1);
    currentOuterAltNumber++;
  }
  lexerOuterAlternative
  )+
  )
;

```

```

ruleBlock
@init {
  enterRuleBlock($start);
}
@after {
  exitRuleBlock($start);
}
: ^( BLOCK
  ( {
    currentOuterAltRoot = (GrammarAST)input.LT(1);
    currentOuterAltNumber++;
  }
  outerAlternative
  )+
  )
;

```

```

lexerOuterAlternative
@init {
  enterLexerOuterAlternative((AltAST)$start);
  discoverOuterAlt((AltAST)$start);
}
@after {
  finishOuterAlt((AltAST)$start);
  exitLexerOuterAlternative((AltAST)$start);
}
: lexerAlternative
;

```

```

outerAlternative
@init {
  enterOuterAlternative((AltAST)$start);
  discoverOuterAlt((AltAST)$start);
}

```

```

@after {
  finishOuterAlt((AltAST)$start);
  exitOuterAlternative((AltAST)$start);
}
: alternative
;

```

lexerAlternative

```

@init {
  enterLexerAlternative($start);
}
@after {
  exitLexerAlternative($start);
}
: ^(LEXER_ALT_ACTION lexerElements lexerCommand+)
  | lexerElements
;

```

lexerElements

```

@init {
  enterLexerElements($start);
}
@after {
  exitLexerElements($start);
}
: ^(ALT lexerElement+)
;

```

lexerElement

```

@init {
  enterLexerElement($start);
}
@after {
  exitLexerElement($start);
}
: labeledLexerElement
  | lexerAtom
  | lexerSubrule
  | ACTION    {actionInAlt((ActionAST)$ACTION);}
  | SEMPRED   {sempredInAlt((PredAST)$SEMPRED);}
  | ^(ACTION elementOptions) {actionInAlt((ActionAST)$ACTION);}
  | ^(SEMPRED elementOptions) {sempredInAlt((PredAST)$SEMPRED);}
  | EPSILON
;

```

labeledLexerElement

```

@init {
  enterLabeledLexerElement($start);
}

```

```
}
@after {
  exitLabeledLexerElement($start);
}
: ^((ASSIGN|PLUS_ASSIGN) ID (lexerAtom|block))
;
```

```
lexerBlock
@init {
  enterLexerBlock($start);
}
@after {
  exitLexerBlock($start);
}
: ^(BLOCK optionsSpec? lexerAlternative+)
;
```

```
lexerAtom
@init {
  enterLexerAtom($start);
}
@after {
  exitLexerAtom($start);
}
: terminal
| ^(NOT blockSet)
| blockSet
| ^(WILDCARD elementOptions)
| WILDCARD
| LEXER_CHAR_SET
| range
| ruleref
;
```

```
actionElement
@init {
  enterActionElement($start);
}
@after {
  exitActionElement($start);
}
: ACTION
| ^(ACTION elementOptions)
| SEMPRED
| ^(SEMPRED elementOptions)
;
```

```
alternative
```

```

@init {
  enterAlternative((AltAST)$start);
  discoverAlt((AltAST)$start);
}
@after {
  finishAlt((AltAST)$start);
  exitAlternative((AltAST)$start);
}
: ^(ALT elementOptions? element+)
| ^(ALT elementOptions? EPSILON)
;

lexerCommand
@init {
  enterLexerCommand($start);
}
@after {
  exitLexerCommand($start);
}
: ^(LEXER_ACTION_CALL ID lexerCommandExpr)
  {lexerCallCommand(currentOuterAltNumber, $ID, $lexerCommandExpr.start);}
| ID
  {lexerCommand(currentOuterAltNumber, $ID);}
;

lexerCommandExpr
@init {
  enterLexerCommandExpr($start);
}
@after {
  exitLexerCommandExpr($start);
}
: ID
| INT
;

element
@init {
  enterElement($start);
}
@after {
  exitElement($start);
}
: labeledElement
| atom
| subrule
| ACTION    {actionInAlt((ActionAST)$ACTION);}
| SEMPRED   {sempredInAlt((PredAST)$SEMPRED);}

```

```

| ^(ACTION elementOptions) {actionInAlt((ActionAST)$ACTION);}
| ^(SEMPRED elementOptions) {sempredInAlt((PredAST)$SEMPRED);}
| range
| ^(NOT blockSet)
| ^(NOT block)
;

astOperand
@init {
enterAstOperand($start);
}
@after {
exitAstOperand($start);
}
: atom
| ^(NOT blockSet)
| ^(NOT block)
;

labeledElement
@init {
enterLabeledElement($start);
}
@after {
exitLabeledElement($start);
}
: ^((ASSIGN|PLUS_ASSIGN) ID element) {label($start, $ID, $element.start);}
;

subrule
@init {
enterSubrule($start);
}
@after {
exitSubrule($start);
}
: ^(blockSuffix block)
| block
;

lexerSubrule
@init {
enterLexerSubrule($start);
}
@after {
exitLexerSubrule($start);
}
: ^(blockSuffix lexerBlock)

```

| lexerBlock

;

blockSuffix

@init {

enterBlockSuffix(\$start);

}

@after {

exitBlockSuffix(\$start);

}

: ebnfSuffix

;

ebnfSuffix

@init {

enterEbnfSuffix(\$start);

}

@after {

exitEbnfSuffix(\$start);

}

: OPTIONAL

| CLOSURE

| POSITIVE_CLOSURE

;

atom

@init {

enterAtom(\$start);

}

@after {

exitAtom(\$start);

}

: ^(DOT ID terminal)

| ^(DOT ID ruleref)

| ^(WILDCARD elementOptions) {wildcardRef(\$WILDCARD);}

| WILDCARD {wildcardRef(\$WILDCARD);}

| terminal

| blockSet

| ruleref

;

blockSet

@init {

enterBlockSet(\$start);

}

@after {

exitBlockSet(\$start);

}

```

: ^(SET setElement+)
;

setElement
@init {
  enterSetElement($start);
}
@after {
  exitSetElement($start);
}
: ^(STRING_LITERAL elementOptions) {stringRef((TerminalAST)$STRING_LITERAL);}
| ^(TOKEN_REF elementOptions) {tokenRef((TerminalAST)$TOKEN_REF);}
| STRING_LITERAL {stringRef((TerminalAST)$STRING_LITERAL);}
| TOKEN_REF {tokenRef((TerminalAST)$TOKEN_REF);}
| ^(RANGE a=STRING_LITERAL b=STRING_LITERAL)
{
  stringRef((TerminalAST)$a);
  stringRef((TerminalAST)$b);
}
| LEXER_CHAR_SET
;

```

```

block
@init {
  enterBlock($start);
}
@after {
  exitBlock($start);
}
: ^(BLOCK optionsSpec? ruleAction* ACTION? alternative+)
;

```

```

ruleref
@init {
  enterRuleref($start);
}
@after {
  exitRuleref($start);
}
: ^(RULE_REF arg=ARG_ACTION? elementOptions?)
{
  ruleRef($RULE_REF, (ActionAST)$ARG_ACTION);
  if ( $arg!=null ) actionInAlt((ActionAST)$arg);
}
;

```

```

range
@init {

```

```

enterRange($start);
}
@after {
exitRange($start);
}
: ^(RANGE STRING_LITERAL STRING_LITERAL)
;

terminal
@init {
enterTerminal($start);
}
@after {
exitTerminal($start);
}
: ^(STRING_LITERAL elementOptions)
  {stringRef((TerminalAST)$STRING_LITERAL);}
| STRING_LITERAL {stringRef((TerminalAST)$STRING_LITERAL);}
| ^(TOKEN_REF elementOptions) {tokenRef((TerminalAST)$TOKEN_REF);}
| TOKEN_REF {tokenRef((TerminalAST)$TOKEN_REF);}
;

elementOptions
@init {
enterElementOptions($start);
}
@after {
exitElementOptions($start);
}
: ^(ELEMENT_OPTIONS elementOption[(GrammarASTWithOptions)$start.getParent()]*)
;

elementOption[GrammarASTWithOptions t]
@init {
enterElementOption($start);
}
@after {
exitElementOption($start);
}
: ID {elementOption(t, $ID, null);}
| ^(ASSIGN id=ID v=ID) {elementOption(t, $id, $v);}
| ^(ASSIGN ID v=STRING_LITERAL) {elementOption(t, $ID, $v);}
| ^(ASSIGN ID v=ACTION) {elementOption(t, $ID, $v);}
| ^(ASSIGN ID v=INT) {elementOption(t, $ID, $v);}
;

```

Found in path(s):

* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-

jar/org/antlr/v4/parse/GrammarTreeVisitor.g

No license file was found, but licenses were detected in source scan.

```
/*
 * [The "BSD license"]
 * Copyright (c) 2012-2016 Terence Parr
 * Copyright (c) 2012-2016 Sam Harwell
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * 3. The name of the author may not be used to endorse or promote products
 * derived from this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
 * IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
 * IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
 * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
 * DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
 * THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
 * THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */
```

```
/** How to generate rules derived from left-recursive rules.
```

```
 * These rely on recRuleAltPredicate(),
 * recRuleArg(), recRuleSetResultAction(), recRuleSetReturnAction()
 * templates in main language.stg
 */
```

```
group LeftRecursiveRules;
```

```
recRule(ruleName, argName, primaryAlts, opAlts, setResultAction,
        userRetvals, leftRecursiveRuleRefLabels) ::=
```

```
<<
```

```
<ruleName><if(userRetvals)> returns [<userRetvals>]<endif>
```

```
 : ( { } <primaryAlts:{alt | <alt.altText> }; separator="\n    | ">
```

```
 )
```

```
 (
```

```
   <opAlts; separator="\n    | ">
```

```

    )*
    ;
>>

recRuleAlt(alt, precOption, opPrec, pred) ::= <<
{<pred>}?{\<precOption>=<opPrec>}\> <alt.altText>
>>

```

Found in path(s):

```

* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/tool/templates/LeftRecursiveRules.stg

```

No license file was found, but licenses were detected in source scan.

```

/*
* [The "BSD license"]
* Copyright (c) 2012-2016 Terence Parr
* Copyright (c) 2012-2016 Sam Harwell
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products
* derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

// File   : A3Lexer.g
// Author  : Jim Idle (jimi@temporal-wave.com)
// Copyright : Free BSD - See @header clause below
// Version  : First implemented as part of ANTLR 3.2 this is the self
//           hosting ANTLR 3 Lexer.

```

```

//
// Description
// -----
// This is the definitive lexer grammar for parsing ANTLR V3.x.x grammars. All other
// grammars are derived from this grammar via source code control integration (perforce)
// or by the gdiff tool.
//
// This grammar and its associated grmmars A3Parser.g and A3Walker.g exhibit the following
// traits, which are recommended for all production quality grammars:
//
// 1) They are separate grammars, not composite grammars;
// 2) They implement all supporting methods in a superclass (at least this is recommended
//    for language targets that support inheritance;
// 3) All errors are pushed as far down the parsing chain as possible, which means
//    that the lexer tries to defer error reporting to the parser, and the parser
//    tries to defer error reporting to a semantic phase consisting of a single
//    walk of the AST. The reason for this is that the error messages produced
//    from later phases of the parse will generally have better context and so
//    be more useful to the end user. Consider the message: "Syntax error at 'options'"
//    vs: "You cannot specify two options{ } sections in a single grammar file".
// 4) The lexer is 'programmed' to catch common mistakes such as unterminated literals
//    and report them specifically and not just issue confusing lexer mismatch errors.
//
/** Read in an ANTLR grammar and build an AST. Try not to do
 * any actions, just build the tree.
 *
 * The phases are:
 *
 * A3Lexer.g (this file)
 *     A3Parser.g
 *     A3Verify.g (derived from A3Walker.g)
 * assign.types.g
 * define.g
 * buildnfa.g
 * antlr.print.g (optional)
 * codegen.g
 *
 * Terence Parr
 * University of San Francisco
 * 2005
 * Jim Idle (this v3 grammar)
 * Temporal Wave LLC
 * 2009
 */
lexer grammar ANTLRLexer;

// =====

```

```

// Note that while this grammar does not care about order of constructs
// that don't really matter, such as options before @header etc, it must first
// be parsed by the original v2 parser, before it replaces it. That parser does
// care about order of structures. Hence we are constrained by the v2 parser
// for at least the first bootstrap release that causes this parser to replace
// the v2 version.
// =====

// -----
// Options
//
// V3 option directives to tell the tool what we are asking of it for this
// grammar.
//
options {

// Target language is Java, which is the default but being specific
// here as this grammar is also meant as a good example grammar for
// for users.
//
language    = Java;

// The super class that this lexer should expect to inherit from, and
// which contains any and all support routines for the lexer. This is
// commented out in this baseline (definitive or normative grammar)
// - see the ANTLR tool implementation for hints on how to use the super
// class
//
//superclass = AbstractA3Lexer;
}

tokens { SEMPREP; TOKEN_REF; RULE_REF; LEXER_CHAR_SET; ARG_ACTION; }

// Include the copyright in this source and also the generated source
//
@lexer::header {
/*
[The "BSD licence"]
Copyright (c) 2005-2009 Terence Parr
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the

```

documentation and/or other materials provided with the distribution.

3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```
*/
```

```
package org.antlr.v4.parse;  
import org.antlr.v4.tool.*;  
import org.antlr.v4.runtime.misc.Interval;  
}
```

```
@members {
```

```
public static final int COMMENTS_CHANNEL = 2;
```

```
public CommonTokenStream tokens; // track stream we push to; need for context info  
public boolean isLexerRule = false;
```

```
public void grammarError(ErrorType etype, org.antlr.runtime.Token token, Object... args) { }
```

```
/** scan backwards from current point in this.tokens list
```

```
 * looking for the start of the rule or subrule.
```

```
 * Return token or null if for some reason we can't find the start.
```

```
*/
```

```
public Token getRuleOrSubruleStartToken() {
```

```
    if ( tokens==null ) return null;
```

```
    int i = tokens.index();
```

```
    int n = tokens.size();
```

```
    if ( i>=n ) i = n-1; // seems index == n as we lex
```

```
    while ( i>=0 && i<n ) {
```

```
        int ttype = tokens.get(i).getType();
```

```
        if ( ttype == LPAREN || ttype == TOKEN_REF || ttype == RULE_REF ) {
```

```
            return tokens.get(i);
```

```
        }
```

```
        i--;
```

```
    }
```

```
    return null;
```

```
}
```

```
}
```

```

// -----
// Comments
//
// ANTLR comments can be multi or single line and we don't care
// which particularly. However we also accept Javadoc style comments
// of the form: /** ... */ and we do take care to distinguish those
// from ordinary multi-line comments
// Note how we guide the lexical PATH because we want to issue a decriptive
// error message in case of a standalone '/' character, which makes no
// sense in ANTLR source code. We also trap unterminated multi-line comments
//
fragment DOC_COMMENT : ;
COMMENT
@init {

// Record the start line and offsets as if we need to report an
// unterminated comment, then we want to show the start of the comment
// we think is broken, not the end, where people will have to try and work
// it out themselves.
//
int startLine = $line;
int offset = getCharPositionInLine();
}
: // Eat the first character only, then see if we have a comment
// or something silly.
//
'/' // Comment introducer

(
// Single line comment, possibly with embedded src/line directives
// in a similar style to the C pre-processor, allowing generated
// code to refer the programmer back to the original source code
// in case of error.
//
'/'
(
(' $ANTLR')=> ' $ANTLR' SRC
| ~(NLCHARS)*
)

| // Multi-line comment, which may be a documentation comment
// if it starts /** (note that we protect against accidentally
// recognizing a comment /**/ as a documentation comment
//
'/**' (
{ input.LA(2) != '/' }?=> '*' { $type = DOC_COMMENT; }
| { true }?=> // Required to cover all alts with predicates

```

```

)

// Should we support embedded multiline comments here?
//
(
    // Pick out end of multiline comment and exit the loop
    // if we find it.
    //
    { !(input.LA(1) == '*' && input.LA(2) == '/') }?

    // Anything else other than the non-greedy match of
    // the comment close sequence
    //
    .
)*
(
    // Look for the comment terminator, but if it is accidentally
    // unterminated, then we will hit EOF, which will trigger the
    // epsilon alt and hence we can issue an error message relative
    // to the start of the unterminated multi-line comment
    //
    /*'

| // Unterminated comment!
//
{
    // ErrorManager.msg(Msg.UNTERMINATED_DOC_COMMENT, startLine, offset, $pos, startLine,
offset, $pos, (Object)null);
}
)

| // There was nothing that made sense following the opening '/' and so
// we issue an error regarding the malformed comment
//
{
    // TODO: Insert error message relative to comment start
    //
}
)
{
    // We do not wish to pass the comments in to the parser. If you are
    // writing a formatter then you will want to preserve the comments off
    // channel, but could just skip and save token space if not.
    //
    $channel=COMMENTS_CHANNEL;
}
;

```

```

ARG_OR_CHARSET
options {k=1;}
: {isLexerRule}?=> LEXER_CHAR_SET {$type=LEXER_CHAR_SET;}
| {!isLexerRule}?=> ARG_ACTION
{
$type=ARG_ACTION;
// Set the token text to our gathered string minus outer [ ]
String t = $text;
t = t.substring(1,t.length()-1);
setText(t);
}
;

```

fragment

LEXER_CHAR_SET

```

: '['
( '\\ ~('\r'\n')
| ~('\r'\n'\\'']')
)*
']'
;

```

// -----

// Argument specs

//

// Certain argument lists, such as those specifying call parameters

// to a rule invocation, or input parameters to a rule specification

// are contained within square brackets. In the lexer we consume them

// all at once and sort them out later in the grammar analysis.

//

fragment

ARG_ACTION

```

: '['
(
ARG_ACTION

| ("")=>ACTION_STRING_LITERAL

| ("\")=>ACTION_CHAR_LITERAL

| ~('[']')
)*

']'
;

```

// -----

// Actions

```

//
// Other than making sure to distinguish between { and } embedded
// within what we have assumed to be literals in the action code, the
// job of the lexer is merely to gather the code within the action
// (delimited by {}) and pass it to the parser as a single token.
// We know that this token will be asked for its text somewhere
// in the upcoming parse, so setting the text here to exclude
// the delimiting {} is no additional overhead.
//
ACTION
: NESTED_ACTION
('?' {$type = SEMPRED;}
 ( (WSNLCHARS* '=>') => WSNLCHARS* '=>' // v3 gated sempred
  {
    Token t = new CommonToken(input, state.type, state.channel, state.tokenStartCharIndex, getCharIndex()-1);
    t.setLine(state.tokenStartLine);
    t.setText(state.text);
    t.setCharPositionInLine(state.tokenStartCharPositionInLine);
    grammarError(ErrorType.V3_GATED_SEMPRED, t);
  }
 )?
 )?
 ;

// -----
// Action structure
//
// Many language targets use {} as block delimiters and so we
// must recursively match {} delimited blocks to balance the
// braces. Additionally, we must make some assumptions about
// literal string representation in the target language. We assume
// that they are delimited by ' or " and so consume these
// in their own alts so as not to inadvertently match {}.
// This rule calls itself on matching a {
//
fragment
NESTED_ACTION
@init {

// Record the start line and offsets as if we need to report an
// unterminated block, then we want to show the start of the comment
// we think is broken, not the end, where people will have to try and work
// it out themselves.
//
int startLine = getLine();
int offset = getCharPositionInLine();
}

```

```

: // Action and other blocks start with opening {
//
'{'
(
// And now we can match one of a number of embedded
// elements within the action until we find a
// } that balances the opening {. If we do not find
// the balanced } then we will hit EOF and can issue
// an error message about the brace that we believe to
// be mismatched. This won't be foolproof but we will
// be able to at least report an error against the
// opening brace that we feel is in error and this will
// guide the user to the correction as best we can.
//

// An embedded {} block
//
NESTED_ACTION

| // What appears to be a literal
//
ACTION_CHAR_LITERAL

| // We have assumed that the target language has C/Java
// type comments.
//
COMMENT

| // What appears to be a literal
//
ACTION_STRING_LITERAL

| // What appears to be an escape sequence
//
ACTION_ESC

| // Some other single character that is not
// handled above
//
~(\\|'|\"|/|'{'|'})

)*

(
// Correctly balanced closing brace
//
'}'

```

```

|// Looks like have an imbalanced {} block, report
|// with respect to the opening brace.
|//
|{
|    // TODO: Report imbalanced {}
|    System.out.println("Block starting at line " + startLine + " offset " + (offset+1) + " contains imbalanced {} or
is missing a }");
|}
|)
|;

```

```

// Keywords
// -----
// keywords used to specify ANTLR v3 grammars. Keywords may not be used as
// labels for rules or in any other context where they would be ambiguous
// with the keyword vs some other identifier
// OPTIONS, TOKENS, and CHANNELS must also consume the opening brace that captures
// their option block, as this is the easiest way to parse it separate
// to an ACTION block, despite it using the same {} delimiters.
//

```

```

OPTIONS    : 'options' WSNLCHARS* '{' ;
TOKENS_SPEC : 'tokens' WSNLCHARS* '{' ;
CHANNELS   : 'channels' WSNLCHARS* '{' ;

```

```

IMPORT     : 'import'          ;
FRAGMENT   : 'fragment'       ;
LEXER      : 'lexer'          ;
PARSER     : 'parser'         ;
GRAMMAR    : 'grammar'        ;
TREE_GRAMMAR : 'tree' WSNLCHARS* 'grammar' ;
PROTECTED  : 'protected'      ;
PUBLIC     : 'public'         ;
PRIVATE    : 'private'        ;
RETURNS    : 'returns'        ;
LOCALS     : 'locals'         ;
THROWS     : 'throws'         ;
CATCH      : 'catch'          ;
FINALLY    : 'finally'        ;
MODE       : 'mode'           ;

```

```

// -----
// Punctuation
//
// Character sequences used as separators, delimiters, operators, etc
//
COLON      : ':'

```

```

    {
    // scan backwards, looking for a RULE_REF or TOKEN_REF.
    // which would indicate the start of a rule definition.
    // If we see a LPAREN, then it's the start of the subrule.
    // this.tokens is the token string we are pushing into, so
    // just loop backwards looking for a rule definition. Then
    // we set isLexerRule.
    Token t = getRuleOrSubruleStartToken();
    if ( t!=null ) {
        if ( t.getType()==RULE_REF ) isLexerRule = false;
        else if ( t.getType()==TOKEN_REF ) isLexerRule = true;
        // else must be subrule; don't alter context
    }
    }
;
COLONCOLON  : '::'      ;
COMMA       : ','      ;
SEMI        : ';'      ;
LPAREN      : '('      ;
RPAREN      : ')'      ;
RARROW      : '->'    ;
LT          : '<'      ;
GT          : '>'      ;
ASSIGN      : '='      ;
QUESTION    : '?'      ;
SYNPRED     : '=>'    ;
    {
        Token t = new CommonToken(input, state.type, state.channel,
            state.tokenStartCharIndex, getCharIndex()-1);
        t.setLine(state.tokenStartLine);
        t.setText(state.text);
        t.setCharPositionInLine(state.tokenStartCharPositionInLine);
        grammarError(ErrorType.V3_SYNPRED, t);
        $channel=HIDDEN;
    }
;
STAR        : '*'      ;
PLUS        : '+'      ;
PLUS_ASSIGN : '+='     ;
OR          : '|'      ;
DOLLAR      : '$'      ;
DOT         : '.'      ; // can be WILDCARD or DOT in qid or imported rule ref
RANGE      : '..'      ;
AT          : '@'      ;
POUND      : '#'      ;
NOT        : '~'      ;
RBRACE     : '}'      ;

```

```

/** Allow unicode rule/token names */
ID : a=NameStartChar NameChar*
{
  if ( Grammar.isTokenName($a.text) ) $type = TOKEN_REF;
  else $type = RULE_REF;
}
;

fragment
NameChar : NameStartChar
  | '0'..'9'
  | '_'
  | '\u00B7'
  | '\u0300'..\u036F'
  | '\u203F'..\u2040'
  ;

fragment
NameStartChar
  : 'A'..'Z' | 'a'..'z'
  | '\u00C0'..\u00D6'
  | '\u00D8'..\u00F6'
  | '\u00F8'..\u02FF'
  | '\u0370'..\u037D'
  | '\u037F'..\u1FFF'
  | '\u200C'..\u200D'
  | '\u2070'..\u218F'
  | '\u2C00'..\u2FEF'
  | '\u3001'..\uD7FF'
  | '\uF900'..\uFDCF'
  | '\uFDF0'..\uFEFE'
  | '\uFF00'..\uFFFD'
  ; // ignores | [\u10000-\uEFFFF] ;

// -----
// Literals embedded in actions
//
// Note that we have made the assumption that the language used within
// actions uses the fairly standard " and ' delimiters for literals and
// that within these literals, characters are escaped using the \ character.
// There are some languages which do not conform to this in all cases, such
// as by using /string/ and so on. We will have to deal with such cases if
// if they come up in targets.
//
// Within actions, or other structures that are not part of the ANTLR
// syntax, we may encounter literal characters. Within these, we do
// not want to inadvertently match things like '}' and so we eat them

```

```

// specifically. While this rule is called CHAR it allows for the fact that
// some languages may use/allow ' as the string delimiter.
//
fragment
ACTION_CHAR_LITERAL
: "\"" ((\\)=>ACTION_ESC | ~\" )* "\""
;

// Within actions, or other structures that are not part of the ANTLR
// syntax, we may encounter literal strings. Within these, we do
// not want to inadvertently match things like '}' and so we eat them
// specifically.
//
fragment
ACTION_STRING_LITERAL
: "\"" ((\\)=>ACTION_ESC | ~\" )* "\""
;

// Within literal strings and characters that are not part of the ANTLR
// syntax, we must allow for escaped character sequences so that we do not
// inadvertently recognize the end of a string or character when the terminating
// delimiter has been escaped.
//
fragment
ACTION_ESC
: '\\' .
;

// -----
// Integer
//
// Obviously (I hope) match an arbitrary long sequence of digits.
//
INT : ('0'..'9')+
;

// -----
// Source spec
//
// A fragment rule for picking up information about an originating
// file from which the grammar we are parsing has been generated. This allows
// ANTLR to report errors against the originating file and not the generated
// file.
//
fragment
SRC : 'src' WSCHARS+ file=ACTION_STRING_LITERAL WSCHARS+ line=INT
{
    // TODO: Add target specific code to change the source file name and current line number

```

```

    //
    }
;

// -----
// Literal string
//
// ANTLR makes no distinction between a single character literal and a
// multi-character string. All literals are single quote delimited and
// may contain unicode escape sequences of the form \uxxxx or \u{xxxxxx},
// where x is a valid hexadecimal number.
STRING_LITERAL
: "\"" ( ( ESC_SEQ | ~("\\"|\\"|\r|\n' ) ) ) *
  ( "\""
  | // Unterminated string literal
    {
      Token t = new CommonToken(input, state.type, state.channel, state.tokenStartCharIndex, getCharIndex()-1);
      t.setLine(state.tokenStartLine);
      t.setText(state.text);
      t.setCharPositionInLine(state.tokenStartCharPositionInLine);
      grammarError(ErrorType.UNTERMINATED_STRING_LITERAL, t);
    }
  )
;

// A valid hex digit specification
//
fragment
HEX_DIGIT : ('0'..'9'|'a'..'f'|'A'..'F') ;

// Any kind of escaped character that we can embed within ANTLR
// literal strings.
//
fragment
ESC_SEQ
: "\"
  (
    // The standard escaped character set such as tab, newline, etc...
    'b'|'t'|'n'|'f'|'r'|'\"'|'\'

    | // A Java style Unicode escape sequence
    UNICODE_ESC

    | // A Swift/Hack style Unicode escape sequence
    UNICODE_EXTENDED_ESC

    | // An illegal escape sequence
    ~(b|t|n|f|r|\"|'|\\"|'u) // \x for any invalid x (make sure to match char here)
  )
;

```

```

    {
        Token t = new CommonToken(input, state.type, state.channel, getCharIndex()-2, getCharIndex()-1);
        t.setText(t.getText());
        t.setLine(input.getLine());
        t.setCharPositionInLine(input.getCharPositionInLine()-2);
        grammarError(ErrorType.INVALID_ESCAPE_SEQUENCE, t, input.substring(getCharIndex()-
2,getCharIndex()-1));
    }
)
;

```

fragment

UNICODE_ESC

@init {

// Flag to tell us whether we have a valid number of
// hex digits in the escape sequence

//

int hCount = 0;

}

: 'u' // Leadin for unicode escape sequence

// We now require 4 hex digits. Note though
// that we accept any number of characters
// and issue an error if we do not get 4. We cannot
// use an infinite count such as + because this
// might consume too many, so we lay out the lexical
// options and issue an error at the invalid paths.

//

(

(

HEX_DIGIT { hCount++; }

(

HEX_DIGIT { hCount++; }

(

HEX_DIGIT { hCount++; }

(

// Four valid hex digits, we are good

//

HEX_DIGIT { hCount++; }

|// Three valid digits

)

|// Two valid digits

)

|// One valid digit

```

    )
    )
    | // No valid hex digits at all
)

// Now check the digit count and issue an error if we need to
//
{
    if (hCount < 4) {
        Interval badRange = Interval.of(getCharIndex()-2-hCount, getCharIndex());
        String lastChar = input.substring(badRange.b, badRange.b);
        if ( lastChar.codePointAt(0)=="\") {
            badRange.b--;
        }
        String bad = input.substring(badRange.a, badRange.b);
        Token t = new CommonToken(input, state.type, state.channel, badRange.a, badRange.b);
        t.setLine(input.getLine());
        t.setCharacterPositionInLine(input.getCharacterPositionInLine()-hCount-2);
        grammarError(ErrorType.INVALID_ESCAPE_SEQUENCE, t, bad);
    }
}
;

fragment
UNICODE_EXTENDED_ESC
: 'u{' // Leadin for unicode extended escape sequence

    HEX_DIGIT+ // One or more hexadecimal digits

    '}' // Leadout for unicode extended escape sequence

// Now check the digit count and issue an error if we need to
{
    int numDigits = getCharIndex()-state.tokenStartCharIndex-6;
    if (numDigits > 6) {
        Token t = new CommonToken(input, state.type, state.channel, state.tokenStartCharIndex, getCharIndex()-
1);
        t.setText(t.getText());
        t.setLine(input.getLine());
        t.setCharacterPositionInLine(input.getCharacterPositionInLine()-numDigits);
        grammarError(ErrorType.INVALID_ESCAPE_SEQUENCE, t,
input.substring(state.tokenStartCharIndex,getCharIndex()-1));
    }
}
;

// -----
// Whitespace

```

```

//
// Characters and character constructs that are of no import
// to the parser and are used to make the grammar easier to read
// for humans.
//
WS
: (
  ''
  | '\t'
  | '\r'
  | '\n'
  | '\f'
)+
  {$channel=HIDDEN;}
;

// A fragment rule for use in recognizing end of line in
// rules like COMMENT.
//
fragment
NLCHARS
: '\n' | '\r'
;

// A fragment rule for recognizing traditional whitespace
// characters within lexer rules.
//
fragment
WSCHARS
: ' ' | '\t' | '\f'
;

// A fragment rule for recognizing both traditional whitespace and
// end of line markers, when we don't care to distinguish but don't
// want any action code going on.
//
fragment
WSNLCHARS
: ' ' | '\t' | '\f' | '\n' | '\r'
;

// This rule allows ANTLR 4 to parse grammars using the UTF-8 encoding with a
// byte order mark. Since this Unicode character doesn't appear as a token
// anywhere else in the grammar, we can simply skip all instances of it without
// problem. This rule will not break usage of \uFEFF inside a LEXER_CHAR_SET or
// STRING_LITERAL.
UnicodeBOM
: '\uFEFF' {skip();}

```

```

;

// -----
// Illegal Character
//
// This is an illegal character trap which is always the last rule in the
// lexer specification. It matches a single character of any value and being
// the last rule in the file will match when no other rule knows what to do
// about the character. It is reported as an error but is not passed on to the
// parser. This means that the parser to deal with the gramamr file anyway
// but we will not try to analyse or code generate from a file with lexical
// errors.
//
ERRCHAR
: .
{
    Token t = new CommonToken(input, state.type, state.channel, state.tokenStartCharIndex, getCharIndex()-1);
    t.setLine(state.tokenStartLine);
    t.setText(state.text);
    t.setCharPositionInLine(state.tokenStartCharPositionInLine);
    String msg = getTokenErrorDisplay(t) + " came as a complete surprise to me";
    grammarError(ErrorType.SYNTAX_ERROR, t, msg);
    state.syntaxErrors++;
    skip();
}
;

```

Found in path(s):

* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/parse/ANTLRLexer.g

No license file was found, but licenses were detected in source scan.

/*

[The "BSD licence"]

Copyright (c) 2005-20012 Terence Parr

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR

IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

Found in path(s):

* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/parse/ANTLRParser.java

No license file was found, but licenses were detected in source scan.

/*

* [The "BSD license"]

* Copyright (c) 2012-2016 Terence Parr

* Copyright (c) 2012-2016 Sam Harwell

* All rights reserved.

*

* Redistribution and use in source and binary forms, with or without

* modification, are permitted provided that the following conditions

* are met:

*

* 1. Redistributions of source code must retain the above copyright

* notice, this list of conditions and the following disclaimer.

* 2. Redistributions in binary form must reproduce the above copyright

* notice, this list of conditions and the following disclaimer in the

* documentation and/or other materials provided with the distribution.

* 3. The name of the author may not be used to endorse or promote products

* derived from this software without specific prior written permission.

*

* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR

* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES

* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.

* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,

* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT

* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,

* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY

* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT

* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF

* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

javaTypeInitMap ::= [

"int": "0",

"long": "0",

```

"float": "0.0f",
"double": "0.0",
"boolean": "false",
"byte": "0",
"short": "0",
"char": "0",
default: "null" // anything other than a primitive type is an object
]

```

```
// args must be <object-model-object>, <fields-resulting-in-STs>
```

```

ParserFile(file, parser, namedActions, contextSuperClass) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
<if(file.genPackage)>
package <file.genPackage>;
<endif>
<namedActions.header>
import org.antlr.v4.runtime.atn.*;
import org.antlr.v4.runtime.dfa.DFA;
import org.antlr.v4.runtime.*;
import org.antlr.v4.runtime.misc.*;
import org.antlr.v4.runtime.tree.*;
import java.util.List;
import java.util.Iterator;
import java.util.ArrayList;

<parser>
>>

```

```

ListenerFile(file, header, namedActions) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
<if(file.genPackage)>
package <file.genPackage>;
<endif>
<header>
import org.antlr.v4.runtime.tree.ParseTreeListener;

/**
 * This interface defines a complete listener for a parse tree produced by
 * { @link <file.parserName> }.
 */
public interface <file.grammarName>Listener extends ParseTreeListener {
    <file.listenerNames> {Iname |
/**
<if(file.listenerLabelRuleNames.(Iname))>
 * Enter a parse tree produced by the { @code <Iname>}
 * labeled alternative in { @link <file.parserName>#<file.listenerLabelRuleNames.(Iname)> }.
<else>

```

```

* Enter a parse tree produced by { @link <file.parserName>#<lname>\}.
<endif>
* @param ctx the parse tree
*/
void enter<lname; format="cap">(<file.parserName>.<lname; format="cap">Context ctx);
/**
<if(file.listenerLabelRuleNames.(lname))>
* Exit a parse tree produced by the { @code <lname>\}
* labeled alternative in { @link <file.parserName>#<file.listenerLabelRuleNames.(lname)>\}.
<else>
* Exit a parse tree produced by { @link <file.parserName>#<lname>\}.
<endif>
* @param ctx the parse tree
*/
void exit<lname; format="cap">(<file.parserName>.<lname; format="cap">Context ctx);; separator="\n">
}
>>

```

```

BaseListenerFile(file, header, namedActions) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
<if(file.genPackage)>
package <file.genPackage>;
<endif>
<header>

```

```

import org.antlr.v4.runtime.ParserRuleContext;
import org.antlr.v4.runtime.tree.ErrorNode;
import org.antlr.v4.runtime.tree.TerminalNode;

```

```

/**
* This class provides an empty implementation of { @link <file.grammarName>Listener},
* which can be extended to create a listener which only needs to handle a subset
* of the available methods.
*/
public class <file.grammarName>BaseListener implements <file.grammarName>Listener {
<file.listenerNames: {lname |
/**
* { @inheritDoc\}
*
* \<p>The default implementation does nothing.\</p>
*/
@Override public void enter<lname; format="cap">(<file.parserName>.<lname; format="cap">Context ctx) { \}
/**
* { @inheritDoc\}
*
* \<p>The default implementation does nothing.\</p>
*/
@Override public void exit<lname; format="cap">(<file.parserName>.<lname; format="cap">Context ctx) { \} };

```

```

separator="\n">

/**
 * {@inheritDoc}
 *
 * \<p>The default implementation does nothing.\</p>
 */
@Override public void enterEveryRule(ParserRuleContext ctx) { }
/**
 * {@inheritDoc}
 *
 * \<p>The default implementation does nothing.\</p>
 */
@Override public void exitEveryRule(ParserRuleContext ctx) { }
/**
 * {@inheritDoc}
 *
 * \<p>The default implementation does nothing.\</p>
 */
@Override public void visitTerminal(TerminalNode node) { }
/**
 * {@inheritDoc}
 *
 * \<p>The default implementation does nothing.\</p>
 */
@Override public void visitErrorNode(ErrorNode node) { }
}
>>

VisitorFile(file, header, namedActions) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
<if(file.genPackage)>
package <file.genPackage>;
<endif>
<header>
import org.antlr.v4.runtime.tree.ParseTreeVisitor;

/**
 * This interface defines a complete generic visitor for a parse tree produced
 * by { @link <file.parserName> }.
 *
 * @param \<T> The return type of the visit operation. Use { @link Void } for
 * operations with no return type.
 */
public interface <file.grammarName>Visitor\<T> extends ParseTreeVisitor\<T> {
    <file.visitorNames>: { Iname |
/**
<if(file.visitorLabelRuleNames.(Iname))>

```

```

* Visit a parse tree produced by the { @code <lname>\}
* labeled alternative in { @link <file.parserName>#<file.visitorLabelRuleNames.(lname)>\}.
<else>
* Visit a parse tree produced by { @link <file.parserName>#<lname>\}.
<endif>
* @param ctx the parse tree
* @return the visitor result
*/
T visit<lname; format="cap">(<file.parserName>.<lname; format="cap">Context ctx);}; separator="\n">
}
>>

```

```

BaseVisitorFile(file, header, namedActions) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
<if(file.genPackage)>
package <file.genPackage>;
<endif>
<header>
import org.antlr.v4.runtime.tree.AbstractParseTreeVisitor;

/**
 * This class provides an empty implementation of { @link <file.grammarName>Visitor},
 * which can be extended to create a visitor which only needs to handle a subset
 * of the available methods.
 *
 * @param \<T> The return type of the visit operation. Use { @link Void} for
 * operations with no return type.
 */
public class <file.grammarName>BaseVisitor\<T> extends AbstractParseTreeVisitor\<T> implements
<file.grammarName>Visitor\<T> {
    <file.visitorNames:{lname |
/**
 * { @inheritDoc\}
 *
 * \<p>The default implementation returns the result of calling
 * { @link #visitChildren\} on { @code ctx\}.\</p>
 */
@Override public T visit<lname; format="cap">(<file.parserName>.<lname; format="cap">Context ctx) { return
visitChildren(ctx); \} }; separator="\n">
}
>>

```

```

fileHeader(grammarFileName, ANTLRVersion) ::= <<
// Generated from <grammarFileName; format="java-escape"> by ANTLR <ANTLRVersion>
>>

```

```

Parser(parser, funcs, atn, sempredFuncs, superClass) ::= <<
<Parser_(ctor="parser_ctor", ...)>

```

>>

```
Parser_(parser, funcs, atn, sempredFuncs, ctor, superClass) ::= <<
@SuppressWarnings({"all", "warnings", "unchecked", "unused", "cast"})
public class <parser.name> extends <superClass; null="Parser"> {
    static { RuntimeMetaData.checkVersion("<file.ANTLRVersion>", RuntimeMetaData.VERSION); }

    protected static final DFA[] _decisionToDFA;
    protected static final PredictionContextCache _sharedContextCache =
        new PredictionContextCache();
    <if(parser.tokens)>
    public static final int
        <parser.tokens:{k | <k>=<parser.tokens.(k)>}; separator=", ", wrap, anchor>;
    <endif>
    public static final int
        <parser.rules:{r | RULE_<r.name> = <r.index>}; separator=", ", wrap, anchor>;
    private static String[] makeRuleNames() {
        return new String[] {
            <parser.ruleNames:{r | "<r>"}; separator=", ", wrap, anchor>
        };
    }
    public static final String[] ruleNames = makeRuleNames();

    <vocabulary(parser.literalNames, parser.symbolicNames)>

    @Override
    public String getGrammarFileName() { return "<parser.grammarFileName; format="java-escape">"; }

    @Override
    public String[] getRuleNames() { return ruleNames; }

    @Override
    public String getSerializedATN() { return _serializedATN; }

    @Override
    public ATN getATN() { return _ATN; }

    <namedActions.members>
    <parser:(ctor)()>
    <funcs; separator="\n">

    <if(sempredFuncs)>
    public boolean sempred(RuleContext _localctx, int ruleIndex, int predIndex) {
        switch (ruleIndex) {
            <parser.sempredFuncs.values:{f}
        case <f.ruleIndex>:
            return <f.name>_sempred((<f.ctxType>)_localctx, predIndex);}; separator="\n">
        }
    }
}
```

```

    return true;
}
<sempredFuncs.values; separator="\n">
<endif>

<atn>
}
>>

vocabulary(literalNames, symbolicNames) ::= <<
private static String[] makeLiteralNames() {
return new String[] {
<literalNames:{t | <t>}; null="null", separator=", ", wrap, anchor>
};
}
private static final String[] _LITERAL_NAMES = makeLiteralNames();
private static String[] makeSymbolicNames() {
return new String[] {
<symbolicNames:{t | <t>}; null="null", separator=", ", wrap, anchor>
};
}
private static final String[] _SYMBOLIC_NAMES = makeSymbolicNames();
public static final Vocabulary VOCABULARY = new VocabularyImpl(_LITERAL_NAMES,
_SYMBOLIC_NAMES);

/**
 * @deprecated Use {@link #VOCABULARY} instead.
 */
@Deprecated
public static final String[] tokenNames;
static {
tokenNames = new String[_SYMBOLIC_NAMES.length];
for (int i = 0; i < tokenNames.length; i++) {
tokenNames[i] = VOCABULARY.getLiteralName(i);
if (tokenNames[i] == null) {
tokenNames[i] = VOCABULARY.getSymbolicName(i);
}

if (tokenNames[i] == null) {
tokenNames[i] = "\<INVALID>";
}
}
}

@Override
@Deprecated
public String[] getTokenNames() {
return tokenNames;
}

```

```

}

@Override

public Vocabulary getVocabulary() {
    return VOCABULARY;
}
>>

dumpActions(recog, argFuncs, actionFuncs, sempredFuncs) ::= <<
<if(actionFuncs)>
@Override
public void action(RuleContext _localctx, int ruleIndex, int actionIndex) {
    switch (ruleIndex) {
        <recog.actionFuncs.values:{f}
    case <f.ruleIndex>:
        <f.name>_action((<f.ctxType>)_localctx, actionIndex);
        break;}; separator="\n">
    }
}
<actionFuncs.values; separator="\n">
<endif>
<if(sempredFuncs)>
@Override
public boolean sempred(RuleContext _localctx, int ruleIndex, int predIndex) {
    switch (ruleIndex) {
        <recog.sempredFuncs.values:{f}
    case <f.ruleIndex>:
        return <f.name>_sempred((<f.ctxType>)_localctx, predIndex);}; separator="\n">
    }
    return true;
}
<sempredFuncs.values; separator="\n">
<endif>
>>

parser_ctor(p) ::= <<
public <p.name>(TokenStream input) {
    super(input);
    _interp = new ParserATNSimulator(this,_ATN,_decisionToDFA,_sharedContextCache);
}
>>

/* This generates a private method since the actionIndex is generated, making an
* overriding implementation impossible to maintain.
*/
RuleActionFunction(r, actions) ::= <<
private void <r.name>_action(<r.ctxType> _localctx, int actionIndex) {

```

```

switch (actionIndex) {
  <actions:{index|
case <index>:
  <actions.(index)>
  break;}; separator="\n">
  }
}
>>

/* This generates a private method since the predIndex is generated, making an
 * overriding implementation impossible to maintain.
 */
RuleSempredFunction(r, actions) ::= <<
private boolean <r.name>_sempred(<r.ctxType> _localctx, int predIndex) {
  switch (predIndex) {
    <actions:{index|
case <index>:
  return <actions.(index)>;}; separator="\n">
  }
  return true;
}
>>

RuleFunction(currentRule,args,code,locals,ruleCtx,altLabelCtxs,namedActions,finallyAction,postamble,exceptions)
::= <<

<ruleCtx>
<altLabelCtxs:{l | <altLabelCtxs.(l)>}; separator="\n">

<if(currentRule.modifiers)><currentRule.modifiers:{f | <f> }><else>public final <endif><currentRule.ctxType>
<currentRule.name><(<args; separator=",">) throws RecognitionException {
  <currentRule.ctxType> _localctx = new <currentRule.ctxType>(<_ctx, getState()<currentRule.args:{a | ,
<a.name>}>>);
  enterRule(_localctx, <currentRule.startState>, RULE_<currentRule.name>);
  <namedActions.init>
  <locals; separator="\n">
  try {
    <if(currentRule.hasLookaheadBlock)>
      int _alt;
    <endif>
    <code>
    <postamble; separator="\n">
    <namedActions.after>
  }
  <if(exceptions)>
    <exceptions; separator="\n">
  <else>
    catch (RecognitionException re) {

```

```

    _localctx.exception = re;
    _errHandler.reportError(this, re);
    _errHandler.recover(this, re);
}
<endif>
finally {
    <finallyAction>
    exitRule();
}
return _localctx;
}
>>

```

```

LeftRecursiveRuleFunction(currentRule,args,code,locals,ruleCtx,altLabelCtxs,
    namedActions,finallyAction,postamble) ::=
<<

```

```

<ruleCtx>
<altLabelCtxs:{l | <altLabelCtxs.(l)>}; separator="\n">

```

```

<if(currentRule.modifiers)><currentRule.modifiers:{f | <f> }><else>public final <endif><currentRule.ctxType>
<currentRule.name><(args; separator=", ">) throws RecognitionException {
    return <currentRule.name>(0<currentRule.args:{a | , <a.name>}>);
}

```

```

private <currentRule.ctxType> <currentRule.name>(int _p<args:{a | , <a>}>) throws RecognitionException {
    ParserRuleContext _parentctx = _ctx;
    int _parentState = getState();
    <currentRule.ctxType> _localctx = new <currentRule.ctxType>(_ctx, _parentState<currentRule.args:{a | ,
    <a.name>}>);
    <currentRule.ctxType> _prevctx = _localctx;
    int _startState = <currentRule.startState>;
    enterRecursionRule(_localctx, <currentRule.startState>, RULE_<currentRule.name>, _p);
    <namedActions.init>
    <locals; separator="\n">
    try {
    <if(currentRule.hasLookaheadBlock)>
        int _alt;
    <endif>
        <code>
        <postamble; separator="\n">
        <namedActions.after>
    }
    catch (RecognitionException re) {
        _localctx.exception = re;
        _errHandler.reportError(this, re);
        _errHandler.recover(this, re);
    }
}

```

```

finally {
  <finallyAction>
  unrollRecursionContexts(_parentctx);
}
return _localctx;
}
>>

```

```

CodeBlockForOuterMostAlt(currentOuterMostAltCodeBlock, locals, preamble, ops) ::= <<
<if(currentOuterMostAltCodeBlock.altLabel)>_localctx = new <currentOuterMostAltCodeBlock.altLabel;
format="cap">Context(_localctx);<endif>
enterOuterAlt(_localctx, <currentOuterMostAltCodeBlock.alt.altNum>);
<CodeBlockForAlt(currentAltCodeBlock=currentOuterMostAltCodeBlock, ...)>
>>

```

```

CodeBlockForAlt(currentAltCodeBlock, locals, preamble, ops) ::= <<
{
  <locals; separator="\n">
  <preamble; separator="\n">
  <ops; separator="\n">
}
>>

```

```

LL1AltBlock(choice, preamble, alts, error) ::= <<
setState(<choice.stateNumber>);
_errHandler.sync(this);
<if(choice.label)><labelref(choice.label)> = _input.LT(1);<endif>
<preamble; separator="\n">
switch (_input.LA(1)) {
<choice.altLook,alts:{look,alt| <cases(ttypes=look)>
  <alt>
  break;}; separator="\n">
default:
  <error>
}
>>

```

```

LL1OptionalBlock(choice, alts, error) ::= <<
setState(<choice.stateNumber>);
_errHandler.sync(this);
switch (_input.LA(1)) {
<choice.altLook,alts:{look,alt| <cases(ttypes=look)>
  <alt>
  break;}; separator="\n">
default:
  break;
}
>>

```

```

LL1OptionalBlockSingleAlt(choice, expr, alts, preamble, error, followExpr) ::= <<
setState(<choice.stateNumber>);
_errHandler.sync(this);
<preamble; separator="\n">
if (<expr>) {
  <alts; separator="\n">
}
<!else if ( !(<followExpr> ) <error!>
>>

```

```

LL1StarBlockSingleAlt(choice, loopExpr, alts, preamble, iteration) ::= <<
setState(<choice.stateNumber>);
_errHandler.sync(this);
<preamble; separator="\n">
while (<loopExpr>) {
  <alts; separator="\n">
  setState(<choice.loopBackStateNumber>);
  _errHandler.sync(this);
  <iteration>
}
>>

```

```

LL1PlusBlockSingleAlt(choice, loopExpr, alts, preamble, iteration) ::= <<
setState(<choice.blockStartStateNumber>); <! alt block decision !>
_errHandler.sync(this);
<preamble; separator="\n">
do {
  <alts; separator="\n">
  setState(<choice.stateNumber>); <! loopback/exit decision !>
  _errHandler.sync(this);
  <iteration>
} while ( <loopExpr> );
>>

```

// LL(*) stuff

```

AltBlock(choice, preamble, alts, error) ::= <<
setState(<choice.stateNumber>);
_errHandler.sync(this);
<if(choice.label)><labelref(choice.label)> = _input.LT(1);<endif>
<preamble; separator="\n">
switch ( getInterpreter().adaptivePredict(_input,<choice.decision>,_ctx) ) {
  <alts:{alt |
case <i>:
  <alt>
break;}; separator="\n">
}

```

>>

```
OptionalBlock(choice, alts, error) ::= <<
setState(<choice.stateNumber>);
_errHandler.sync(this);
switch ( getInterpreter().adaptivePredict(_input,<choice.decision>,_ctx) ) {
<alts:{ alt |
case <i><if(!choice.ast.greedy)>+1<endif>:
<alt>
break;}; separator="\n">
}
}>>
```

```
StarBlock(choice, alts, sync, iteration) ::= <<
setState(<choice.stateNumber>);
_errHandler.sync(this);
_alt = getInterpreter().adaptivePredict(_input,<choice.decision>,_ctx);
while ( _alt!=<choice.exitAlt> && _alt!=org.antlr.v4.runtime.atn.ATN.INVALID_ALT_NUMBER ) {
if ( _alt==1<if(!choice.ast.greedy)>+1<endif> ) {
<iteration>
<alts> <! should only be one !>
}
setState(<choice.loopBackStateNumber>);
_errHandler.sync(this);
_alt = getInterpreter().adaptivePredict(_input,<choice.decision>,_ctx);
}
}>>
```

```
PlusBlock(choice, alts, error) ::= <<
setState(<choice.blockStartStateNumber>); <! alt block decision !>
_errHandler.sync(this);
_alt = 1<if(!choice.ast.greedy)>+1<endif>;
do {
switch ( _alt ) {
<alts:{ alt|
case <i><if(!choice.ast.greedy)>+1<endif>:
<alt>
break;}; separator="\n">
default:
<error>
}
setState(<choice.loopBackStateNumber>); <! loopback/exit decision !>
_errHandler.sync(this);
_alt = getInterpreter().adaptivePredict(_input,<choice.decision>,_ctx);
} while ( _alt!=<choice.exitAlt> && _alt!=org.antlr.v4.runtime.atn.ATN.INVALID_ALT_NUMBER );
}>>
```

```
Sync(s) ::= "sync(<s.expecting.name>);"
```

```

ThrowNoViableAlt(t) ::= "throw new NoViableAltException(this);"

TestSetInline(s) ::= <<
<s.bitsets:{bits | <if(rest(rest(bits.ttypes))>><bitsetBitfieldComparison(s, bits)><else><bitsetInlineComparison(s,
bits)><endif>}; separator=" || ">
>>

// Java language spec 15.19 - shift operators mask operands rather than overflow to 0... need range test
testShiftInRange(shiftAmount) ::= <<
((<shiftAmount>) & ~0x3f) == 0
>>

// produces smaller bytecode only when bits.ttypes contains more than two items
bitsetBitfieldComparison(s, bits) ::= <%
(<testShiftInRange({<offsetShift(s.varName, bits.shift)>})> && ((1L \<< <offsetShift(s.varName, bits.shift)>) &
(<bits.ttypes:{ttype | (1L \<< <offsetShift(ttype, bits.shift)>)}; separator=" | ">)) != 0)
%>

isZero ::= [
"0":true,
default:false
]

offsetShift(shiftAmount, offset) ::= <%
<if(!isZero.(offset))><shiftAmount> - <offset><else><shiftAmount><endif>
%>

// produces more efficient bytecode when bits.ttypes contains at most two items
bitsetInlineComparison(s, bits) ::= <%
<bits.ttypes:{ttype | <s.varName>==<ttype>}; separator=" || ">
%>

cases(ttypes) ::= <<
<ttypes:{t | case <t>}; separator="\n">
>>

InvokeRule(r, argExprsChunks) ::= <<
setState(<r.stateNumber>);
<if(r.labels)><r.labels:{l | <labelref(l)> =
}><endif><r.name><(<if(r.ast.options.p)><r.ast.options.p><if(argExprsChunks)><endif><endif><argExprsChunks>
);
>>

MatchToken(m) ::= <<
setState(<m.stateNumber>);
<if(m.labels)><m.labels:{l | <labelref(l)> = }><endif>match(<m.name>);
>>

```

```
MatchSet(m, expr, capture) ::= "<CommonSetStuff(m, expr, capture, false)>"
```

```
MatchNotSet(m, expr, capture) ::= "<CommonSetStuff(m, expr, capture, true)>"
```

```
CommonSetStuff(m, expr, capture, invert) ::= <<
setState(<m.stateNumber>);
<if(m.labels)><m.labels:{1 | <labelref(l)> = }>_input.LT(1);<endif>
<capture>
if ( <if(invert)><m.varName> \<= 0 || <else>!<endif>(<expr> ) ) {
  <if(m.labels)><m.labels:{1 | <labelref(l)> = (Token)}><endif>_errHandler.recoverInline(this);
}
else {
  if ( _input.LA(1)==Token.EOF ) matchedEOF = true;
  _errHandler.reportMatch(this);
  consume();
}
>>
```

```
Wildcard(w) ::= <<
setState(<w.stateNumber>);
<if(w.labels)><w.labels:{1 | <labelref(l)> = }><endif>matchWildcard();
>>
```

```
// ACTION STUFF
```

```
Action(a, foo, chunks) ::= "<chunks>"
```

```
ArgAction(a, chunks) ::= "<chunks>"
```

```
SemPred(p, chunks, failChunks) ::= <<
setState(<p.stateNumber>);
if (!(<chunks>)) throw new FailedPredicateException(this, <p.predicate><if(failChunks)>,
<failChunks><elseif(p.msg)>, <p.msg><endif>);
>>
```

```
ExceptionClause(e, catchArg, catchAction) ::= <<
catch (<catchArg>) {
  <catchAction>
}
>>
```

```
// lexer actions are not associated with model objects
```

```
LexerSkipCommand() ::= "skip()";
```

```
LexerMoreCommand() ::= "more()";
```

```
LexerPopModeCommand() ::= "popMode()";
```

```

LexerTypeCommand(arg, grammar) ::= "_type = <arg>";
LexerChannelCommand(arg, grammar) ::= "_channel = <arg>";
LexerModeCommand(arg, grammar) ::= "_mode = <arg>";
LexerPushModeCommand(arg, grammar) ::= "pushMode(<arg>);"

ActionText(t) ::= "<t.text>"
ActionTemplate(t) ::= "<t.st>"
ArgRef(a) ::= "_localctx.<a.name>"
LocalRef(a) ::= "_localctx.<a.name>"
RetValRef(a) ::= "_localctx.<a.name>"
QRetValRef(a) ::= "<ctx(a)>.<a.dict>.<a.name>"
/** How to translate $tokenLabel */
TokenRef(t) ::= "<ctx(t)>.<t.name>"
LabelRef(t) ::= "<ctx(t)>.<t.name>"
ListLabelRef(t) ::= "<ctx(t)>.<ListLabelName(t.name)>"
SetAttr(s,rhsChunks) ::= "<ctx(s)>.<s.name> = <rhsChunks>";

TokenLabelType() ::= "<file.TokenLabelType; null={Token}>"
InputSymbolType() ::= "<file.InputSymbolType; null={Token}>"

TokenPropertyRef_text(t) ::= "<ctx(t)>.<t.label>!=null?<ctx(t)>.<t.label>.getText():null)"
TokenPropertyRef_type(t) ::= "<ctx(t)>.<t.label>!=null?<ctx(t)>.<t.label>.getType():0)"
TokenPropertyRef_line(t) ::= "<ctx(t)>.<t.label>!=null?<ctx(t)>.<t.label>.getLine():0)"
TokenPropertyRef_pos(t) ::= "<ctx(t)>.<t.label>!=null?<ctx(t)>.<t.label>.getCharPositionInLine():0)"
TokenPropertyRef_channel(t) ::= "<ctx(t)>.<t.label>!=null?<ctx(t)>.<t.label>.getChannel():0)"
TokenPropertyRef_index(t) ::= "<ctx(t)>.<t.label>!=null?<ctx(t)>.<t.label>.getTokenIndex():0)"
TokenPropertyRef_int(t) ::= "<ctx(t)>.<t.label>!=null?Integer.valueOf(<ctx(t)>.<t.label>.getText()):0)"

RulePropertyRef_start(r) ::= "<ctx(r)>.<r.label>!=null?<ctx(r)>.<r.label>.start():null)"
RulePropertyRef_stop(r) ::= "<ctx(r)>.<r.label>!=null?<ctx(r)>.<r.label>.stop():null)"
RulePropertyRef_text(r) ::=
"<ctx(r)>.<r.label>!=null?_input.getText(<ctx(r)>.<r.label>.start,<ctx(r)>.<r.label>.stop):null)"
RulePropertyRef_ctx(r) ::= "<ctx(r)>.<r.label>"
RulePropertyRef_parser(r) ::= "this"

ThisRulePropertyRef_start(r) ::= "_localctx.start"
ThisRulePropertyRef_stop(r) ::= "_localctx.stop"
ThisRulePropertyRef_text(r) ::= "_input.getText(_localctx.start, _input.LT(-1))"
ThisRulePropertyRef_ctx(r) ::= "_localctx"
ThisRulePropertyRef_parser(r) ::= "this"

NonLocalAttrRef(s) ::= "(<s.ruleName; format=\"cap\">Context)getInvokingContext(<s.ruleIndex>).<s.name>"
SetNonLocalAttr(s, rhsChunks) ::=
"(<s.ruleName; format=\"cap\">Context)getInvokingContext(<s.ruleIndex>).<s.name> = <rhsChunks>";

AddToLabelList(a) ::= "<ctx(a.label)>.<a.listName>.add(<labelref(a.label)>);"

TokenDecl(t) ::= "<TokenLabelType()> <t.name>"

```

```

TokenTypeDecl(t) ::= "int <t.name>";
TokenListDecl(t) ::= "List<<Token> <t.name> = new ArrayList<<Token>()"
RuleContextDecl(r) ::= "<r.ctxName> <r.name>"
RuleContextListDecl(rdecl) ::= "List<<<rdecl.ctxName>> <rdecl.name> = new ArrayList<<<rdecl.ctxName>>()"

ContextTokenGetterDecl(t) ::=
    "public TerminalNode <t.name>() { return getToken(<parser.name>.<t.name>, 0); }"
ContextTokenListGetterDecl(t) ::=
    "public List<<TerminalNode> <t.name>() { return getTokens(<parser.name>.<t.name>); }"
ContextTokenListIndexedGetterDecl(t) ::= <<
public TerminalNode <t.name>(int i) {
    return getToken(<parser.name>.<t.name>, i);
}
>>
ContextRuleGetterDecl(r) ::= <<
public <r.ctxName> <r.name>() {
    return getRuleContext(<r.ctxName>.class,0);
}
>>
ContextRuleListGetterDecl(r) ::= <<
public List<<<r.ctxName>> <r.name>() {
    return getRuleContexts(<r.ctxName>.class);
}
>>
ContextRuleListIndexedGetterDecl(r) ::= <<
public <r.ctxName> <r.name>(int i) {
    return getRuleContext(<r.ctxName>.class,i);
}
>>

LexerRuleContext() ::= "RuleContext"

/** The rule context name is the rule followed by a suffix; e.g.,
 * r becomes rContext.
 */
RuleContextNameSuffix() ::= "Context"

ImplicitTokenLabel(tokenName) ::= "<tokenName>"
ImplicitRuleLabel(ruleName) ::= "<ruleName>"
ImplicitSetLabel(id) ::= "_tset<id>"
ListLabelName(label) ::= "<label>"

CaptureNextToken(d) ::= "<d.varName> = _input.LT(1);"
CaptureNextTokenType(d) ::= "<d.varName> = _input.LA(1);"

StructDecl(struct,ctorAttrs,attrs,getters,dispatchMethods,interfaces,extensionMembers)
::= <<
public static class <struct.name> extends

```

```

<if(contextSuperClass)><contextSuperClass><else>ParserRuleContext<endif><if(interfaces)> implements
<interfaces; separator=", "><endif> {
  <attrs:{ a | public <a>;}; separator="\n">
  <getters:{ g | <g>;}; separator="\n">
  <if(ctorAttrs)>public <struct.name>(ParserRuleContext parent, int invokingState) { super(parent, invokingState);
  }<endif>
  public <struct.name>(ParserRuleContext parent, int invokingState<ctorAttrs:{ a | , <a>}>) {
    super(parent, invokingState);
    <struct.ctorAttrs:{ a | this.<a.name> = <a.name>;}; separator="\n">
  }
  @Override public int getRuleIndex() { return RULE_<struct.derivedFromName>; }
  <if(struct.provideCopyFrom)> <! don't need copy unless we have subclasses !>
  public <struct.name>() { }
  public void copyFrom(<struct.name> ctx) {
    super.copyFrom(ctx);
    <struct.attrs:{ a | this.<a.name> = ctx.<a.name>;}; separator="\n">
  }
  <endif>
  <dispatchMethods; separator="\n">
  <extensionMembers; separator="\n">
  }
  >>

```

```

AltLabelStructDecl(struct,attrs,getters,dispatchMethods) ::= <<
public static class <struct.name> extends <currentRule.name; format="cap">Context {
  <attrs:{ a | public <a>;}; separator="\n">
  <getters:{ g | <g>;}; separator="\n">
  public <struct.name>(<currentRule.name; format="cap">Context ctx) { copyFrom(ctx); }
  <dispatchMethods; separator="\n">
  }
  >>

```

```

ListenerDispatchMethod(method) ::= <<
@Override
public void <if(method.isEnter)>enter<else>exit<endif>Rule(ParseTreeListener listener) {
  if ( listener instanceof <parser.grammarName>Listener )
  ((<parser.grammarName>Listener)listener).<if(method.isEnter)>enter<else>exit<endif><struct.derivedFromName;
  format="cap">(this);
  }
  >>

```

```

VisitorDispatchMethod(method) ::= <<
@Override
public <T> T accept(ParseTreeVisitor<? extends T> visitor) {
  if ( visitor instanceof <parser.grammarName>Visitor ) return ((<parser.grammarName>Visitor<? extends
  T>)visitor).visit<struct.derivedFromName; format="cap">(this);
  else return visitor.visitChildren(this);
  }
  >>

```

>>

```
AttributeDecl(d) ::= "<d.type> <d.name><if(d.initValue)> = <d.initValue><endif>"
```

```
/** If we don't know location of label def x, use this template */
```

```
labelref(x) ::= "<if(!x.isLocal)>((<x.ctx.name>)_localctx).<endif><x.name>"
```

```
/** For any action chunk, what is correctly-typed context struct ptr? */
```

```
ctx(actionChunk) ::= "((<actionChunk.ctx.name>)_localctx)"
```

```
// used for left-recursive rules
```

```
recRuleAltPredicate(ruleName,opPrec) ::= "precpred(_ctx, <opPrec>)"
```

```
recRuleSetReturnAction(src,name) ::= "$<name>=<src>.<name>";
```

```
recRuleSetStopToken() ::= "_ctx.stop = _input.LT(-1);"
```

```
recRuleAltStartAction(ruleName, ctxName, label, isListLabel) ::= <<
```

```
_localctx = new <ctxName>Context(_parentctx, _parentState);
```

```
<if(label)>
```

```
<if(isListLabel)>
```

```
_localctx.<label>.add(_prevctx);
```

```
<else>
```

```
_localctx.<label> = _prevctx;
```

```
<endif>
```

```
<endif>
```

```
<if(label)>_localctx.<label> = _prevctx;<endif>
```

```
pushNewRecursionContext(_localctx, _startState, RULE_<ruleName>);
```

```
>>
```

```
recRuleLabeledAltStartAction(ruleName, currentAltLabel, label, isListLabel) ::= <<
```

```
_localctx = new <currentAltLabel; format="cap">Context(new <ruleName; format="cap">Context(_parentctx,  
_parentState));
```

```
<if(label)>
```

```
<if(isListLabel)>
```

```
((<currentAltLabel; format="cap">Context)_localctx).<label>.add(_prevctx);
```

```
<else>
```

```
((<currentAltLabel; format="cap">Context)_localctx).<label> = _prevctx;
```

```
<endif>
```

```
<endif>
```

```
pushNewRecursionContext(_localctx, _startState, RULE_<ruleName>);
```

```
>>
```

```
recRuleReplaceContext(ctxName) ::= <<
```

```
_localctx = new <ctxName>Context(_localctx);
```

```
_ctx = _localctx;
```

```
_prevctx = _localctx;
```

```
>>
```

```
recRuleSetPrevCtx() ::= <<
```

```

if ( _parseListeners!=null ) triggerExitRuleEvent();
_prevctx = _localctx;
>>

```

```

LexerFile(lexerFile, lexer, namedActions) ::= <<
<fileHeader(lexerFile.grammarFileName, lexerFile.ANTLRVersion)>
<if(lexerFile.genPackage)>
package <lexerFile.genPackage>;
<endif>
<namedActions.header>
import org.antlr.v4.runtime.Lexer;
import org.antlr.v4.runtime.CharStream;
import org.antlr.v4.runtime.Token;
import org.antlr.v4.runtime.TokenStream;
import org.antlr.v4.runtime.*;
import org.antlr.v4.runtime.atn.*;
import org.antlr.v4.runtime.dfa.DFA;
import org.antlr.v4.runtime.misc.*;

<lexer>
>>

```

```

Lexer(lexer, atn, actionFuncs, sempredFuncs, superClass) ::= <<
@SuppressWarnings({"all", "warnings", "unchecked", "unused", "cast"})
public class <lexer.name> extends <superClass; null="Lexer"> {
    static { RuntimeMetaData.checkVersion("<lexerFile.ANTLRVersion>", RuntimeMetaData.VERSION); }

    protected static final DFA[] _decisionToDFA;
    protected static final PredictionContextCache _sharedContextCache =
        new PredictionContextCache();
    <if(lexer.tokens)>
    public static final int
    <lexer.tokens:{k | <k>=<lexer.tokens.(k)>}; separator=", ", wrap, anchor>;
    <endif>
    <if(lexer.channels)>
    public static final int
    <lexer.channels:{c | <c>=<lexer.channels.(c)>}; separator=", ", wrap, anchor>;
    <endif>
    <if(rest(lexer.modes)>
    public static final int
    <rest(lexer.modes){m | <m>=<i>}; separator=", ", wrap, anchor>;
    <endif>
    public static String[] channelNames = {
        "DEFAULT_TOKEN_CHANNEL", "HIDDEN"<if (lexer.channels)>, <lexer.channels:{c| "<c>"}; separator=", ",
wrap, anchor><endif>
    };
}

```

```

public static String[] modeNames = {
    <lexer.modes:{m| "<m>"}; separator=", ", wrap, anchor>
};

private static String[] makeRuleNames() {
    return new String[] {
        <lexer.ruleNames:{r| "<r>"}; separator=", ", wrap, anchor>
    };
}
public static final String[] ruleNames = makeRuleNames();

<vocabulary(lexer.literalNames, lexer.symbolicNames)>

<namedActions.members>

public <lexer.name>(CharStream input) {
    super(input);
    _interp = new LexerATNSimulator(this, _ATN, _decisionToDFA, _sharedContextCache);
}

@Override
public String getGrammarFileName() { return "<lexer.grammarFileName>"; }

@Override
public String[] getRuleNames() { return ruleNames; }

@Override
public String getSerializedATN() { return _serializedATN; }

@Override
public String[] getChannelNames() { return channelNames; }

@Override
public String[] getModeNames() { return modeNames; }

@Override
public ATN getATN() { return _ATN; }

<dumpActions(lexer, "", actionFuncs, sempredFuncs)>
<atn>
}
>>

SerializedATN(model) ::= <<
<if(rest(model.segments))>
<! requires segmented representation !>
private static final int _serializedATNSegments = <length(model.segments)>;
<model.segments:{segment|private static final String _serializedATNSegment<i0> =

```

```

"<segment; wrap={"+<\n><\t>">"; separator="\n">
public static final String _serializedATN = Utils.join(
new String[] {
<model.segments:{segment | _serializedATNSegment<i0>}; separator=",\n">
},
""
);
<else>
<! only one segment, can be inlined !>
public static final String _serializedATN =
"<model.serialized; wrap={"+<\n><\t>">";
<endif>
public static final ATN _ATN =
new ATNDeserializer().deserialize(_serializedATN.toCharArray());
static {
_decisionToDFA = new DFA[_ATN.getNumberOfDecisions()];
for (int i = 0; i < _ATN.getNumberOfDecisions(); i++) {
_decisionToDFA[i] = new DFA(_ATN.getDecisionState(i), i);
}
<! org.antlr.v4.tool.DOTGenerator dot = new org.antlr.v4.tool.DOTGenerator(null);!>
<! System.out.println(dot.getDOT(_ATN.decisionToState.get(0), ruleNames, false));!>
<! System.out.println(dot.getDOT(_ATN.ruleToStartState[2], ruleNames, false));!>
}
>>

/** Using a type to init value map, try to init a type; if not in table
* must be an object, default value is "null".
*/
initValue(typeName) ::= <<
<javaTypeInitMap.(typeName)>
>>

codeFileExtension() ::= ".java"

```

Found in path(s):

```

* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/tool/templates/codegen/Java/Java.stg

```

No license file was found, but licenses were detected in source scan.

```

/*
* [The "BSD license"]
* Copyright (c) 2012-2016 Terence Parr
* Copyright (c) 2012-2016 Sam Harwell
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:

```

*
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * 3. The name of the author may not be used to endorse or promote products
 * derived from this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
 * IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
 * IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
 * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
 * DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
 * THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
 * THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */

```
phpTypeInitMap ::= [
  "int": "0",
  "long": "0",
  "float": "0.0",
  "double": "0.0",
  "boolean": "false",
  default: "null"
]
```

```
// args must be <object-model-object>, <fields-resulting-in-STs>
```

```
ParserFile(file, parser, namedActions, contextSuperClass) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
<parser>
>>
```

```
ListenerFile(file, header, namedActions) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
<if(file.genPackage)>
namespace <file.genPackage>;
<endif>
<header>
use Antlr\Antlr4\Runtime\Tree\ParseTreeListener;
```

```
/**
```

```
* This interface defines a complete listener for a parse tree produced by
* { @see <file.parserName> }.
```

```

*/
interface <file.grammarName>Listener extends ParseTreeListener {
  <file.listenerNames: {Iname |
/**
<if(file.listenerLabelRuleNames.(Iname))>
* Enter a parse tree produced by the `<Iname>`
* labeled alternative in { @see <file.parserName>::<file.listenerLabelRuleNames.(Iname)>()}.
<else>
* Enter a parse tree produced by { @see <file.parserName>::<Iname>()}.
<endif>
* @param $context The parse tree.
*/
public function enter<Iname; format="cap">(Context\|<Iname; format="cap">Context $context) : void;
/**
<if(file.listenerLabelRuleNames.(Iname))>
* Exit a parse tree produced by the `<Iname>` labeled alternative
* in { @see <file.parserName>::<file.listenerLabelRuleNames.(Iname)>()}.
<else>
* Exit a parse tree produced by { @see <file.parserName>::<Iname>()}.
<endif>
* @param $context The parse tree.
*/
public function exit<Iname; format="cap">(Context\|<Iname; format="cap">Context $context) : void;};
separator="\n">
}
>>

```

```

BaseListenerFile(file, header, namedActions) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
<if(file.genPackage)>
namespace <file.genPackage>;
<endif>
<header>

```

```

use Antlr\Antlr4\Runtime\ParserRuleContext;
use Antlr\Antlr4\Runtime\Tree\ErrorNode;
use Antlr\Antlr4\Runtime\Tree\TerminalNode;

```

```

/**
* This class provides an empty implementation of { @see <file.grammarName>Listener},
* which can be extended to create a listener which only needs to handle a subset
* of the available methods.
*/
class <file.grammarName>BaseListener implements <file.grammarName>Listener
{
  <file.listenerNames: {Iname |
/**
* { @inheritdoc}

```

```

*
* The default implementation does nothing.
*/
public function enter<lname; format="cap">(Context\<lname; format="cap">Context $context) : void {}

/**
 * {@inheritdoc}
 *
 * The default implementation does nothing.
 */
public function exit<lname; format="cap">(Context\<lname; format="cap">Context $context) : void {\};
separator="\n">

/**
 * {@inheritdoc}
 *
 * The default implementation does nothing.
 */
public function enterEveryRule(ParserRuleContext $context) : void {}

/**
 * {@inheritdoc}
 *
 * The default implementation does nothing.
 */
public function exitEveryRule(ParserRuleContext $context) : void {}

/**
 * {@inheritdoc}
 *
 * The default implementation does nothing.
 */
public function visitTerminal(TerminalNode $node) : void {}

/**
 * {@inheritdoc}
 *
 * The default implementation does nothing.
 */
public function visitErrorNode(ErrorNode $node) : void {}
}
>>

VisitorFile(file, header, namedActions) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
<if(file.genPackage)>
namespace <file.genPackage>;
<endif>

```

```

<header>
use Antlr\Antlr4\Runtime\Tree\ParseTreeVisitor;

/**
 * This interface defines a complete generic visitor for a parse tree produced by { @see <file.parserName> }.
 */
interface <file.grammarName>Visitor extends ParseTreeVisitor
{
    <file.visitorNames:{ lname |
/**
<if(file.visitorLabelRuleNames.(lname))>
 * Visit a parse tree produced by the `<lname>` labeled alternative
 * in { @see <file.parserName>::<file.visitorLabelRuleNames.(lname)>() }.
<else>
 * Visit a parse tree produced by { @see <file.parserName>::<lname>() }.
<endif>
 *
 * @param Context\<lname; format="cap">Context $context The parse tree.
 *
 * @return mixed The visitor result.
 */
public function visit<lname; format="cap">(Context\<lname; format="cap">Context $context);
separator="\n\n"
}
>>

BaseVisitorFile(file, header, namedActions) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
<if(file.genPackage)>
namespace <file.genPackage>;
<endif>
<header>
use Antlr\Antlr4\Runtime\Tree\AbstractParseTreeVisitor;

/**
 * This class provides an empty implementation of { @see <file.grammarName>Visitor },
 * which can be extended to create a visitor which only needs to handle a subset
 * of the available methods.
 */
class <file.grammarName>BaseVisitor extends AbstractParseTreeVisitor implements <file.grammarName>Visitor
{
    <file.visitorNames:{ lname |
/**
 * { @inheritdoc }
 *
 * The default implementation returns the result of calling
 * { @see self::visitChildren() } on `context`.

```

```

*/
public function visit<lname; format="cap">(Context\<lname; format="cap">Context $context)
{
    return $this->visitChildren($context);
\}}; separator="\n\n">
}
>>

```

```

fileHeader(grammarFileName, ANTLRVersion) ::= <<
\<?php

```

```

/*
* Generated from <grammarFileName> by ANTLR <ANTLRVersion>
*/

```

```

>>
Parser(parser, funcs, atn, sempredFuncs, superClass) ::= <<
<Parser_(ctor="parser_ctor", ...)>
>>

```

```

Parser_(parser, funcs, atn, sempredFuncs, ctor, superClass) ::= <<
namespace<if(file.genPackage)> <file.genPackage><endif> {
<if(namedActions.header)><namedActions.header><endif>
    use Antlr\Antlr4\Runtime\Atn\ATN;
    use Antlr\Antlr4\Runtime\Atn\ATNDeserializer;
    use Antlr\Antlr4\Runtime\Atn\ParserATNSimulator;
    use Antlr\Antlr4\Runtime\Dfa\DFA;
    use Antlr\Antlr4\Runtime\Error\Exceptions\FailedPredicateException;
    use Antlr\Antlr4\Runtime\Error\Exceptions\NoViableAltException;
    use Antlr\Antlr4\Runtime\PredictionContexts\PredictionContextCache;
    use Antlr\Antlr4\Runtime\Error\Exceptions\RecognitionException;
    use Antlr\Antlr4\Runtime\RuleContext;
    use Antlr\Antlr4\Runtime\Token;
    use Antlr\Antlr4\Runtime\TokenStream;
    use Antlr\Antlr4\Runtime\Vocabulary;
    use Antlr\Antlr4\Runtime\VocabularyImpl;
    use Antlr\Antlr4\Runtime\RuntimeMetaData;
    use Antlr\Antlr4\Runtime\Parser;
<if(namedActions.definitions)><namedActions.definitions><endif>

```

```

final class <parser.name> extends <superClass; null="Parser">
{
    <if(parser.tokens)>
    public const <parser.tokens>:{k | <k> = <parser.tokens.(k)>}; separator=", ", wrap, anchor>;
    <endif>

    public const <parser.rules>:{r | RULE_<r.name> = <r.index>}; separator=", ", wrap, anchor>;

```

```

/**
 * @var array<string>
 */
public const RULE_NAMES = [
<parser.ruleNames:{r | '<r'>}; separator=", ", wrap, anchor>
];

<vocabulary(parser.literalNames, parser.symbolicNames)>

<atn>
protected static $atn;
protected static $decisionToDFA;
protected static $sharedContextCache;
<if(namedActions.members)>

<namedActions.members>
<endif>

<parser:(ctor())>

private static function initialize() : void
{
if (self::$atn !== null) {
return;
}

RuntimeMetaData::checkVersion('<file.ANTLRVersion>', RuntimeMetaData::VERSION);

$atn = (new ATNDeserializer()->deserialize(self::SERIALIZED_ATN);

$decisionToDFA = [];
for ($i = 0, $count = $atn->getNumberOfDecisions(); $i < $count; $i++) {
$decisionToDFA[] = new DFA($atn->getDecisionState($i), $i);
}

self::$atn = $atn;
self::$decisionToDFA = $decisionToDFA;
self::$sharedContextCache = new PredictionContextCache();
}

public function getGrammarFileName() : string
{
return "<parser.grammarFileName>";
}

public function getRuleNames() : array
{

```

```

return self::RULE_NAMES;
}

public function getSerializedATN() : string
{
return self::SERIALIZED_ATN;
}

public function getATN() : ATN
{
return self::$atn;
}

public function getVocabulary() : Vocabulary
{
static $vocabulary;

return $vocabulary = $vocabulary ?? new VocabularyImpl(self::LITERAL_NAMES, self::SYMBOLIC_NAMES);
}
<if(funcs)>

<funcs; separator="\n\n">
<endif>
<if(sempredFuncs)>

public function sempred(?RuleContext $localContext, int $ruleIndex, int $predicateIndex) : bool
{
switch ($ruleIndex) {
<parser.sempredFuncs.values:{f}
case <f.ruleIndex>:
return $this->sempred<f.name; format="cap">($localContext, $predicateIndex);}; separator="\n\n">

default:
return true;
}
}

<sempredFuncs.values; separator="\n\n">
<endif>
}
}

namespace <if(file.genPackage)><file.genPackage>\<endif>Context {
use Antlr\Antlr4\Runtime\ParserRuleContext;
use Antlr\Antlr4\Runtime\Token;
use Antlr\Antlr4\Runtime\Tree\ParseTreeVisitor;
use Antlr\Antlr4\Runtime\Tree\TerminalNode;
use Antlr\Antlr4\Runtime\Tree\ParseTreeListener;

```

```

use <if(file.genPackage)><file.genPackage>\\<endif><parser.name>;
<if (file.genVisitor)>use <if(file.genPackage)><file.genPackage>\\<endif><file.grammarName>Visitor;<endif>
<if (file.genListener)>use <if(file.genPackage)><file.genPackage>\\<endif><file.grammarName>Listener;<endif>
<namedActions.contexts>

<funcs :{ func | <func.ruleCtx><if(func.altLabelCtxs)>

<func.altLabelCtxs:{1 | <func.altLabelCtxs.(1)>; separator="\n\n"><endif> }; separator="\n\n">
}
>>

vocabulary(literalNames, symbolicNames) ::= <<
/**
 * @var array<string|null>
 */
private const LITERAL_NAMES = [
  <literalNames:{t | <t>; null="null", separator=", ", wrap, anchor>
];

/**
 * @var array<string>
 */
private const SYMBOLIC_NAMES = [
  <symbolicNames:{t | <t>; null="null", separator=", ", wrap, anchor>
];
>>

dumpActions(recog, argFuncs, actionFuncs, sempredFuncs) ::= <<
<if(actionFuncs)>

public function action(?RuleContext $localContext, int $ruleIndex, int $actionIndex) : void
{
  switch ($ruleIndex) {
    <recog.actionFuncs.values:{f}
    case <f.ruleIndex>:
      $this->action<f.name; format="cap">($localContext, $actionIndex);
      break; }; separator="\n\n">
  }
}

<actionFuncs.values; separator="\n">
<endif>
<if(sempredFuncs)>

public function sempred(?RuleContext $localContext, int $ruleIndex, int $predicateIndex) : bool
{
  switch ($ruleIndex) {
    <recog.sempredFuncs.values:{f}

```

```

    case <f.ruleIndex>:
        return $this->sempred<f.name; format="cap">($localContext, $predicateIndex);}; separator="\n\n">
    }

return true;
}
<sempredFuncs.values; separator="\n\n">
<endif>
>>

parser_ctor(p) ::= <<
public function __construct(TokenStream $input)
{
    parent::__construct($input);

    self::initialize();

    $this->interp = new ParserATNSimulator($this, self::$atn, self::$decisionToDFA, self::$sharedContextCache);
}
>>

/**
 * This generates a private method since the actionIndex is generated, making
 * an overriding implementation impossible to maintain.
 */
RuleActionFunction(r, actions) ::= <<
private function action<r.name; format="cap">(Context $localContext, int $actionIndex) : void
{
    switch ($actionIndex) {
        <actions:{index|
        case <index>:
            <actions.(index)>

        break;}; separator="\n\n">
    }
}
>>

/**
 * This generates a private method since the predicateIndex is generated, making
 * an overriding implementation impossible to maintain.
 */
RuleSempredFunction(r, actions) ::= <<
private function sempred<r.name; format="cap">(Context $localContext, int $predicateIndex) : bool
{
    switch ($predicateIndex) {
        <actions:{index|
        case <index>:

```

```

        return <actions.(index)>;}; separator="\n\n">
    }

    return true;
}
>>

RuleFunction(currentRule,args,code,locals,ruleCtx,altLabelCtxs,namedActions,finallyAction,exceptions,postamble)
::= <<
/**
 * @throws RecognitionException
 */
<if(currentRule.modifiers)><currentRule.modifiers:{f | <f> }><endif>public function <currentRule.name>(<args;
separator=", ">) : Context\|<currentRule.ctxType>
{
    $localContext = new Context\|<currentRule.ctxType>($this->ctx, $this->getState())<currentRule.args:{a | ,
$a.name}>>);

    $this->enterRule($localContext, <currentRule.startState>, self::RULE_<currentRule.name>);
    <namedActions.init>
    <locals; separator="\n\n">

    try {
        <code>
        <postamble; separator="\n\n">
        <namedActions.after>
    }<if(exceptions)><exceptions; separator="\n\n"><else> catch (RecognitionException $exception) {
        $localContext->exception = $exception;
        $this->errorHandler->reportError($this, $exception);
        $this->errorHandler->recover($this, $exception);
    }<endif> finally {
        <finallyAction>
        $this->exitRule();
    }

    return $localContext;
}
>>

LeftRecursiveRuleFunction(currentRule,args,code,locals,ruleCtx,altLabelCtxs,namedActions,finallyAction,postamb
le) ::= <<
/**
 * @throws RecognitionException
 */
<if(currentRule.modifiers)><currentRule.modifiers:{f | <f> }><endif>public function <currentRule.name>(<args;
separator=", ">) : Context\|<currentRule.ctxType>
{
    return $this->recursive<currentRule.name; format="cap">(0<currentRule.args:{a | , <a.name>>);
}

```

```

}

/**
 * @throws RecognitionException
 */
private function recursive<currentRule.name; format="cap">(int $precedence<args:{a | , <a>}>) :
Context\<currentRule.ctxType>
{
    $parentContext = $this->ctx;
    $parentState = $this->getState();
    $localContext = new Context\<currentRule.ctxType>($this->ctx, $parentState<currentRule.args:{a | ,
<a.name>}>);
    $previousContext = $localContext;
    $startState = <currentRule.startState>;
    $this->enterRecursionRule($localContext, <currentRule.startState>, self::RULE_<currentRule.name>,
$precedence);
    <namedActions.init>
    <locals; separator="\n">

    try {
        <code>
        <postamble; separator="\n">
        <namedActions.after>
    } catch (RecognitionException $exception) {
        $localContext->exception = $exception;
        $this->errorHandler->reportError($this, $exception);
        $this->errorHandler->recover($this, $exception);
    } finally {
        <finallyAction>
        $this->unrollRecursionContexts($parentContext);
    }

    return $localContext;
}
>>

```

```

CodeBlockForOuterMostAlt(currentOuterMostAltCodeBlock, locals, preamble, ops) ::= <<
<if(currentOuterMostAltCodeBlock.altLabel)>$localContext = new
Context\<currentOuterMostAltCodeBlock.altLabel; format="cap">Context($localContext);<endif>
$this->enterOuterAlt($localContext, <currentOuterMostAltCodeBlock.alt.altNum>);
<CodeBlockForAlt(currentAltCodeBlock=currentOuterMostAltCodeBlock, ...)>
>>

```

```

CodeBlockForAlt(currentAltCodeBlock, locals, preamble, ops) ::= <<
<locals; separator="\n">
<preamble; separator="\n">
<ops; separator="\n">
>>

```

```

LL1AltBlock(choice, preamble, alts, error) ::= <<
$this->setState(<choice.stateNumber>);
$this->errorHandler->sync($this);
<if(choice.label)><labelref(choice.label)> = $this->input->LT(1);<endif>
<preamble; separator="\n">

switch ($this->input->LA(1)) {
  <choice.altLook,alts:{look,alt| <cases(ttypes=look)>
  <alt>
  break;}; separator="\n\n">

default:
  <error>
}
>>

```

```

LL1OptionalBlock(choice, alts, error) ::= <<
$this->setState(<choice.stateNumber>);
$this->errorHandler->sync($this);

switch ($this->input->LA(1)) {
  <choice.altLook,alts:{look,alt| <cases(ttypes=look)>
  <alt>
  break;}; separator="\n\n">

default:
  break;
}
>>

```

```

LL1OptionalBlockSingleAlt(choice, expr, alts, preamble, error, followExpr) ::= <<
$this->setState(<choice.stateNumber>);
$this->errorHandler->sync($this);
<preamble; separator="\n">

if (<expr>) {
  <alts; separator="\n">
}
>>

```

```

LL1StarBlockSingleAlt(choice, loopExpr, alts, preamble, iteration) ::= <<
$this->setState(<choice.stateNumber>);
$this->errorHandler->sync($this);

<preamble; separator="\n">
while (<loopExpr>) {
  <alts; separator="\n">
}
>>

```

```

$this->setState(<choice.loopBackStateNumber>);
$this->errorHandler->sync($this);
<iteration>
}
>>

```

```

LL1PlusBlockSingleAlt(choice, loopExpr, alts, preamble, iteration) ::= <<
$this->setState(<choice.blockStartStateNumber>); <! alt block decision !>
$this->errorHandler->sync($this);

```

```

<preamble; separator="\n">
do {
<alts; separator="\n">
$this->setState(<choice.stateNumber>); <! loopback/exit decision !>
$this->errorHandler->sync($this);
<iteration>
} while (<loopExpr>);
>>

```

```

// LL(*) stuff

```

```

AltBlock(choice, preamble, alts, error) ::= <<
$this->setState(<choice.stateNumber>);
$this->errorHandler->sync($this);
<if(choice.label)><labelref(choice.label)> = $this->input->LT(1);<endif>
<preamble; separator="\n">

```

```

switch ($this->getInterpreter()->adaptivePredict($this->input, <choice.decision>, $this->ctx)) {
<alts:{ alt |
case <i>:
<alt>
break;}; separator="\n\n">
}
>>

```

```

OptionalBlock(choice, alts, error) ::= <<
$this->setState(<choice.stateNumber>);
$this->errorHandler->sync($this);

```

```

switch ($this->getInterpreter()->adaptivePredict($this->input, <choice.decision>, $this->ctx)) {
<alts:{ alt |
case <i><if(!choice.ast.greedy)>+1<endif>:
<alt>
break;}; separator="\n\n">
}
>>

```

```

StarBlock(choice, alts, sync, iteration) ::= <<

```

```

$this->setState(<choice.stateNumber>);
$this->errorHandler->sync($this);

$alt = $this->getInterpreter()->adaptivePredict($this->input, <choice.decision>, $this->ctx);

while ($alt !== <choice.exitAlt> && $alt !== ATN::INVALID_ALT_NUMBER) {
if ($alt === 1<if(!choice.ast.greedy)>+1<endif>) {
    <iteration>
    <alts> <! should only be one !>
}

$this->setState(<choice.loopBackStateNumber>);
$this->errorHandler->sync($this);

$alt = $this->getInterpreter()->adaptivePredict($this->input, <choice.decision>, $this->ctx);
}
>>

PlusBlock(choice, alts, error) ::= <<
$this->setState(<choice.blockStartStateNumber>); <! alt block decision !>
$this->errorHandler->sync($this);

$alt = 1<if(!choice.ast.greedy)>+1<endif>;

do {
switch ($alt) {
<alts:{alt|
case <i><if(!choice.ast.greedy)>+1<endif>:
<alt>
break;}; separator="\n\n">
default:
<error>
}

$this->setState(<choice.loopBackStateNumber>); <! loopback/exit decision !>
$this->errorHandler->sync($this);

$alt = $this->getInterpreter()->adaptivePredict($this->input, <choice.decision>, $this->ctx);
} while ($alt !== <choice.exitAlt> && $alt !== ATN::INVALID_ALT_NUMBER);
>>

Sync(s) ::= "sync(<s.expecting.name>);"

ThrowNoViableAlt(t) ::= "throw new NoViableAltException($this);"

TestSetInline(s) ::= <<
<s.bitsets:{bits | <if(rest(rest(bits.ttypes)))><bitsetBitfieldComparison(s, bits)><else><bitsetInlineComparison(s,
bits)><endif>}; separator=" || ">

```

```

>>

// Java language spec 15.19 - shift operators mask operands rather than overflow to 0... need range test
testShiftInRange(shiftAmount) ::= <<
((<shiftAmount>) & ~0x3f) === 0
>>

// produces smaller bytecode only when bits.ttypes contains more than two items
bitsetBitfieldComparison(s, bits) ::= <%
(<testShiftInRange(<offsetShiftVar(s.varName, bits.shift)>)> && ((1 \<< <offsetShiftVar(s.varName,
bits.shift)>) & (<bits.ttypes:{ttype | (1 \<< <offsetShiftConst(ttype, bits.shift)>)}; separator=" | ">)) !== 0)
%>

isZero ::= [
"0":true,
default:false
]

offsetShiftVar(shiftAmount, offset) ::= <%
<if(!isZero.(offset))>(<shiftAmount> - <offset>)<else><shiftAmount><endif>
%>
offsetShiftConst(shiftAmount, offset) ::= <%
<if(!isZero.(offset))>(self::<shiftAmount> - <offset>)<else>self::<shiftAmount><endif>
%>

// produces more efficient bytecode when bits.ttypes contains at most two items
bitsetInlineComparison(s, bits) ::= <%
<bits.ttypes:{ttype | $<s.varName> === self::<ttype>}; separator=" || ">
%>

cases(ttypes) ::= <<
<ttypes:{t | case self::<t>:}; separator="\n">
>>

InvokeRule(r, argExprsChunks) ::= <<
$this->setState(<r.stateNumber>);
<if(r.labels)><r.labels:{1 | <labelref(l)> = }><endif>$this-><if(r.ast.options.p)>recursive<r.name>;
format="cap"><else><r.name><endif>(<if(r.ast.options.p)><r.ast.options.p><if(argExprsChunks)>,<endif><endif>
<argExprsChunks>);
>>

MatchToken(m) ::= <<
$this->setState(<m.stateNumber>);
<if(m.labels)><m.labels:{1 | <labelref(l)> = }><endif>$this->match(self::<m.name>);
>>

MatchSet(m, expr, capture) ::= "<CommonSetStuff(m, expr, capture, false)>"

```

```
MatchNotSet(m, expr, capture) ::= "<CommonSetStuff(m, expr, capture, true)>"
```

```
CommonSetStuff(m, expr, capture, invert) ::= <<  
$this->setState(<m.stateNumber>);
```

```
<if(m.labels)><m.labels:{1 | <labelref(l)> = }>$this->input->LT(1);<endif>  
<capture>
```

```
if (<if(invert)>$<m.varName> \<= 0 || <else>!<endif>(<expr>)) {  
    <if(m.labels)><m.labels:{1 | <labelref(l)> = }><endif>$this->errorHandler->recoverInline($this);  
} else {  
    if ($this->input->LA(1) === Token::EOF) {  
        $this->matchedEOF = true;  
    }  
}
```

```
$this->errorHandler->reportMatch($this);  
$this->consume();  
}  
>>
```

```
Wildcard(w) ::= <<  
$this->setState(<w.stateNumber>);  
<if(w.labels)><w.labels:{1 | <labelref(l)> = }><endif>$this->matchWildcard();  
>>
```

```
// ACTION STUFF
```

```
Action(a, foo, chunks) ::= "<chunks>"
```

```
ArgAction(a, chunks) ::= "<chunks>"
```

```
SemPred(p, chunks, failChunks) ::= <<  
$this->setState(<p.stateNumber>);
```

```
if (!(<chunks>)) {  
    throw new FailedPredicateException($this, <p.predicate><if(failChunks)>, <failChunks><elseif(p.msg)>,  
<p.msg><endif>);  
}  
>>
```

```
ExceptionClause(e, catchArg, catchAction) ::= <<  
catch (<catchArg>) {  
    <catchAction>  
}  
>>
```

```
// lexer actions are not associated with model objects
```

```

LexerSkipCommand() ::= "$this->skip();"
LexerMoreCommand() ::= "$this->more();"
LexerPopModeCommand() ::= "$this->popMode();"

LexerTypeCommand(arg, grammar) ::= "$this->type = <arg>";
LexerChannelCommand(arg, grammar) ::= "$this->channel = <arg>";
LexerModeCommand(arg, grammar) ::= "$this->mode = <arg>";
LexerPushModeCommand(arg, grammar) ::= "$this->pushMode(<arg>);"

ActionText(t) ::= "<t.text>"
ActionTemplate(t) ::= "<t.st>"
ArgRef(a) ::= "$localContext-><a.name>"
LocalRef(a) ::= "$localContext-><a.name>"
RetValRef(a) ::= "$localContext-><a.name>"
QRetValRef(a) ::= "<ctx(a)>-><a.dict>-><a.name>"
/** How to translate $tokenLabel */
TokenRef(t) ::= "<ctx(t)>-><t.name>"
LabelRef(t) ::= "<ctx(t)>-><t.name>"
ListLabelRef(t) ::= "<ctx(t)>-><ListLabelName(t.name)>"
SetAttr(s,rhsChunks) ::= "<ctx(s)>-><s.name> = <rhsChunks>";

TokenLabelType() ::= "<file.TokenLabelType; null={Token}>"
InputSymbolType() ::= "<file.InputSymbolType; null={Token}>"

TokenPropertyRef_text(t) ::= "<ctx(t)>-><t.label> !== null ? <ctx(t)>-><t.label>->getText() : null)"
TokenPropertyRef_type(t) ::= "<ctx(t)>-><t.label> !== null ? <ctx(t)>-><t.label>->getType() : 0)"
TokenPropertyRef_line(t) ::= "<ctx(t)>-><t.label> !== null ? <ctx(t)>-><t.label>->getLine() : 0)"
TokenPropertyRef_pos(t) ::= "<ctx(t)>-><t.label> !== null ? <ctx(t)>-><t.label>->getCharPositionInLine() : 0)"
TokenPropertyRef_channel(t) ::= "<ctx(t)>-><t.label> !== null ? <ctx(t)>-><t.label>->getChannel() : 0)"
TokenPropertyRef_index(t) ::= "<ctx(t)>-><t.label> !== null ? <ctx(t)>-><t.label>->getTokenIndex() : 0)"
TokenPropertyRef_int(t) ::= "<ctx(t)>-><t.label> !== null ? (int) <ctx(t)>-><t.label>->getText() : 0)"

RulePropertyRef_start(r) ::= "<ctx(r)>-><r.label> !== null ? (<ctx(r)>-><r.label>->start) : null)"
RulePropertyRef_stop(r) ::= "<ctx(r)>-><r.label> !== null ? (<ctx(r)>-><r.label>->stop) : null)"
RulePropertyRef_text(r) ::= "<ctx(r)>-><r.label> !== null ? $this->input->getTextByTokens(<ctx(r)>-><r.label>->start, <ctx(r)>-><r.label>->stop) : null)"
RulePropertyRef_ctx(r) ::= "<ctx(r)>-><r.label>"
RulePropertyRef_parser(r) ::= "\$this"

ThisRulePropertyRef_start(r) ::= "$localContext->start"
ThisRulePropertyRef_stop(r) ::= "$localContext->stop"
ThisRulePropertyRef_text(r) ::= "$this->input->getTextByTokens($localContext->start, $this->input->LT(-1))"
ThisRulePropertyRef_ctx(r) ::= "$localContext"
ThisRulePropertyRef_parser(r) ::= "$this"

NonLocalAttrRef(s) ::= "\$this->getInvokingContext(<s.ruleIndex>-><s.name>"
SetNonLocalAttr(s, rhsChunks) ::= "\$this->getInvokingContext(<s.ruleIndex>-><s.name> = <rhsChunks>";"

```

```
AddToLabelList(a) ::= "<ctx(a.label)>-><a.listName>[] = <labelref(a.label)>";
```

```
TokenDecl(t) ::= "<TokenLabelType()> $<t.name>"
```

```
TokenTypeDecl(t) ::= ""
```

```
TokenListDecl(t) ::= "array $<t.name> = []"
```

```
RuleContextDecl(r) ::= "<r.ctxName> $<r.name>"
```

```
RuleContextListDecl(rdecl) ::= "array $<rdecl.name> = []"
```

```
AttributeDecl(d) ::= "<d.type> $<d.name><if(d.initValue)> = <d.initValue><endif>"
```

```
PropertiesDecl(struct) ::= <<
```

```
<if(struct.tokenListDecls)>
```

```
<struct.tokenListDecls : {d | /**
```

```
* @var array<<Token>|null $<d.name>
```

```
*/
```

```
public $<d.name>;}; separator="\n\n">
```

```
<endif>
```

```
<if(struct.tokenDecls)>
```

```
<if(struct.tokenListDecls)>
```

```
<endif>
```

```
<struct.tokenDecls : {d | /**
```

```
* @var <TokenLabelType()>|null $<d.name>
```

```
*/
```

```
public $<d.name>;}; separator="\n\n">
```

```
<endif>
```

```
<if(struct.ruleContextDecls)>
```

```
<if(struct.tokenListDecls || struct.tokenDecls)>
```

```
<endif>
```

```
<struct.ruleContextDecls : {d | /**
```

```
* @var <d.ctxName>|null $<d.name>
```

```
*/
```

```
public $<d.name>;}; separator="\n\n">
```

```
<endif>
```

```
<if(struct.ruleContextListDecls)>
```

```
<if(struct.tokenListDecls || struct.tokenDecls || struct.ruleContextDecls)>
```

```
<endif>
```

```
<struct.ruleContextListDecls : {d | /**
```

```
* @var array<<<d.ctxName>>|null $<d.name>
```

```
*/
```

```
public $<d.name>;}; separator="\n\n">
```

```
<endif>
```

```
<if(struct.attributeDecls)>
```

```
<if(struct.tokenListDecls || struct.tokenDecls || struct.ruleContextDecls || struct.ruleContextListDecls)>
```

```
<endif>
```

```
<struct.attributeDecls : {d | /**
```

```

* @var <d.type><if(!d.initValue)>|null<endif> $<d.name>
*/
public $<d.name><if(d.initValue)> = <d.initValue><endif>;}; separator="\n\n">
<endif>

```

```
>>
```

```

ContextTokenGetterDecl(t) ::= <<
public function <t.name>() : ?TerminalNode
{
    return $this->getToken(<parser.name>::<t.name>, 0);
}
>>

```

```

ContextTokenListGetterDecl(t) ::= <<
>>

```

```

ContextTokenListIndexedGetterDecl(t) ::= <<
/**
* @return array<TerminalNode>|TerminalNode|null
*/
public function <t.name>(int $index = null)
{
    if ($index === null) {
        return $this->getTokens(<parser.name>::<t.name>);
    }

    return $this->getToken(<parser.name>::<t.name>, $index);
}
>>

```

```

ContextRuleGetterDecl(r) ::= <<
public function <r.name>() : ?<r.ctxName>
{
    return $this->getTypedRuleContext(<r.ctxName>::class, 0);
}
>>

```

```

ContextRuleListGetterDecl(r) ::= <<
>>

```

```

ContextRuleListIndexedGetterDecl(r) ::= <<
/**
* @return array<<r.ctxName>>|<r.ctxName>|null
*/
public function <r.name>(int $index = null)
{
    if ($index === null) {

```

```

return $this->getTypedRuleContexts(<r.ctxName>::class);
}

return $this->getTypedRuleContext(<r.ctxName>::class, $index);
}
>>

LexerRuleContext() ::= "RuleContext"

/**
 * The rule context name is the rule followed by a suffix; e.g., r becomes rContext.
 */
RuleContextNameSuffix() ::= "Context"

ImplicitTokenLabel(tokenName) ::= "<tokenName>"
ImplicitRuleLabel(ruleName) ::= "<ruleName>"
ImplicitSetLabel(id) ::= "_tset<id>"
ListLabelName(label) ::= "<label>"

CaptureNextToken(d) ::= "$<d.varName> = \$this->input->LT(1);"
CaptureNextTokenType(d) ::= "$<d.varName> = $this->input->LA(1);"

StructDecl(struct,ctorAttrs,attrs,getters,dispatchMethods,interfaces,extensionMembers) ::= <<
class <struct.name> extends
<if(contextSuperClass)><contextSuperClass><else>ParserRuleContext<endif><if(interfaces)> implements
<interfaces; separator=", "><endif>
{
<PropertiesDecl(struct)>
public function __construct(?ParserRuleContext $parent, ?int $invokingState = null<ctorAttrs:{a | , ?<a> = null}>)
{
parent::__construct($parent, $invokingState);
<if(struct.ctorAttrs)>

<struct.ctorAttrs:{a | $this-><a.name> = $<a.name> ?? $this-><a.name>;}; separator="\n">
<endif>
}

public function getRuleIndex() : int
{
return <parser.name>::RULE_<struct.derivedFromName>;
}
<if(getters)>

<getters:{g | <g>;}; separator="\n\n">
<endif>
<if(struct.provideCopyFrom)> <! don't need copy unless we have subclasses !>
public function copyFrom(ParserRuleContext $context) : void
{

```

```

parent::copyFrom($context);

<struct.attrs:{a | $this-><a.name> = $context-><a.name>;}; separator="\n">
}
<endif>
<if(dispatchMethods)>

<dispatchMethods; separator="\n\n">
<endif>
<if(extensionMembers)>

<extensionMembers; separator="\n\n">
<endif>
}
>>

AltLabelStructDecl(struct,attrs, getters,dispatchMethods) ::= <<
class <struct.name> extends <struct.parentRule; format="cap">Context
{
<PropertiesDecl(struct)>
public function __construct(<struct.parentRule; format="cap">Context $context)
{
parent::__construct($context);

$this->copyFrom($context);
}
<if(getters)>

<getters:{g | <g>;}; separator="\n\n">
<endif>
<if(dispatchMethods)>

<dispatchMethods; separator="\n\n">
<endif>
}
>>

ListenerDispatchMethod(method) ::= <<
public function <if(method.isEnter)>enter<else>exit<endif>Rule(ParseTreeListener $listener) : void
{
if ($listener instanceof <parser.grammarName>Listener) {
$listener-><if(method.isEnter)>enter<else>exit<endif><struct.derivedFromName; format="cap">($this);
}
}
>>

VisitorDispatchMethod(method) ::= <<
public function accept(ParseTreeVisitor $visitor)

```

```

{
if ($visitor instanceof <parser.grammarName>Visitor) {
    return $visitor->visit<struct.derivedFromName; format="cap">($this);
}

return $visitor->visitChildren($this);
}
>>

/** If we don't know location of label def x, use this template */
labelref(x) ::= "<if(!x.isLocal)>$localContext-><endif><x.name>"

/** For any action chunk, what is correctly-typed context struct ptr? */
ctx(actionChunk) ::= "$localContext"

// used for left-recursive rules
recRuleAltPredicate(ruleName,opPrec) ::= "\$this->precpred(\$this->ctx, <opPrec>)"

recRuleSetReturnAction(src,name) ::= "\$<name> = \$<src>-><name>;"

recRuleSetStopToken()      ::= "$this->ctx->stop = $this->input->LT(-1);"

recRuleAltStartAction(ruleName, ctxName, label, isListLabel) ::= <<
$localContext = new Context\\<ctxName>Context($parentContext, $parentState);
<if(label)>
<if(isListLabel)>
$localContext-><label>[] = $previousContext;
<else>
$localContext-><label> = $previousContext;
<endif>
<endif>

$this->pushNewRecursionContext($localContext, $startState, self::RULE_<ruleName>);
>>

recRuleLabeledAltStartAction(ruleName, currentAltLabel, label, isListLabel) ::= <<
$localContext = new Context\\<currentAltLabel; format="cap">Context(new Context\\<ruleName;
format="cap">Context($parentContext, $parentState));
<if(label)>
<if(isListLabel)>
$localContext-><label>[] = $previousContext;
<else>
$localContext-><label> = $previousContext;
<endif>
<endif>

$this->pushNewRecursionContext($localContext, $startState, self::RULE_<ruleName>);
>>

```

```

recRuleReplaceContext(ctxName) ::= <<
$localContext = new Context<<ctxName>Context($localContext);
$this->ctx = $localContext;
$previousContext = $localContext;
>>

```

```

recRuleSetPrevCtx() ::= <<
if ($this->getParseListeners() !== null) {
    $this->triggerExitRuleEvent();
}

```

```

$previousContext = $localContext;
>>

```

```

LexerFile(lexerFile, lexer, namedActions) ::= <<
<fileHeader(lexerFile.grammarFileName, lexerFile.ANTLRVersion)>
<lexer>
>>

```

```

Lexer(lexer, atn, actionFuncs, sempredFuncs, superClass) ::= <<
namespace<if(lexerFile.genPackage)> <lexerFile.genPackage><endif> {
<if(namedActions.header)><namedActions.header><endif>
use Antlr\Antlr4\Runtime\Atn\ATNDeserializer;
use Antlr\Antlr4\Runtime\Atn\LexerATNSimulator;
use Antlr\Antlr4\Runtime\Lexer;
use Antlr\Antlr4\Runtime\CharStream;
use Antlr\Antlr4\Runtime\PredictionContexts\PredictionContextCache;
use Antlr\Antlr4\Runtime\RuleContext;
use Antlr\Antlr4\Runtime\Atn\ATN;
use Antlr\Antlr4\Runtime\Dfa\DFA;
use Antlr\Antlr4\Runtime\Vocabulary;
use Antlr\Antlr4\Runtime\RuntimeMetaData;
use Antlr\Antlr4\Runtime\VocabularyImpl;
<if(namedActions.definitions)><namedActions.definitions><endif>

```

```

final class <lexer.name> extends <superClass; null="Lexer">
{
<if(lexer.tokens)>
public const <lexer.tokens:{k | <k> = <lexer.tokens.(k)>}; separator=" ", wrap, anchor>;
<endif>

<if(lexer.channels)>
public const <lexer.channels:{c | <c> = <lexer.channels.(c)>}; separator=" ", wrap, anchor>;
<endif>

<if(rest(lexer.modes))>

```

```

public const <rest(lexer.modes):{m | <m>=<i>}; separator=", ", wrap, anchor>;
<endif>

/**
 * @var array<string>
 */
public const CHANNEL_NAMES = [
  'DEFAULT_TOKEN_CHANNEL', 'HIDDEN'<if (lexer.channels)>, <lexer.channels:{c | '<c>'}; separator=", ",
wrap, anchor><endif>
];

/**
 * @var array<string>
 */
public const MODE_NAMES = [
  <lexer.modes:{m | '<m>'}; separator=", ", wrap, anchor>
];

/**
 * @var array<string>
 */
public const RULE_NAMES = [
  <lexer.ruleNames:{r | '<r>'}; separator=", ", wrap, anchor>
];

<vocabulary(lexer.literalNames, lexer.symbolicNames)>

<atn>
protected static $atn;
protected static $decisionToDFA;
protected static $sharedContextCache;
<if(namedActions.members)>

  <namedActions.members>
<endif>

public function __construct(CharStream $input)
{
  parent::__construct($input);

  self::initialize();

  $this->interp = new LexerATNSimulator($this, self::$atn, self::$decisionToDFA, self::$sharedContextCache);
}

private static function initialize() : void
{
  if (self::$atn !== null) {

```

```

return;
}

RuntimeMetaData::checkVersion('<lexerFile.ANTLRVersion>', RuntimeMetaData::VERSION);

$atn = (new ATNDeserializer()->deserialize(self::SERIALIZED_ATN);

$decisionToDFA = [];
for ($i = 0, $count = $atn->getNumberOfDecisions(); $i < $count; $i++) {
    $decisionToDFA[] = new DFA($atn->getDecisionState($i), $i);
}

self::$atn = $atn;
self::$decisionToDFA = $decisionToDFA;
self::$sharedContextCache = new PredictionContextCache();
}

public static function vocabulary() : Vocabulary
{
    static $vocabulary;

    return $vocabulary = $vocabulary ?? new VocabularyImpl(self::LITERAL_NAMES, self::SYMBOLIC_NAMES);
}

public function getGrammarFileName() : string
{
    return '<lexer.grammarFileName>';
}

public function getRuleNames() : array
{
    return self::RULE_NAMES;
}

public function getSerializedATN() : string
{
    return self::SERIALIZED_ATN;
}

/**
 * @return array<string>
 */
public function getChannelNames() : array
{
    return self::CHANNEL_NAMES;
}

/**

```

```

* @return array<string>
*/
public function getModeNames() : array
{
    return self::MODE_NAMES;
}

public function getATN() : ATN
{
    return self::$atn;
}

public function getVocabulary() : Vocabulary
{
    return self::vocabulary();
}
<dumpActions(lexer, "", actionFuncs, sempredFuncs)>
}
}
>>

SerializedATN(model) ::= <<
<if(rest(model.segments))>
/**
* @var string
*/
private const SERIALIZED_ATN =
    <model.segments:{segment| "<segment; wrap={ " .<n>" }>"; separator=" .\n">;
<else>
/**
* @var string
*/
private const SERIALIZED_ATN =
    "<model.serialized; wrap={ " .<n>  " }>";
<endif>
>>

/**
* Using a type to init value map, try to init a type; if not in table
* must be an object, default value is `null`.
*/
initValue(typeName) ::= <<
<phpTypeInitMap.(typeName)>
>>

codeFileExtension() ::= ".php"

Found in path(s):

```

* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/tool/templates/codegen/PHP/PHP.stg

No license file was found, but licenses were detected in source scan.

/*

* Copyright (c) 2012-2017 The ANTLR Project. All rights reserved.

* Use of this file is governed by the BSD 3-clause license that

* can be found in the LICENSE.txt file in the project root.

*/

/**

[The "BSD license"]

Copyright (c) 2011 Cay Horstmann

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

Found in path(s):

* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/gui/GraphicsSupport.java

No license file was found, but licenses were detected in source scan.

/*

[The "BSD licence"]

Copyright (c) 2006 Kay Roepke

All rights reserved.

Redistribution and use in source and binary forms, with or without

modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

/*

This file contains the actual layout of the messages emitted by ANTLR.

The text itself is coming out of the languages/*stg files, according to the chosen locale.

This file contains the format that mimicks GCC output.

*/

location(file, line, column) ::= "<file>:<line>:<column>:"

message(id, text) ::= "<text> [error <id>]"

report(location, message, type) ::= "<location> <type>: <message>"

wantsSingleLineMessage() ::= "true"

Found in path(s):

* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/tool/templates/messages/formats/gnu.stg

No license file was found, but licenses were detected in source scan.

/*

* [The "BSD license"]

* Copyright (c) 2016, Mike Lischke

* All rights reserved.

*

* Redistribution and use in source and binary forms, with or without

* modification, are permitted provided that the following conditions

* are met:

```

*
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products
* derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

```

Found in path(s):

```

* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/tool/templates/codegen/Cpp/Files.stg

```

No license file was found, but licenses were detected in source scan.

```

/*
* [The "BSD license"]
* Copyright (c) 2012-2016 Terence Parr
* Copyright (c) 2012-2016 Sam Harwell
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products
* derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,

```

```

* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

```

```

/** Find left-recursive rules */

```

```

tree grammar LeftRecursiveRuleWalker;

```

```

options {
    tokenVocab=ANTLRParser;
    ASTLabelType=GrammarAST;
}

```

```

@header {
    package org.antlr.v4.parse;

```

```

import org.antlr.v4.misc.*;
import org.antlr.v4.tool.*;
import org.antlr.v4.tool.ast.*;
}

```

```

@members {
    private String ruleName;
    private int currentOuterAltNumber; // which outer alt of rule?
    public int numAlts; // how many alts for this rule total?

```

```

    public void setAltAssoc(AltAST altTree, int alt) {}
    public void binaryAlt(AltAST altTree, int alt) {}
    public void prefixAlt(AltAST altTree, int alt) {}
    public void suffixAlt(AltAST altTree, int alt) {}
    public void otherAlt(AltAST altTree, int alt) {}
    public void setReturnValues(GrammarAST t) {}
}

```

```

@rulecatch { }

```

```

// TODO: can get parser errors for not matching pattern; make them go away

```

```

public
rec_rule returns [boolean isLeftRec]
@init
{
    currentOuterAltNumber = 1;
}
: ^( r=RULE id=RULE_REF {ruleName=$id.getText();}
    ruleModifier?

```

```

// (ARG_ACTION)? shouldn't allow args, right?
^(RETURNS a=ARG_ACTION {setReturnValues($a;})?)?
// (^(THROWS .+)? don't allow
  (^(LOCALS ARG_ACTION)? // TODO: copy these to gen'd code
    (^(OPTIONS .*)
      | ^(AT ID ACTION) // TODO: copy
    )*)
ruleBlock { $isLeftRec = $ruleBlock.isLeftRec;}
exceptionGroup
)
;

exceptionGroup
: exceptionHandler* finallyClause?
;

exceptionHandler
: ^(CATCH ARG_ACTION ACTION)
;

finallyClause
: ^(FINALLY ACTION)
;

ruleModifier
: PUBLIC
| PRIVATE
| PROTECTED
;

ruleBlock returns [boolean isLeftRec]
@init{boolean lr=false; this.numAlts = $start.getChildCount();}
: ^( BLOCK
  (
    o=outerAlternative
    {if ($o.isLeftRec) $isLeftRec = true;}
    {currentOuterAltNumber++;}
  )+
)
;

/** An alt is either prefix, suffix, binary, or ternary operation or "other" */
outerAlternative returns [boolean isLeftRec]
: (binary)=>      binary
      {binaryAlt((AltAST)$start, currentOuterAltNumber); $isLeftRec=true;}
| (prefix)=>      prefix
      {prefixAlt((AltAST)$start, currentOuterAltNumber);}
| (suffix)=>      suffix

```

```

        {suffixAlt((AltAST)$start, currentOuterAltNumber); $isLeftRec=true;}
    | nonLeftRecur    {otherAlt((AltAST)$start, currentOuterAltNumber);}
    ;

binary
: ^( ALT elementOptions? recurse element* recurse epsilonElement* )
  {setAltAssoc((AltAST)$ALT,currentOuterAltNumber);}
;

prefix
: ^( ALT elementOptions?
  element+
  recurse epsilonElement*
  )
  {setAltAssoc((AltAST)$ALT,currentOuterAltNumber);}
;

suffix
: ^( ALT elementOptions? recurse element+ )
  {setAltAssoc((AltAST)$ALT,currentOuterAltNumber);}
;

nonLeftRecur
: ^(ALT elementOptions? element+)
;

recurse
: ^(ASSIGN ID recurseNoLabel)
| ^(PLUS_ASSIGN ID recurseNoLabel)
| recurseNoLabel
;

recurseNoLabel : {((CommonTree)input.LT(1)).getText().equals(ruleName)}? RULE_REF;

token returns [GrammarAST t=null]
: ^(ASSIGN ID s=token {$t = $s.t;})
| ^(PLUS_ASSIGN ID s=token {$t = $s.t;})
| b=STRING_LITERAL    {$t = $b;}
| ^(b=STRING_LITERAL elementOptions) {$t = $b;}
| ^(c=TOKEN_REF elementOptions) {$t = $c;}
| c=TOKEN_REF        {$t = $c;}
;

elementOptions
: ^(ELEMENT_OPTIONS elementOption*)
;

elementOption

```

```
: ID
| ^(ASSIGN ID ID)
| ^(ASSIGN ID STRING_LITERAL)
| ^(ASSIGN ID ACTION)
| ^(ASSIGN ID INT)
;
```

element

```
: atom
| ^(NOT element)
| ^(RANGE atom atom)
| ^(ASSIGN ID element)
| ^(PLUS_ASSIGN ID element)
| ^(SET setElement+)
| RULE_REF
| ebnf
| epsilonElement
;
```

epsilonElement

```
: ACTION
| SEMPRED
| EPSILON
| ^(ACTION elementOptions)
| ^(SEMPRED elementOptions)
;
```

setElement

```
: ^(STRING_LITERAL elementOptions)
| ^(TOKEN_REF elementOptions)
| STRING_LITERAL
| TOKEN_REF
;
```

ebnf: block

```
| ^( OPTIONAL block )
| ^( CLOSURE block )
| ^( POSITIVE_CLOSURE block )
;
```

block

```
: ^(BLOCK ACTION? alternative+)
;
```

alternative

```
: ^(ALT elementOptions? element+)
;
```

```
atom
: ^(RULE_REF ARG_ACTION? elementOptions?)
| ^(STRING_LITERAL elementOptions)
| STRING_LITERAL
| ^(TOKEN_REF elementOptions)
| TOKEN_REF
| ^(WILDCARD elementOptions)
| WILDCARD
| ^(DOT ID element)
;
```

Found in path(s):

```
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/parse/LeftRecursiveRuleWalker.g
```

No license file was found, but licenses were detected in source scan.

```
/*
* [The "BSD license"]
* Copyright (c) 2015 Dan McLaughlin, Mike Lischke
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products
* derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/
```

Found in path(s):

```
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/tool/templates/codegen/Cpp/Cpp.stg
```

No license file was found, but licenses were detected in source scan.

```
/*
 * [The "BSD license"]
 * Copyright (c) 2012-2016 Terence Parr
 * Copyright (c) 2012-2016 Sam Harwell
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * 3. The name of the author may not be used to endorse or promote products
 * derived from this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
 * IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
 * IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
 * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
 * DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
 * THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
 * THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */
```

```
atn(startState, states, edges, rankdir, decisionRanks, useBox) ::= <<
digraph ATN {
rankdir=LR;
<decisionRanks; separator="\n">
<states; separator="\n">
<edges; separator="\n">
}
>>
```

```
dfa(name, startState, states, edges, rankdir, decisionRanks, useBox) ::= <<
digraph <name> {
<if(rankdir)>rankdir=<rankdir>;<endif>
<decisionRanks; separator="\n">
<states; separator="\n">
<edges; separator="\n">
}
```

>>

```
decision-rank(states) ::= <<
{rank=same; rankdir=TB; <states:{s | s<s>}; separator="; ">}
>>
```

```
edge(src,target,label,arrowhead,transitionIndex) ::= <<
<src><if(transitionIndex)>p<transitionIndex><endif> -> <target> [fontsize=11, fontname="Courier", arrowsize=.7,
label = "<label>"<if(arrowhead)>, arrowhead = <arrowhead><endif>];
>>
```

```
action-edge(src,target,label,arrowhead,transitionIndex) ::= <<
<src><if(transitionIndex)>p<transitionIndex><endif> -> <target> [fontsize=11, fontname="Courier", arrowsize=.7,
label = "<label>"<if(arrowhead)>, arrowhead = <arrowhead><endif>];
>>
```

```
epsilon-edge(src,label,target,arrowhead,transitionIndex,loopback=false) ::= <<
<src><if(transitionIndex)>p<transitionIndex><endif> -> <target> [fontname="Times-Italic",
label="&epsilon;"<if(loopback)>, style="dashed"<endif>];
>>
```

```
state(state, label, name, transitions) ::= <%
<name>[fontsize=11,
label="
<! rest(transition) tests for decision states: these nodes have a non-empty set of transitions after the first one. !>
<if(rest(transitions))>
{
<! Label on the left side of the record node. !>
<label>
|
<! Named ports in order on right side of record node, no display text. !>
{<transitions:{t|\<p><i0>>}; separator="|">}
<else>
<label>
<endif>
"
<if(rest(transitions))>
, shape=record, fixedsize=false
<else>
, shape=circle, fixedsize=true, width=.55
<endif>
, peripheries=1];
%>
```

```
stopstate(name,label,actionIndex,useBox) ::= <<
<name>[fontsize=11, label="<label>"<if(actionIndex)>,\naction:<actionIndex><endif>",
<if(useBox)>shape=polygon,sides=4,peripheries=2,fixedsize=false<else>shape=doublecircle, fixedsize=true,
width=.6<endif>];
```

>>

Found in path(s):

```
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-  
jar/org/antlr/v4/tool/templates/dot/graphs.stg
```

No license file was found, but licenses were detected in source scan.

```
/*
```

```
[The "BSD licence"]
```

```
Copyright (c) 2006 Kay Roepke
```

```
All rights reserved.
```

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```
*/
```

```
/*
```

```
This file contains the actual layout of the messages emitted by ANTLR.
```

```
This file contains the default format ANTLR uses.
```

```
*/
```

```
location(file, line, column) ::= "<file>:<line>:<column>:"
```

```
message(id, text) ::= "<id> <text>"
```

```
report(location, message, type) ::= "<type><message.id>: <location> <message.text>"
```

```
wantsSingleLineMessage() ::= "false"
```

Found in path(s):

```
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-  
jar/org/antlr/v4/tool/templates/messages/formats/antlr.stg
```

No license file was found, but licenses were detected in source scan.

```
/*
```

```
* [The "BSD license"]
```

```
* Copyright (c) 2012-2016 Terence Parr
```

```
* Copyright (c) 2012-2016 Sam Harwell
```

```
* All rights reserved.
```

```
*
```

```
* Redistribution and use in source and binary forms, with or without
```

```
* modification, are permitted provided that the following conditions
```

```
* are met:
```

```
*
```

```
* 1. Redistributions of source code must retain the above copyright
```

```
* notice, this list of conditions and the following disclaimer.
```

```
* 2. Redistributions in binary form must reproduce the above copyright
```

```
* notice, this list of conditions and the following disclaimer in the
```

```
* documentation and/or other materials provided with the distribution.
```

```
* 3. The name of the author may not be used to endorse or promote products
```

```
* derived from this software without specific prior written permission.
```

```
*
```

```
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
```

```
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
```

```
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
```

```
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
```

```
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
```

```
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
```

```
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
```

```
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
```

```
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
```

```
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

```
*/
```

```
tree grammar ATNBuilder;
```

```
options {
```

```
language = Java;
```

```
tokenVocab = ANTLRParser;
```

```
ASTLabelType = GrammarAST;
```

```
// filter = true;
```

```
}
```

```
// Include the copyright in this source and also the generated source
```

```
@header {
```

```
/*
```

```
[The "BSD license"]
```

```
Copyright (c) 2010 Terence Parr
```

```
All rights reserved.
```

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

```
package org.antlr.v4.parse;
import org.antlr.v4.tool.*;
import org.antlr.v4.tool.ast.*;
import org.antlr.v4.automata.ATNFactory;
}
```

```
@members {
    ATNFactory factory;
    public ATNBuilder(TreeNodeStream input, ATNFactory factory) {
        this(input);
        this.factory = factory;
    }
}
```

```
dummy : block[null] ; // avoid error about no start rule
```

```
ruleBlock[GrammarAST ebnfRoot] returns [ATNFactory.Handle p]
@init {
    List<ATNFactory.Handle> alts = new ArrayList<ATNFactory.Handle>();
    int alt = 1;
    factory.setCurrentOuterAlt(alt);
}
: ^(BLOCK
    (^ (OPTIONS .*)?
    ( a=alternative
        {alts.add($a.p); factory.setCurrentOuterAlt(++alt);}
    )+

```

```

    )
    {$p = factory.block((BlockAST)$BLOCK, ebnfRoot, alts);}
;

block[GrammarAST ebnfRoot] returns [ATNFactory.Handle p]
@init {List<ATNFactory.Handle> alts = new ArrayList<ATNFactory.Handle>();}
: ^(BLOCK (^OPTIONS .*))?(a=alternative {alts.add($a.p);})+
  {$p = factory.block((BlockAST)$BLOCK, ebnfRoot, alts);}
;

alternative returns [ATNFactory.Handle p]
@init {List<ATNFactory.Handle> els = new ArrayList<ATNFactory.Handle>();}
: ^(LEXER_ALT_ACTION a=alternative lexerCommands)
  {$p = factory.lexerAltCommands($a.p,$lexerCommands.p);}
| ^(ALT elementOptions? EPSILON)    {$p = factory.epsilon($EPSILON);}
| ^(ALT elementOptions? (e=element {els.add($e.p);})+) {$p = factory.alt(els);}
;

lexerCommands returns [ATNFactory.Handle p]
@init {List<ATNFactory.Handle> cmds = new ArrayList<ATNFactory.Handle>();}
: (c=lexerCommand {if ($c.cmd != null) cmds.add($c.cmd);})+
  {
  $p = factory.alt(cmds);
  }
;

lexerCommand returns [ATNFactory.Handle cmd]
: ^(LEXER_ACTION_CALL ID lexerCommandExpr)
  {$cmd = factory.lexerCallCommand($ID, $lexerCommandExpr.start);}
| ID
  {$cmd = factory.lexerCommand($ID);}
;

lexerCommandExpr
: ID
| INT
;

element returns [ATNFactory.Handle p]
: labeledElement  {$p = $labeledElement.p;}
| atom           {$p = $atom.p;}
| subrule       {$p = $subrule.p;}
| ACTION        {$p = factory.action((ActionAST)$ACTION);}
| SEMPRED       {$p = factory.sempred((PredAST)$SEMPRED);}
| ^(ACTION .)   {$p = factory.action((ActionAST)$ACTION);}
| ^(SEMPRED .)  {$p = factory.sempred((PredAST)$SEMPRED);}
| ^(NOT b=blockSet[true]) {$p = $b.p;}
| LEXER_CHAR_SET {$p = factory.charSetLiteral($start);}

```

```

;

astOperand returns [ATNFactory.Handle p]
: atom    {$p = $atom.p;}
| ^(NOT blockSet[true]) {$p = $blockSet.p;}
;

labeledElement returns [ATNFactory.Handle p]
: ^(ASSIGN ID element)  {$p = factory.label($element.p);}
| ^(PLUS_ASSIGN ID element) {$p = factory.listLabel($element.p);}
;

subrule returns [ATNFactory.Handle p]
: ^(OPTIONAL block[$start]) {$p = $block.p;}
| ^(CLOSURE block[$start]) {$p = $block.p;}
| ^(POSITIVE_CLOSURE block[$start]) {$p = $block.p;}
| block[null]    {$p = $block.p;}
;

blockSet[boolean invert] returns [ATNFactory.Handle p]
@init {List<GrammarAST> alts = new ArrayList<GrammarAST>();}
: ^(SET (setElement {alts.add($setElement.start);})+) {$p = factory.set($start, alts, $invert);}
;

/** Don't combine with atom otherwise it will build spurious ATN nodes */
setElement
: ^(STRING_LITERAL .)
| ^(TOKEN_REF .)
| STRING_LITERAL
| TOKEN_REF
| ^(RANGE a=STRING_LITERAL b=STRING_LITERAL)
| LEXER_CHAR_SET
;

atom returns [ATNFactory.Handle p]
: range    {$p = $range.p;}
| ^(DOT ID terminal) {$p = $terminal.p;}
| ^(DOT ID ruleref) {$p = $ruleref.p;}
| ^(WILDCARD .) {$p = factory.wildcard($start);}
| WILDCARD {$p = factory.wildcard($start);}
| blockSet[false] {$p = $blockSet.p;}
| terminal    {$p = $terminal.p;}
| ruleref    {$p = $ruleref.p;}
;

ruleref returns [ATNFactory.Handle p]
: ^(RULE_REF ARG_ACTION? ^(ELEMENT_OPTIONS .*)) {$p = factory.ruleRef($RULE_REF);}
| ^(RULE_REF ARG_ACTION?) {$p = factory.ruleRef($RULE_REF);}

```

```

| RULE_REF      {$p = factory.ruleRef($RULE_REF);}
;

range returns [ATNFactory.Handle p]
: ^(RANGE a=STRING_LITERAL b=STRING_LITERAL) {$p = factory.range($a,$b);}
;

terminal returns [ATNFactory.Handle p]
: ^(STRING_LITERAL .) {$p = factory.stringLiteral((TerminalAST)$start);}
| STRING_LITERAL {$p = factory.stringLiteral((TerminalAST)$start);}
| ^(TOKEN_REF ARG_ACTION .) {$p = factory.tokenRef((TerminalAST)$start);}
| ^(TOKEN_REF .) {$p = factory.tokenRef((TerminalAST)$start);}
| TOKEN_REF {$p = factory.tokenRef((TerminalAST)$start);}
;

elementOptions
: ^(ELEMENT_OPTIONS elementOption*)
;

elementOption
: ID
| ^(ASSIGN ID ID)
| ^(ASSIGN ID STRING_LITERAL)
| ^(ASSIGN ID ACTION)
| ^(ASSIGN ID INT)
;

```

Found in path(s):

* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/parse/ATNBuilder.g

No license file was found, but licenses were detected in source scan.

```

/*
 * [The "BSD license"]
 * Copyright (c) 2013 Terence Parr
 * Copyright (c) 2013 Sam Harwell
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * 3. The name of the author may not be used to endorse or promote products
 * derived from this software without specific prior written permission.

```

```
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/
```

```
// args must be <object-model-object>, <fields-resulting-in-STs>
```

```
ParserFile(file, parser, namedActions, contextSuperClass) ::= <<
```

```
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
```

```
<if(file.genPackage)>
```

```
namespace <file.genPackage> {
```

```
<endif>
```

```
<namedActions.header>
```

```
using System;
```

```
using System.IO;
```

```
using System.Text;
```

```
using System.Diagnostics;
```

```
using System.Collections.Generic;
```

```
using Antlr4.Runtime;
```

```
using Antlr4.Runtime.Atn;
```

```
using Antlr4.Runtime.Misc;
```

```
using Antlr4.Runtime.Tree;
```

```
using DFA = Antlr4.Runtime.Dfa.DFA;
```

```
<parser>
```

```
<if(file.genPackage)>
```

```
} // namespace <file.genPackage>
```

```
<endif>
```

```
>>
```

```
ListenerFile(file, header, namedActions) ::= <<
```

```
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
```

```
<if(file.genPackage)>
```

```
namespace <file.genPackage> {
```

```
<endif>
```

```
<header>
```

```
using Antlr4.Runtime.Misc;
```

```
using IParseTreeListener = Antlr4.Runtime.Tree.IParseTreeListener;
```

```
using IToken = Antlr4.Runtime.IToken;
```

```

/// \<summary>
/// This interface defines a complete listener for a parse tree produced by
/// \<see cref="<csIdentifier.(file.parserName)>"/>.
/// \</summary>
[System.CodeDom.Compiler.GeneratedCode("ANTLR", "<file.ANTLRVersion>")]
[System.CLSCompliant(false)]
public interface I<file.grammarName>Listener : IParseTreeListener {
    <file.listenerNames: {Iname |
/// \<summary>
<if(file.listenerLabelRuleNames.(Iname))>
/// Enter a parse tree produced by the \<c><Iname>\</c>
/// labeled alternative in \<see cref="<file.parserName>.<file.listenerLabelRuleNames.(Iname)>"/>.
<else>
/// Enter a parse tree produced by \<see cref="<file.parserName>.<Iname>"/>.
<endif>
/// \</summary>
/// \<param name="context">The parse tree.\</param>
void Enter<Iname; format="cap">([NotNull] <csIdentifier.(file.parserName)>.<Iname; format="cap">Context
context);
/// \<summary>
<if(file.listenerLabelRuleNames.(Iname))>
/// Exit a parse tree produced by the \<c><Iname>\</c>
/// labeled alternative in \<see cref="<file.parserName>.<file.listenerLabelRuleNames.(Iname)>"/>.
<else>
/// Exit a parse tree produced by \<see cref="<file.parserName>.<Iname>"/>.
<endif>
/// \</summary>
/// \<param name="context">The parse tree.\</param>
void Exit<Iname; format="cap">([NotNull] <csIdentifier.(file.parserName)>.<Iname; format="cap">Context
context);}; separator="\n"
}
<if(file.genPackage)>
} // namespace <file.genPackage>
<endif>
>>

```

```

BaseListenerFile(file, header, namedActions) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
<if(file.genPackage)>
namespace <file.genPackage> {
<endif>
<header>

using Antlr4.Runtime.Misc;
using IErrorNode = Antlr4.Runtime.Tree.IErrorNode;
using ITerminalNode = Antlr4.Runtime.Tree.ITerminalNode;
using IToken = Antlr4.Runtime.IToken;
using ParserRuleContext = Antlr4.Runtime.ParserRuleContext;

```

```

/// \<summary>
/// This class provides an empty implementation of \<see cref="I<file.grammarName>Listener"/>,
/// which can be extended to create a listener which only needs to handle a subset
/// of the available methods.
/// \</summary>
[System.CodeDom.Compiler.GeneratedCode("ANTLR", "<file.ANTLRVersion>")]
[System.CLSCompliant(false)]
public partial class <file.grammarName>BaseListener : I<file.grammarName>Listener {
    <file.listenerNames: {Iname |
/// \<summary>
<if(file.listenerLabelRuleNames.(Iname))>
/// Enter a parse tree produced by the \<c><Iname>\</c>
/// labeled alternative in \<see cref="<file.parserName>.<file.listenerLabelRuleNames.(Iname)>"/>.
<else>
/// Enter a parse tree produced by \<see cref="<file.parserName>.<Iname>"/>.
<endif>
/// \<para>The default implementation does nothing.\</para>
/// \</summary>
/// \<param name="context">The parse tree.\</param>
public virtual void Enter<Iname; format="cap">([NotNull] <csIdentifier.(file.parserName)>.<Iname;
format="cap">Context context) { \}
/// \<summary>
<if(file.listenerLabelRuleNames.(Iname))>
/// Exit a parse tree produced by the \<c><Iname>\</c>
/// labeled alternative in \<see cref="<file.parserName>.<file.listenerLabelRuleNames.(Iname)>"/>.
<else>
/// Exit a parse tree produced by \<see cref="<file.parserName>.<Iname>"/>.
<endif>
/// \<para>The default implementation does nothing.\</para>
/// \</summary>
/// \<param name="context">The parse tree.\</param>
public virtual void Exit<Iname; format="cap">([NotNull] <csIdentifier.(file.parserName)>.<Iname;
format="cap">Context context) { \} }; separator="\n"

/// \<inheritdoc>
/// \<remarks>The default implementation does nothing.\</remarks>
public virtual void EnterEveryRule([NotNull] ParserRuleContext context) { }
/// \<inheritdoc>
/// \<remarks>The default implementation does nothing.\</remarks>
public virtual void ExitEveryRule([NotNull] ParserRuleContext context) { }
/// \<inheritdoc>
/// \<remarks>The default implementation does nothing.\</remarks>
public virtual void VisitTerminal([NotNull] ITerminalNode node) { }
/// \<inheritdoc>
/// \<remarks>The default implementation does nothing.\</remarks>
public virtual void VisitErrorNode([NotNull] IErrorNode node) { }
}

```

```

<if(file.genPackage)>
} // namespace <file.genPackage>
<endif>
>>

VisitorFile(file, header, namedActions) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
<if(file.genPackage)>
namespace <file.genPackage> {
<endif>
<header>
using Antlr4.Runtime.Misc;
using Antlr4.Runtime.Tree;
using IToken = Antlr4.Runtime.IToken;

/// \<summary>
/// This interface defines a complete generic visitor for a parse tree produced
/// by \<see cref="<csIdentifier.(file.parserName)>">/>.
/// \</summary>
/// \<typeparam name="Result">The return type of the visit operation.\</typeparam>
[System.CodeDom.Compiler.GeneratedCode("ANTLR", "<file.ANTLRVersion>")]
[System.CLSCompliant(false)]
public interface I<file.grammarName>Visitor<Result> : IParseTreeVisitor<Result> {
    <file.visitorNames>: { Iname |
/// \<summary>
<if(file.visitorLabelRuleNames.(Iname))>
/// Visit a parse tree produced by the \<c><Iname>\</c>
/// labeled alternative in \<see cref="<file.parserName>.<file.visitorLabelRuleNames.(Iname)>">/>.
<else>
/// Visit a parse tree produced by \<see cref="<file.parserName>.<Iname>">/>.
<endif>
/// \</summary>
/// \<param name="context">The parse tree.\</param>
/// \<return>The visitor result.\</return>
Result Visit<Iname; format="cap">([NotNull] <csIdentifier.(file.parserName)>.<Iname; format="cap">Context
context);}; separator="\n"
}
<if(file.genPackage)>
} // namespace <file.genPackage>
<endif>
>>

BaseVisitorFile(file, header, namedActions) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
<if(file.genPackage)>
namespace <file.genPackage> {
<endif>
<header>

```

```

using Antlr4.Runtime.Misc;
using Antlr4.Runtime.Tree;
using IToken = Antlr4.Runtime.IToken;
using ParserRuleContext = Antlr4.Runtime.ParserRuleContext;

/// <summary>
/// This class provides an empty implementation of <see cref="I<file.grammarName>Visitor{Result}" />,
/// which can be extended to create a visitor which only needs to handle a subset
/// of the available methods.
/// </summary>
/// <typeparam name="Result">The return type of the visit operation.</typeparam>
[System.CodeDom.Compiler.GeneratedCode("ANTLR", "<file.ANTLRVersion>")]
[System.CLSCompliant(false)]
public partial class <file.grammarName>BaseVisitor<Result> : AbstractParseTreeVisitor<Result>,
I<file.grammarName>Visitor<Result> {
    <file.visitorNames> { Iname |
    /// <summary>
    <if(file.visitorLabelRuleNames.(Iname))>
    /// Visit a parse tree produced by the <c><Iname></c>
    /// labeled alternative in <see cref="<file.parserName>.<file.visitorLabelRuleNames.(Iname)>" />.
    <else>
    /// Visit a parse tree produced by <see cref="<file.parserName>.<Iname>" />.
    <endif>
    /// <para>
    /// The default implementation returns the result of calling <see
    cref="AbstractParseTreeVisitor{Result}.VisitChildren(IRuleNode)" />
    /// on <paramref name="context" />.
    /// </para>
    /// </summary>
    /// <param name="context">The parse tree.</param>
    /// <return>The visitor result.</return>
    public virtual Result Visit<Iname; format="cap">([NotNull] <csIdentifier.(file.parserName)>.<Iname;
    format="cap">Context context) { return VisitChildren(context); \}; separator="\n">
    }
    <if(file.genPackage)>
    } // namespace <file.genPackage>
    <endif>
    >>

fileHeader(grammarFileName, ANTLRVersion) ::= <<
//-----
// <auto-generated>
// This code was generated by a tool.
// ANTLR Version: <ANTLRVersion>
//
// Changes to this file may cause incorrect behavior and will be lost if
// the code is regenerated.
// </auto-generated>

```

```
//-----

// Generated from <grammarFileName> by ANTLR <ANTLRVersion>

// Unreachable code detected
#pragma warning disable 0162
// The variable '...' is assigned but its value is never used
#pragma warning disable 0219
// Missing XML comment for publicly visible type or member '...'
#pragma warning disable 1591
// Ambiguous reference in cref attribute
#pragma warning disable 419

>>

Parser(parser, funcs, atn, sempredFuncs, superClass) ::= <<
<Parser_(ctor="parser_ctor", ...)>
>>

Parser_(parser, funcs, atn, sempredFuncs, ctor, superClass) ::= <<
[System.CodeDom.Compiler.GeneratedCode("ANTLR", "<file.ANTLRVersion>")]
[System.CLSCompliant(false)]
public partial class <csIdentifier.(parser.name)> : <superClass>; null="Parser"> {
    protected static DFA[] decisionToDFA;
    protected static PredictionContextCache sharedContextCache = new PredictionContextCache();
    <if(parser.tokens)>
    public const int
        <parser.tokens:{k | <k>=<parser.tokens.(k)>}; separator=", ", wrap, anchor>;
    <endif>
    public const int
        <parser.rules:{r | RULE_<r.name> = <r.index>}; separator=", ", wrap, anchor>;
    public static readonly string[] ruleNames = {
        <parser.ruleNames:{r | "<r>"}; separator=", ", wrap, anchor>
    };

    <vocabulary(parser.literalNames, parser.symbolicNames)>

    public override string GrammarFileName { get { return "<parser.grammarFileName>"; } }

    public override string[] RuleNames { get { return ruleNames; } }

    public override string SerializedAtn { get { return new string(_serializedATN); } }

    static <csIdentifier.(parser.name)>() {
        decisionToDFA = new DFA[_ATN.NumberOfDecisions];
        for (int i = 0; i < _ATN.NumberOfDecisions; i++) {
            decisionToDFA[i] = new DFA(_ATN.GetDecisionState(i), i);
        }
    }
}

```

```

}

<namedActions.members>
<parser:(ctor)()>
<funcs; separator="\n">

<if(sempredFuncs)>
public override bool Sempred(RuleContext _localctx, int ruleIndex, int predIndex) {
    switch (ruleIndex) {
        <parser.sempredFuncs.values:{f}
case <f.ruleIndex>: return <f.name>_sempred((<f.ctxType>)_localctx, predIndex);}; separator="\n">
    }
    return true;
}
<sempredFuncs.values; separator="\n">
<endif>

<atn>
}
>>

vocabulary(literalNames, symbolicNames) ::= <<
private static readonly string[] _LiteralNames = {
    <literalNames:{t | <t>}; null="null", separator=", ", wrap, anchor>
};
private static readonly string[] _SymbolicNames = {
    <symbolicNames:{t | <t>}; null="null", separator=", ", wrap, anchor>
};
public static readonly IVocabulary DefaultVocabulary = new Vocabulary(_LiteralNames, _SymbolicNames);

[NotNull]
public override IVocabulary Vocabulary
{
    get
    {
        return DefaultVocabulary;
    }
}
>>

dumpActions(recog, argFuncs, actionFuncs, sempredFuncs) ::= <<
<if(actionFuncs)>
public override void Action(RuleContext _localctx, int ruleIndex, int actionIndex) {
    switch (ruleIndex) {
        <recog.actionFuncs.values:{f}
case <f.ruleIndex> : <f.name>_action(<if(!recog.modes)>(<f.ctxType>)<endif>_localctx, actionIndex); break;};
separator="\n">
    }
}

```

```

}
<actionFuncs.values; separator="\n">
<endif>
<if(sempredFuncs)>
public override bool Sempred(RuleContext _localctx, int ruleIndex, int predIndex) {
    switch (ruleIndex) {
        <recog.sempredFuncs.values: { f |
        case <f.ruleIndex> : return <f.name>_sempred(<if(!recog.modes)><f.ctxType><endif>_localctx, predIndex);};
        separator="\n">
    }
    return true;
}
<sempredFuncs.values; separator="\n">
<endif>
>>

```

```

parser_ctor(parser) ::= <<
public <csIdentifier.(parser.name)>(ITokenStream input) : this(input, Console.Out, Console.Error) { }

public <csIdentifier.(parser.name)>(ITokenStream input, TextWriter output, TextWriter errorOutput)
: base(input, output, errorOutput)
{
    Interpreter = new ParserATNSimulator(this, _ATN, decisionToDFA, sharedContextCache);
}
>>

```

/* This generates a private method since the actionIndex is generated, making an
* overriding implementation impossible to maintain.

```

*/
RuleActionFunction(r, actions) ::= <<
private void <r.name>_action(<r.ctxType> _localctx, int actionIndex) {
    switch (actionIndex) {
        <actions: { index |
        case <index>: <actions.(index)> break; }; separator="\n">
    }
}
>>

```

/* This generates a private method since the predIndex is generated, making an
* overriding implementation impossible to maintain.

```

*/
RuleSempredFunction(r, actions) ::= <<
private bool <r.name>_sempred(<r.ctxType> _localctx, int predIndex) {
    switch (predIndex) {
        <actions: { index |
        case <index>: return <actions.(index)>; }; separator="\n">
    }
    return true;
}

```

```

}
>>

RuleFunction(currentRule,args,code,locals,ruleCtx,altLabelCtxs,namedActions,finallyAction,postamble,exceptions)
::= <<

<if(ruleCtx)>
<ruleCtx>
<endif>
<altLabelCtxs:{l | <altLabelCtxs.(l)>}; separator="\n">

[RuleVersion(<namedActions.version; null="0">)]
<if(currentRule.modifiers)><currentRule.modifiers:{f | <f> }><else>public <endif><currentRule.ctxType>
<csIdentifier.(currentRule.name)><(<args; separator=", ">) {
  <currentRule.ctxType> _localctx = new <currentRule.ctxType>(Context, State<currentRule.args:{a | ,
<csIdentifier.(a.name)>}>);
  EnterRule(_localctx, <currentRule.startState>, RULE_<currentRule.name>);
  <namedActions.init>
  <locals; separator="\n">
  try {
<if(currentRule.hasLookaheadBlock)>
  int _alt;
<endif>
  <code>
  <postamble; separator="\n">
  <namedActions.after>
  }
  <if(exceptions)>
  <exceptions; separator="\n">
  <else>
  catch (RecognitionException re) {
    _localctx.exception = re;
    ErrorHandler.ReportError(this, re);
    ErrorHandler.Recover(this, re);
  }
  <endif>
  finally {
  <finallyAction>
  ExitRule();
  }
  return _localctx;
  }
  >>

LeftFactoredRuleFunction(currentRule,args,code,locals,namedActions,finallyAction,postamble) ::=
<<

<if(currentRule.modifiers)><currentRule.modifiers:{f | <f> }><else>private <endif><currentRule.ctxType>

```

```

<csIdentifier(currentRule.name)><(args; separator=", "> {
  <currentRule.ctxType> _localctx = new <currentRule.ctxType>(Context, State<currentRule.args:{ a | ,
<csIdentifier(a.name)>}>>);
  EnterLeftFactoredRule(_localctx, <currentRule.startState>, RULE_<currentRule.variantOf>);
  <namedActions.init>
  <locals; separator="\n">
  try {
<if(currentRule.hasLookaheadBlock)>
  int _alt;
<endif>
  <code>
  <postamble; separator="\n">
  <namedActions.after>
  }
  catch (RecognitionException re) {
    _localctx.exception = re;
    ErrorHandler.ReportError(this, re);
    ErrorHandler.Recover(this, re);
  }
  finally {
    <finallyAction>
    ExitRule();
  }
  return _localctx;
}
>>

```

```

// This behaves similar to RuleFunction (enterRule is called, and no adjustments
// are made to the parse tree), but since it's still a variant no context class
// needs to be generated.

```

```

LeftUnfactoredRuleFunction(currentRule,args,code,locals,namedActions,finallyAction,postamble) ::=
<<

```

```

<if(currentRule.modifiers)><currentRule.modifiers:{ f | <f> }><else>private <endif><currentRule.ctxType>
<csIdentifier(currentRule.name)><(args; separator=", "> {
  <currentRule.ctxType> _localctx = new <currentRule.ctxType>(Context, State<currentRule.args:{ a | ,
<csIdentifier(a.name)>}>>);
  EnterRule(_localctx, <currentRule.startState>, RULE_<currentRule.variantOf>);
  <namedActions.init>
  <locals; separator="\n">
  try {
<if(currentRule.hasLookaheadBlock)>
  int _alt;
<endif>
  <code>
  <postamble; separator="\n">
  <namedActions.after>
  }

```

```

catch (RecognitionException re) {
    _localctx.exception = re;
    ErrorHandler.ReportError(this, re);
    ErrorHandler.Recover(this, re);
}
finally {
    <finallyAction>
    ExitRule();
}
return _localctx;
}
>>

```

```

LeftRecursiveRuleFunction(currentRule,args,code,locals,ruleCtx,altLabelCtxs,
    namedActions,finallyAction,postamble) ::=
<<

```

```

<ruleCtx>
<altLabelCtxs:{l | <altLabelCtxs.(l)>}; separator="\n">

```

```

[RuleVersion(<namedActions.version; null="0">)]
<if(currentRule.modifiers)><currentRule.modifiers:{f | <f> }><else>public <endif><currentRule.ctxType>
<csIdentifier.(currentRule.name)>(<args; separator=", "> {
    return <csIdentifier.(currentRule.name)>(0<currentRule.args:{a | , <csIdentifier.(a.name)>}>);
}

```

```

private <currentRule.ctxType> <csIdentifier.(currentRule.name)>(int _p<args:{a | , <a>}>) {
    ParserRuleContext _parentctx = Context;
    int _parentState = State;
    <currentRule.ctxType> _localctx = new <currentRule.ctxType>(Context, _parentState<currentRule.args:{a | ,
<csIdentifier.(a.name)>}>);
    <currentRule.ctxType> _prevctx = _localctx;
    int _startState = <currentRule.startState>;
    EnterRecursionRule(_localctx, <currentRule.startState>, RULE_<currentRule.name>, _p);
    <namedActions.init>
    <locals; separator="\n">
    try {
        <if(currentRule.hasLookaheadBlock)>
            int _alt;
        <endif>
        <code>
        <postamble; separator="\n">
        <namedActions.after>
    }
    catch (RecognitionException re) {
        _localctx.exception = re;
        ErrorHandler.ReportError(this, re);
        ErrorHandler.Recover(this, re);
    }
}

```

```

}
finally {
  <finallyAction>
  UnrollRecursionContexts(_parentctx);
}
return _localctx;
}
>>

```

```

CodeBlockForOuterMostAlt(currentOuterMostAltCodeBlock, locals, preamble, ops) ::= <<
<if(currentOuterMostAltCodeBlock.altLabel)>_localctx = new <currentOuterMostAltCodeBlock.altLabel;
format="cap">Context(_localctx);<endif>
EnterOuterAlt(_localctx, <currentOuterMostAltCodeBlock.alt.altNum>);
<CodeBlockForAlt(currentAltCodeBlock=currentOuterMostAltCodeBlock, ...)>
>>

```

```

CodeBlockForAlt(currentAltCodeBlock, locals, preamble, ops) ::= <<
{
  <locals; separator="\n">
  <preamble; separator="\n">
  <ops; separator="\n">
}
>>

```

```

LL1AltBlock(choice, preamble, alts, error) ::= <<
State = <choice.stateNumber>;
ErrorHandler.Sync(this);
<if(choice.label)><labelref(choice.label)> = TokenStream.LT(1);<endif>
<preamble; separator="\n">
switch (TokenStream.LA(1)) {
<choice.altLook,alts:{look,alt| <cases(ttypes=look)>
  <alt>
  break;}; separator="\n">
default:
  <error>
}
>>

```

```

LL1OptionalBlock(choice, alts, error) ::= <<
State = <choice.stateNumber>;
ErrorHandler.Sync(this);
switch (TokenStream.LA(1)) {
<choice.altLook,alts:{look,alt| <cases(ttypes=look)>
  <alt>
  break;}; separator="\n">
default:
  break;
}

```

>>

```
LL1OptionalBlockSingleAlt(choice, expr, alts, preamble, error, followExpr) ::= <<
State = <choice.stateNumber>;
ErrorHandler.Sync(this);
<preamble; separator="\n">
if (<expr>) {
  <alts; separator="\n">
}
<!else if ( !(<followExpr>) ) <error>!>
>>
```

```
LL1StarBlockSingleAlt(choice, loopExpr, alts, preamble, iteration) ::= <<
State = <choice.stateNumber>;
ErrorHandler.Sync(this);
<preamble; separator="\n">
while (<loopExpr>) {
  <alts; separator="\n">
  State = <choice.loopBackStateNumber>;
  ErrorHandler.Sync(this);
  <iteration>
}
>>
```

```
LL1PlusBlockSingleAlt(choice, loopExpr, alts, preamble, iteration) ::= <<
State = <choice.blockStartStateNumber>;<! alt block decision !>
ErrorHandler.Sync(this);
<preamble; separator="\n">
do {
  <alts; separator="\n">
  State = <choice.stateNumber>;<! loopback/exit decision !>
  ErrorHandler.Sync(this);
  <iteration>
} while ( <loopExpr> );
>>
```

// LL(*) stuff

```
AltBlock(choice, preamble, alts, error) ::= <<
State = <choice.stateNumber>;
ErrorHandler.Sync(this);
<if(choice.label)><labelref(choice.label)> = TokenStream.LT(1);<endif>
<preamble; separator="\n">
switch ( Interpreter.AdaptivePredict(TokenStream,<choice.decision>,Context) ) {
<alts:{alt |
case <i>:
  <alt>
break;}; separator="\n">
```

```
}
```

```
>>
```

```
OptionalBlock(choice, alts, error) ::= <<
```

```
State = <choice.stateNumber>;
```

```
ErrorHandler.Sync(this);
```

```
switch ( Interpreter.AdaptivePredict(TokenStream,<choice.decision>,Context) ) {
```

```
<alts:{alt |
```

```
case <i><if(!choice.ast.greedy)>+1<endif>:
```

```
<alt>
```

```
break;}; separator="\n">
```

```
}
```

```
>>
```

```
StarBlock(choice, alts, sync, iteration) ::= <<
```

```
State = <choice.stateNumber>;
```

```
ErrorHandler.Sync(this);
```

```
_alt = Interpreter.AdaptivePredict(TokenStream,<choice.decision>,Context);
```

```
while ( !_alt!=<choice.exitAlt> && !_alt!=global::Antlr4.Runtime.Atn.ATN.INVALID_ALT_NUMBER ) {
```

```
if ( !_alt==1<if(!choice.ast.greedy)>+1<endif> ) {
```

```
<iteration>
```

```
<alts> <! should only be one !>
```

```
}
```

```
State = <choice.loopBackStateNumber>;
```

```
ErrorHandler.Sync(this);
```

```
_alt = Interpreter.AdaptivePredict(TokenStream,<choice.decision>,Context);
```

```
}
```

```
>>
```

```
PlusBlock(choice, alts, error) ::= <<
```

```
State = <choice.blockStartStateNumber>;<! alt block decision !>
```

```
ErrorHandler.Sync(this);
```

```
_alt = 1<if(!choice.ast.greedy)>+1<endif>;
```

```
do {
```

```
switch (_alt) {
```

```
<alts:{alt|
```

```
case <i><if(!choice.ast.greedy)>+1<endif>:
```

```
<alt>
```

```
break;}; separator="\n">
```

```
default:
```

```
<error>
```

```
}
```

```
State = <choice.loopBackStateNumber>;<! loopback/exit decision !>
```

```
ErrorHandler.Sync(this);
```

```
_alt = Interpreter.AdaptivePredict(TokenStream,<choice.decision>,Context);
```

```
} while ( !_alt!=<choice.exitAlt> && !_alt!=global::Antlr4.Runtime.Atn.ATN.INVALID_ALT_NUMBER );
```

```
>>
```

```
Sync(s) ::= "Sync(<s.expecting.name>);"
```

```
ThrowNoViableAlt(t) ::= "throw new NoViableAltException(this);"
```

```
TestSetInline(s) ::= <<
```

```
<s.bitsets: {bits | <if(rest(rest(bits.ttypes)))><bitsetBitfieldComparison(s, bits)><else><bitsetInlineComparison(s,  
bits)><endif>} ; separator=" || ">
```

```
>>
```

```
// Java language spec 15.19 - shift operators mask operands rather than overflow to 0... need range test
```

```
testShiftInRange(shiftAmount) ::= <<
```

```
((<shiftAmount>) & ~0x3f) == 0
```

```
>>
```

```
// produces smaller bytecode only when bits.ttypes contains more than two items
```

```
bitsetBitfieldComparison(s, bits) ::= <%
```

```
(<testShiftInRange({<offsetShift(s.varName, bits.shift)>})> && ((1L \<< <offsetShift(s.varName, bits.shift)>) &  
<bits.ttypes: {ttype | (1L \<< <offsetShift(tokenType.(ttype), bits.shift)>)} ; separator=" | ">) != 0)
```

```
%>
```

```
isZero ::= [
```

```
"0":true,
```

```
default:false
```

```
]
```

```
offsetShift(shiftAmount, offset) ::= <%
```

```
<if(!isZero.(offset))><shiftAmount> - <offset><else><shiftAmount><endif>
```

```
%>
```

```
// produces more efficient bytecode when bits.ttypes contains at most two items
```

```
bitsetInlineComparison(s, bits) ::= <%
```

```
<bits.ttypes: {ttype | <s.varName>==<tokenType.(ttype)>} ; separator=" || ">
```

```
%>
```

```
cases(ttypes) ::= <<
```

```
<ttypes: {t | case <tokenType.(t)>:}; separator="\n">
```

```
>>
```

```
InvokeRule(r, argExprsChunks) ::= <<
```

```
State = <r.stateNumber>; <if(r.labels)><r.labels: {l | <labelref(l)> =
```

```
}><endif><csIdentifier.(r.name)><(<if(r.ast.options.p)><r.ast.options.p><if(argExprsChunks)>,<endif><endif><arg  
ExprsChunks>);
```

```
>>
```

```
MatchToken(m) ::= <<
```

```
State = <m.stateNumber>; <if(m.labels)><m.labels: {l | <labelref(l)> = }><endif>Match(<tokenType.(m.name)>);
```

```
>>
```

```
MatchSet(m, expr, capture) ::= "<CommonSetStuff(m, expr, capture, false)>"
```

```
MatchNotSet(m, expr, capture) ::= "<CommonSetStuff(m, expr, capture, true)>"
```

```
CommonSetStuff(m, expr, capture, invert) ::= <<
```

```
State = <m.stateNumber>;
```

```
<if(m.labels)><m.labels:{1 | <labelref(l)> = }>TokenStream.LT(1);<endif>
```

```
<capture>
```

```
if ( <if(invert)><m.varName> <= 0 || <else>!<endif>(<expr>) ) {
```

```
  <if(m.labels)><m.labels:{1 | <labelref(l)> = }><endif>ErrorHandler.RecoverInline(this);
```

```
}
```

```
else {
```

```
  ErrorHandler.ReportMatch(this);
```

```
  Consume();
```

```
}
```

```
>>
```

```
Wildcard(w) ::= <<
```

```
State = <w.stateNumber>;
```

```
<if(w.labels)><w.labels:{1 | <labelref(l)> = }><endif>MatchWildcard();
```

```
>>
```

```
// ACTION STUFF
```

```
Action(a, foo, chunks) ::= "<chunks>"
```

```
ArgAction(a, chunks) ::= "<chunks>"
```

```
SemPred(p, chunks, failChunks) ::= <<
```

```
State = <p.stateNumber>;
```

```
if (!(<chunks>)) throw new FailedPredicateException(this, <p.predicate><if(failChunks)>,
```

```
<failChunks><elseif(p.msg)>, <p.msg><endif>);
```

```
>>
```

```
ExceptionClause(e, catchArg, catchAction) ::= <<
```

```
catch (<catchArg>) {
```

```
  <catchAction>
```

```
}
```

```
>>
```

```
// lexer actions are not associated with model objects
```

```
LexerSkipCommand() ::= "Skip()";
```

```
LexerMoreCommand() ::= "More()";
```

```
LexerPopModeCommand() ::= "PopMode()";
```

```
LexerTypeCommand(arg, grammar) ::= "_type = <tokenType.(arg)>";
```

```
LexerChannelCommand(arg, grammar) ::= "_channel = <channelName.(arg)>";
```

```

LexerModeCommand(arg, grammar) ::= "_mode = <modeName.(arg)>";
LexerPushModeCommand(arg, grammar) ::= "PushMode(<modeName.(arg)>);"

ActionText(t) ::= "<t.text>"
ActionTemplate(t) ::= "<t.st>"
ArgRef(a) ::= "_localctx.<csIdentifier.(a.name)>"
LocalRef(a) ::= "_localctx.<csIdentifier.(a.name)>"
RetValRef(a) ::= "_localctx.<csIdentifier.(a.name)>"
QRetValRef(a) ::= "<ctx(a)>.<a.dict>.<csIdentifier.(a.name)>"
/** How to translate $tokenLabel */
TokenRef(t) ::= "<ctx(t)>.<csIdentifier.(tokenType.(t.name))>"
LabelRef(t) ::= "<ctx(t)>.<csIdentifier.(t.name)>"
ListLabelRef(t) ::= "<ctx(t)>.<ListLabelName(csIdentifier.(t.name))>"
SetAttr(s,rhsChunks) ::= "<ctx(s)>.<csIdentifier.(s.name)> = <rhsChunks>";

TokenLabelType() ::= "<file.TokenLabelType; null={IToken}>"
InputSymbolType() ::= "<file.InputSymbolType; null={IToken}>"

TokenPropertyRef_text(t) ::= "<ctx(t)>.<tokenType.(t.label)>!=null?<ctx(t)>.<tokenType.(t.label)>.Text:null)"
TokenPropertyRef_type(t) ::= "<ctx(t)>.<tokenType.(t.label)>!=null?<ctx(t)>.<tokenType.(t.label)>.Type:0)"
TokenPropertyRef_line(t) ::= "<ctx(t)>.<tokenType.(t.label)>!=null?<ctx(t)>.<tokenType.(t.label)>.Line:0)"
TokenPropertyRef_pos(t) ::=
"<ctx(t)>.<tokenType.(t.label)>!=null?<ctx(t)>.<tokenType.(t.label)>.CharPositionInLine:0)"
TokenPropertyRef_channel(t) ::=
"<ctx(t)>.<tokenType.(t.label)>!=null?<ctx(t)>.<tokenType.(t.label)>.Channel:0)"
TokenPropertyRef_index(t) ::=
"<ctx(t)>.<tokenType.(t.label)>!=null?<ctx(t)>.<tokenType.(t.label)>.TokenIndex:0)"
TokenPropertyRef_int(t) ::=
"<ctx(t)>.<tokenType.(t.label)>!=null?int.Parse(<ctx(t)>.<tokenType.(t.label)>.Text):0)"

RulePropertyRef_start(r) ::= "<ctx(r)>.<r.label>!=null?(<ctx(r)>.<r.label>.Start):null)"
RulePropertyRef_stop(r) ::= "<ctx(r)>.<r.label>!=null?(<ctx(r)>.<r.label>.Stop):null)"
RulePropertyRef_text(r) ::=
"<ctx(r)>.<r.label>!=null?TokenStream.GetText(<ctx(r)>.<r.label>.Start,<ctx(r)>.<r.label>.Stop):null)"
RulePropertyRef_ctx(r) ::= "<ctx(r)>.<r.label>"
RulePropertyRef_parser(r) ::= "this"

ThisRulePropertyRef_start(r) ::= "_localctx.Start"
ThisRulePropertyRef_stop(r) ::= "_localctx.Stop"
ThisRulePropertyRef_text(r) ::= "TokenStream.GetText(_localctx.Start, TokenStream.LT(-1))"
ThisRulePropertyRef_ctx(r) ::= "_localctx"
ThisRulePropertyRef_parser(r) ::= "this"

NonLocalAttrRef(s) ::= <%((<s.ruleName;
format="cap">Context)GetInvokingContext(<s.ruleIndex>)).<csIdentifier.(s.name)>%>
SetNonLocalAttr(s, rhsChunks) ::=
<%((<s.ruleName; format="cap">Context)GetInvokingContext(<s.ruleIndex>)).<csIdentifier.(s.name)> =
<rhsChunks>%>

```

```

AddToLabelList(a) ::= "<ctx(a.label)>.<a.listName>.Add(<labelref(a.label)>);"

TokenDecl(t) ::= "<TokenLabelType() <csIdentifier.(tokenType.(t.name))>"
TokenTypeDecl(t) ::= "int <csIdentifier.(tokenType.(t.name))>);"
TokenListDecl(t) ::= "IList<IToken> <csIdentifier.(tokenType.(t.name))> = new List<IToken>()"
RuleContextDecl(r) ::= "<r.ctxName> <csIdentifier.(r.name)>"
RuleContextListDecl(rdecl) ::= "IList<<rdecl.ctxName>> <csIdentifier.(rdecl.name)> = new
List<<rdecl.ctxName>>()"

contextGetterCollection(elementType) ::= <%
<elementType>[]
%>

ContextTokenGetterDecl(t) ::=
    "public ITerminalNode <csIdentifier.(tokenType.(t.name))>() { return
    GetToken(<csIdentifier.(parser.name)>.<csIdentifier.(tokenType.(t.name))>, 0); }"
ContextTokenListGetterDecl(t) ::= <<
    public <contextGetterCollection("ITerminalNode")> <csIdentifier.(tokenType.(t.name))>() { return
    GetTokens(<csIdentifier.(parser.name)>.<csIdentifier.(tokenType.(t.name))>); }
    >>
ContextTokenListIndexedGetterDecl(t) ::= <<
    public ITerminalNode <csIdentifier.(tokenType.(t.name))>(int i) {
    return GetToken(<csIdentifier.(parser.name)>.<csIdentifier.(tokenType.(t.name))>, i);
    }
    >>
ContextRuleGetterDecl(r) ::= <<
    public <r.ctxName> <csIdentifier.(r.name)>() {
    return GetRuleContext<<r.ctxName>>();
    }
    >>
ContextRuleListGetterDecl(r) ::= <<
    public <contextGetterCollection({<r.ctxName>})> <csIdentifier.(r.name)>() {
    return GetRuleContexts<<r.ctxName>>();
    }
    >>
ContextRuleListIndexedGetterDecl(r) ::= <<
    public <r.ctxName> <csIdentifier.(r.name)>(int i) {
    return GetRuleContext<<r.ctxName>>(i);
    }
    >>

LexerRuleContext() ::= "RuleContext"

/** The rule context name is the rule followed by a suffix; e.g.,
 * r becomes rContext.
 */
RuleContextNameSuffix() ::= "Context"

```

```
ImplicitTokenLabel(tokenName) ::= "_<tokenType.(tokenName)>"
```

```
ImplicitRuleLabel(ruleName) ::= "_<ruleName>"
```

```
ImplicitSetLabel(id) ::= "_tset<id>"
```

```
ListLabelName(label) ::= "_<label>"
```

```
CaptureNextToken(d) ::= "<d.varName> = TokenStream.LT(1);"
```

```
CaptureNextTokenType(d) ::= "<d.varName> = TokenStream.LA(1);"
```

```
StructDecl(struct,ctorAttrs,attrs, getters,dispatchMethods,interfaces,extensionMembers,  
    superClass={ParserRuleContext}) ::= <<
```

```
public partial class <struct.name> :
```

```
<if(contextSuperClass)><contextSuperClass><else>ParserRuleContext<endif><if(interfaces)>, <interfaces>  
separator=", "><endif> {
```

```
<attrs:{ a | public <a>; }; separator="\n">
```

```
<getters:{ g | <g>; }; separator="\n">
```

```
<if(ctorAttrs)>public <struct.name>(ParserRuleContext parent, int invokingState) : base(parent, invokingState) {  
<endif>
```

```
public <struct.name>(ParserRuleContext parent, int invokingState<ctorAttrs:{ a | , <a>> )  
: base(parent, invokingState)
```

```
{
```

```
<struct.ctorAttrs:{ a | this.<csIdentifier.(a.name)> = <csIdentifier.(a.name)>; }; separator="\n">
```

```
}
```

```
public override int RuleIndex { get { return RULE_<struct.derivedFromName>; } }
```

```
<if(struct.provideCopyFrom)><! don't need copy unless we have subclasses !>
```

```
public <struct.name>() { }
```

```
public virtual void CopyFrom(<struct.name> context) {
```

```
base.CopyFrom(context);
```

```
<struct.attrs:{ a | this.<csIdentifier.(a.name)> = context.<csIdentifier.(a.name)>; }; separator="\n">
```

```
}
```

```
<endif>
```

```
<dispatchMethods; separator="\n">
```

```
<extensionMembers; separator="\n">
```

```
}
```

```
>>
```

```
AltLabelStructDecl(struct,attrs, getters,dispatchMethods) ::= <<
```

```
public partial class <struct.name> : <currentRule.name; format="cap">Context {
```

```
<attrs:{ a | public <a>; }; separator="\n">
```

```
<getters:{ g | <g>; }; separator="\n">
```

```
public <struct.name>(<currentRule.name; format="cap">Context context) { CopyFrom(context); }
```

```
<dispatchMethods; separator="\n">
```

```
}
```

```
>>
```

```
ListenerDispatchMethod(method) ::= <<
```

```
public override void <if(method.isEnter)>Enter<else>Exit<endif>Rule(IParseTreeListener listener) {
```

```
I<parser.grammarName>Listener typedListener = listener as I<parser.grammarName>Listener;
```

```

    if (typedListener != null) typedListener.<if(method.isEnter)>Enter<else>Exit<endif><struct.derivedFromName;
format="cap">(this);
}
>>

```

```

VisitorDispatchMethod(method) ::= <<
public override TResult Accept\<TResult>(IParseTreeVisitor\<TResult> visitor) {
I<parser.grammarName>Visitor\<TResult> typedVisitor = visitor as I<parser.grammarName>Visitor\<TResult>;
if (typedVisitor != null) return typedVisitor.Visit<struct.derivedFromName; format="cap">(this);
else return visitor.VisitChildren(this);
}
>>

```

```

AttributeDecl(d) ::= "<d.type> <csIdentifier.(d.name)><if(d.initValue)> = <d.initValue><endif>"

```

```

/** If we don't know location of label def x, use this template */

```

```

labelref(x) ::= "<if(!x.isLocal)><typedContext(x.ctx)>. <endif><csIdentifier.(x.name)>"

```

```

/** For any action chunk, what is correctly-typed context struct ptr? */

```

```

ctx(actionChunk) ::= "<typedContext(actionChunk.ctx)>"

```

```

// only casts _localctx to the type when the cast isn't redundant (i.e. to a sub-context for a labeled alt)

```

```

typedContext(ctx) ::= "<if(ctx.provideCopyFrom)>((<ctx.name>)_localctx)<else>_localctx<endif>"

```

```

// used for left-recursive rules

```

```

recRuleAltPredicate(ruleName,opPrec) ::= "Precpred(Context, <opPrec>)"

```

```

recRuleSetReturnAction(src,name) ::= "$<name>=<src>.<name>;"

```

```

recRuleSetStopToken() ::= "Context.Stop = TokenStream.LT(-1);"

```

```

recRuleAltStartAction(ruleName, ctxName, label, isListLabel) ::= <<

```

```

_localctx = new <ctxName>Context(_parentctx, _parentState);

```

```

<if(label)>

```

```

<if(isListLabel)>

```

```

_localctx.<label>.Add(_prevctx);

```

```

<else>

```

```

_localctx.<label> = _prevctx;

```

```

<endif>

```

```

<endif>

```

```

PushNewRecursionContext(_localctx, _startState, RULE_<ruleName>);

```

```

>>

```

```

recRuleLabeledAltStartAction(ruleName, currentAltLabel, label, isListLabel) ::= <<

```

```

_localctx = new <currentAltLabel; format="cap">Context(new <ruleName; format="cap">Context(_parentctx,
_parentState));

```

```

<if(label)>

```

```

<if(isListLabel)>

```

```

((<currentAltLabel; format="cap">Context)_localctx).<label>.Add(_prevctx);

```

```

<else>

```

```

((<currentAltLabel; format="cap">Context)_localctx).<label> = _prevctx;
<endif>
<endif>
PushNewRecursionContext(_localctx, _startState, RULE_<ruleName>);
>>

```

```

recRuleReplaceContext(ctxName) ::= <<
_localctx = new <ctxName>Context(_localctx);
Context = _localctx;
_prevctx = _localctx;
>>

```

```

recRuleSetPrevCtx() ::= <<
if ( ParseListeners!=null )
    TriggerExitRuleEvent();
_prevctx = _localctx;
>>

```

```

LexerFile(file, lexer, namedActions) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
<if(file.genPackage)>
namespace <file.genPackage> {
<endif>
<namedActions.header>
using System;
using System.IO;
using System.Text;
using Antlr4.Runtime;
using Antlr4.Runtime.Atn;
using Antlr4.Runtime.Misc;
using DFA = Antlr4.Runtime.Dfa.DFA;

<lexer>
<if(file.genPackage)>
} // namespace <file.genPackage>
<endif>
>>

```

```

Lexer(lexer, atn, actionFuncs, sempredFuncs, superClass) ::= <<
[System.CodeDom.Compiler.GeneratedCode("ANTLR", "<file.ANTLRVersion>")]
[System.CLSCompliant(false)]
public partial class <csIdentifier.(lexer.name)> : <superClass; null="Lexer"> {
    protected static DFA[] decisionToDFA;
    protected static PredictionContextCache sharedContextCache = new PredictionContextCache();
    public const int
        <lexer.tokens:{k | <tokenType.(k)>=<lexer.tokens.(k)>}; separator=",", wrap, anchor>;
    <if(lexer.channels)>

```

```

public const int
<lexer.channels:{k | <csIdentifier.(k)>=<lexer.channels.(k)>}; separator=", ", wrap, anchor>
<endif>
<if(rest(lexer.modes))>
public const int
<rest(lexer.modes):{m | <m>=<i>}; separator=", ", wrap, anchor>
<endif>
public static string[] channelNames = {
"DEFAULT_TOKEN_CHANNEL", "HIDDEN"<if (lexer.channels)>, <lexer.channels:{c| "<c>"}; separator=", ",
wrap, anchor><endif>
};

public static string[] modeNames = {
<lexer.modes:{m| "<m>"}; separator=", ", wrap, anchor>
};

public static readonly string[] ruleNames = {
<lexer.ruleNames:{r | "<r>"}; separator=", ", wrap, anchor>
};

<namedActions.members>

public <csIdentifier.(lexer.name)>(ICharStream input)
: this(input, Console.Out, Console.Error) { }

public <csIdentifier.(lexer.name)>(ICharStream input, TextWriter output, TextWriter errorOutput)
: base(input, output, errorOutput)
{
Interpreter = new LexerATNSimulator(this, _ATN, decisionToDFA, sharedContextCache);
}

<vocabulary(lexer.literalNames, lexer.symbolicNames)>

public override string GrammarFileName { get { return "<lexer.grammarFileName>"; } }

public override string[] RuleNames { get { return ruleNames; } }

public override string[] ChannelNames { get { return channelNames; } }

public override string[] ModeNames { get { return modeNames; } }

public override string SerializedAtn { get { return new string(_serializedATN); } }

static <csIdentifier.(lexer.name)>() {
decisionToDFA = new DFA[_ATN.NumberOfDecisions];
for (int i = 0; i \< _ATN.NumberOfDecisions; i++) {
decisionToDFA[i] = new DFA(_ATN.GetDecisionState(i), i);
}
}

```

```
}  
<dumpActions(lexer, "", actionFuncs, sempredFuncs)>  
<atn>  
}  
>>
```

```
SerializedATN(model) ::= <<  
private static char[] _serializedATN = {  
    <model.serialized; separator=", ", wrap>,  
};  
  
public static readonly ATN _ATN =  
    new ATNDeserializer().Deserialize(_serializedATN);
```

```
>>
```

```
initValue(typeName) ::= <<  
default(<typeName>)  
>>
```

```
codeFileExtension() ::= ".cs"
```

```
modeName ::= [  
    "DEFAULT_MODE" : "DefaultMode",  
    default : key  
]
```

```
channelName ::= [  
    "HIDDEN" : "Hidden",  
    "DEFAULT_TOKEN_CHANNEL" : "DefaultTokenChannel",  
    default : key  
]
```

```
tokenType ::= [  
    "EOF" : "Eof",  
    default : key  
]
```

```
csIdentifier ::= [  
    "abstract" : "@abstract",  
    "as" : "@as",  
    "base" : "@base",  
    "bool" : "@bool",  
    "break" : "@break",  
    "byte" : "@byte",  
    "case" : "@case",
```

"catch" : "@catch",
"char" : "@char",
"checked" : "@checked",
"class" : "@class",
"const" : "@const",
"continue" : "@continue",
"decimal" : "@decimal",
"default" : "@default",
"delegate" : "@delegate",
"do" : "@do",
"double" : "@double",
"else" : "@else",
"enum" : "@enum",
"event" : "@event",
"explicit" : "@explicit",
"extern" : "@extern",
"false" : "@false",
"finally" : "@finally",
"fixed" : "@fixed",
"float" : "@float",
"for" : "@for",
"foreach" : "@foreach",
"goto" : "@goto",
"if" : "@if",
"implicit" : "@implicit",
"in" : "@in",
"int" : "@int",
"interface" : "@interface",
"internal" : "@internal",
"is" : "@is",
"lock" : "@lock",
"long" : "@long",
"namespace" : "@namespace",
"new" : "@new",
"null" : "@null",
"object" : "@object",
"operator" : "@operator",
"out" : "@out",
"override" : "@override",
"params" : "@params",
"private" : "@private",
"protected" : "@protected",
"public" : "@public",
"readonly" : "@readonly",
"ref" : "@ref",
"return" : "@return",
"sbyte" : "@sbyte",
"sealed" : "@sealed",

```
"short" : "@short",
"sizeof" : "@sizeof",
"stackalloc" : "@stackalloc",
"static" : "@static",
"string" : "@string",
"struct" : "@struct",
"switch" : "@switch",
"this" : "@this",
"throw" : "@throw",
>true" : "@true",
"try" : "@try",
"typeof" : "@typeof",
"uint" : "@uint",
"ulong" : "@ulong",
"unchecked" : "@unchecked",
"unsafe" : "@unsafe",
"ushort" : "@ushort",
"using" : "@using",
"virtual" : "@virtual",
"values" : "@values",
"void" : "@void",
"volatile" : "@volatile",
"while" : "@while",
default : key
]
```

Found in path(s):

```
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/tool/templates/codegen/CSharp/CSharp.stg
```

No license file was found, but licenses were detected in source scan.

```
/*
```

```
[The "BSD licence"]
```

```
Copyright (c) 2005-2009 Terence Parr
```

```
All rights reserved.
```

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR

IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

Found in path(s):

* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/parse/ANTLRLexer.java

No license file was found, but licenses were detected in source scan.

/*

[The "BSD license"]

Copyright (c) 2011 Terence Parr

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

Found in path(s):

* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/parse/GrammarTreeVisitor.java

No license file was found, but licenses were detected in source scan.

```
/*
 * [The "BSD license"]
 * Copyright (c) 2012-2016 Terence Parr
 * Copyright (c) 2012-2016 Sam Harwell
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * 3. The name of the author may not be used to endorse or promote products
 * derived from this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
 * IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
 * IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
 * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
 * DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
 * THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
 * THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */
```

```
lexer grammar ActionSplitter;
```

```
options { filter=true; }
```

```
@header {
package org.antlr.v4.parse;
import org.antlr.v4.tool.*;
import org.antlr.v4.tool.ast.*;
}
```

```
@members {
ActionSplitterListener delegate;
```

```
public ActionSplitter(CharStream input, ActionSplitterListener delegate) {
    this(input, new RecognizerSharedState());
    this.delegate = delegate;
```

```

}

/** force filtering (and return tokens). triggers all above actions. */
public List<Token> getActionTokens() {
    List<Token> chunks = new ArrayList<Token>();
    Token t = nextToken();
    while ( t.getType()!=Token.EOF ) {
        chunks.add(t);
        t = nextToken();
    }
    return chunks;
}

private boolean isIDStartChar(int c) {
    return c == '_' || Character.isLetter(c);
}
}

// ignore comments right away

COMMENT
: '/' ( options {greedy=false;} : . )* '*' {delegate.text($text);}
;

LINE_COMMENT
: '/' ~('\n|\r)* '\r'? '\n' {delegate.text($text);}
;

SET_NONLOCAL_ATTR
: '$' x=ID ':' y=ID WS? '=' expr=ATTR_VALUE_EXPR ';'
{
    delegate.setNonLocalAttr($text, $x, $y, $expr);
}
;

NONLOCAL_ATTR
: '$' x=ID ':' y=ID {delegate.nonLocalAttr($text, $x, $y);}
;

QUALIFIED_ATTR
: '$' x=ID '!' y=ID {input.LA(1)!='('}? {delegate.qualifiedAttr($text, $x, $y);}
;

SET_ATTR
: '$' x=ID WS? '=' expr=ATTR_VALUE_EXPR ';'
{
    delegate.setAttr($text, $x, $expr);
}
}

```

```

;

ATTR
: '$' x=ID {delegate.attr($text, $x);}
;

// Anything else is just random text
TEXT
@init {StringBuilder buf = new StringBuilder();}
@after {delegate.text(buf.toString());}
: ( c=~(\\| '$) {buf.append((char)$c);}
| '\\$' {buf.append('$);}
| '\\ c=~('$) {buf.append('\\').append((char)$c);}
| {!isIDStartChar(input.LA(2))}? => '$' {buf.append('$);}
)+
;

fragment
ID : ('a'..'z'|'A'..'Z'|'_') ('a'..'z'|'A'..'Z'|'0'..'9'|'_')*
;

/** Don't allow an = as first char to prevent $x == 3; kind of stuff. */
fragment
ATTR_VALUE_EXPR
: ~'=' (~';')*
;

fragment
WS : ('\\t'|'\\n'|'\\r')+
;

Found in path(s):
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/parse/ActionSplitter.g
No license file was found, but licenses were detected in source scan.

/*
* [The "BSD license"]
* Copyright (c) 2012-2016 Terence Parr
* Copyright (c) 2012-2016 Sam Harwell
* Copyright (c) 2014 Eric Vergnaud
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.

```

```

* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products
* derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

```

```

/** ANTLR tool checks output templates are compatible with tool code generation.
* For now, a simple string match used on x.y of x.y.z scheme.
* Must match Tool.VERSION during load to templates.
*
* REQUIRED.
*/

```

```

pythonTypeInitMap ::= [
  "bool":"False",
  "int":"0",
  "float":"0.0",
  "str": "",
  default:"None" // anything other than a primitive type is an object
]

```

```

// args must be <object-model-object>, <fields-resulting-in-STs>

```

```

ParserFile(file, parser, namedActions, contextSuperClass) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>

```

```

# encoding: utf-8
from antlr4 import *
from io import StringIO
import sys
if sys.version_info[1] > 5:
  from typing import TextIO
else:
  from typing.io import TextIO

```

```

<namedActions.header>

```

```

<parser>

```

>>

```
ListenerFile(file, header, namedActions) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
from antlr4 import *
if __name__ is not None and "." in __name__:
    from .<file.parserName> import <file.parserName>
else:
    from <file.parserName> import <file.parserName>
<header>

# This class defines a complete listener for a parse tree produced by <file.parserName>.
class <file.grammarName>Listener(ParseTreeListener):

    <file.listenerNames: {lname |
# Enter a parse tree produced by <file.parserName>#<lname>.
def enter<lname; format="cap">(self, ctx:<file.parserName>.<lname; format="cap">Context):
    pass

# Exit a parse tree produced by <file.parserName>#<lname>.
def exit<lname; format="cap">(self, ctx:<file.parserName>.<lname; format="cap">Context):
    pass

}; separator="\n">

del <file.parserName>
>>
```

```
VisitorFile(file, header, namedActions) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
from antlr4 import *
if __name__ is not None and "." in __name__:
    from .<file.parserName> import <file.parserName>
else:
    from <file.parserName> import <file.parserName>
<header>

# This class defines a complete generic visitor for a parse tree produced by <file.parserName>.

class <file.grammarName>Visitor(ParseTreeVisitor):

    <file.visitorNames: {lname |
# Visit a parse tree produced by <file.parserName>#<lname>.
def visit<lname; format="cap">(self, ctx:<file.parserName>.<lname; format="cap">Context):
    return self.visitChildren(ctx)
```

```

}; separator="\n">

del <file.parserName>
>>

fileHeader(grammarFileName, ANTLRVersion) ::= <<
# Generated from <grammarFileName> by ANTLR <ANTLRVersion>
>>

Parser(parser, funcs, atn, sempredFuncs, superClass) ::= <<
<Parser_(ctor="parser_ctor", ...)>
>>

Parser_(parser, funcs, atn, sempredFuncs, ctor, superClass) ::= <<
<if(superClass)>
if __name__ is not None and "." in __name__:
    from .<superClass> import <superClass>
else:
    from <superClass> import <superClass>

<endif>
<atn>

class <parser.name> ( <if(superClass)><superClass><else>Parser<endif> ):

    grammarFileName = "<parser.grammarFileName>"

    atn = ATNDeserializer().deserialize(serializedATN())

    decisionsToDFA = [ DFA(ds, i) for i, ds in enumerate(atn.decisionToState) ]

    sharedContextCache = PredictionContextCache()

    literalNames = [ <parser.literalNames:{t | <t>}; null="\\"<INVALID>\"", separator=", ", wrap, anchor> ]

    symbolicNames = [ <parser.symbolicNames:{t | <t>}; null="\\"<INVALID>\"", separator=", ", wrap, anchor> ]

    <parser.rules:{r | RULE_<r.name> = <r.index>}; separator="\n", wrap, anchor>

    ruleNames = [ <parser.ruleNames:{r | "<r>"}; separator=", ", wrap, anchor> ]

    EOF = <TokenLabelType().EOF
    <if(parser.tokens)>
    <parser.tokens:{k | <k>=<parser.tokens.(k)>}; separator="\n", wrap, anchor>
    <endif>

    <parser:(ctor())>

```

```

<namedActions.members>

<funcs; separator="\n">

<if(sempredFuncs)>
  def sempred(self, localctx:RuleContext, ruleIndex:int, predIndex:int):
    if self._predicates == None:
      self._predicates = dict()
<parser.sempredFuncs.values: { f |
  self._predicates[<f.ruleIndex>] = self.<f.name>_sempred }; separator="\n
  " >
  pred = self._predicates.get(ruleIndex, None)
  if pred is None:
    raise Exception("No predicate with index:" + str(ruleIndex))
  else:
    return pred(localctx, predIndex)

  <sempredFuncs.values; separator="\n">
<endif>

>>

dumpActions(recog, argFuncs, actionFuncs, sempredFuncs) ::= <<
<if(actionFuncs)>
def action(self, localctx:RuleContext, ruleIndex:int, actionIndex:int):
  if self._actions is None:
    actions = dict()
<recog.actionFuncs.values: { f |
  actions[<f.ruleIndex>] = self.<f.name>_action }; separator="\n">
  self._actions = actions
  action = self._actions.get(ruleIndex, None)
  if action is not None:
    action(localctx, actionIndex)
  else:
    raise Exception("No registered action for:" + str(ruleIndex))

<actionFuncs.values; separator="\n">

<endif>
<if(sempredFuncs)>
def sempred(self, localctx:RuleContext, ruleIndex:int, predIndex:int):
  if self._predicates is None:
    preds = dict()
<recog.sempredFuncs.values: { f |
  preds[<f.ruleIndex>] = self.<f.name>_sempred }; separator="\n">

```

```

        self._predicates = preds
    pred = self._predicates.get(ruleIndex, None)
    if pred is not None:
        return pred(localctx, predIndex)
    else:
        raise Exception("No registered predicate for:" + str(ruleIndex))

<sempredFuncs.values; separator="\n">
<endif>
>>

parser_ctor(p) ::= <<
def __init__(self, input:TokenStream, output:TextIO = sys.stdout):
    super().__init__(input, output)
    self.checkVersion("<file.ANTLRVersion>")
    self._interp = ParserATNSimulator(self, self.atn, self.decisionsToDFA, self.sharedContextCache)
    self._predicates = None

>>

/* This generates a private method since the actionIndex is generated, making an
* overriding implementation impossible to maintain.
*/
RuleActionFunction(r, actions) ::= <<

def <r.name>_action(self, localctx:<r.ctxType> , actionIndex:int):
<actions:{index|
<if(first(actions))>
    if actionIndex == <index>:
        <actions.(index)>
<elseif(rest(actions))>
    elif actionIndex == <index>:
        <actions.(index)>
<endif> } ; separator="\n">
>>

/* This generates a private method since the predIndex is generated, making an
* overriding implementation impossible to maintain.
*/
RuleSempredFunction(r, actions) ::= <<
def <r.name>_sempred(self, localctx:<r.ctxType>, predIndex:int):
    <actions:{index|
<if(first(actions))>
    if predIndex == <index>:
        return <actions.(index)>
<elseif(rest(actions))>
    elif predIndex == <index>:
        return <actions.(index)>

```

```
<endif> }; separator="\n">
```

```
>>
```

```
RuleFunction(currentRule,args,code,locals,ruleCtx,altLabelCtxs,namedActions,finallyAction,postamble,exceptions) ::= <<
```

```
<ruleCtx>
```

```
<altLabelCtxs:{l | <altLabelCtxs.(l)>}; separator="\n">
```

```
def <currentRule.name>(self<currentRule.args:{a | , <a.name><if(a.type)><a.type><endif>}>):
```

```
  localctx = <parser.name>.<currentRule.ctxType>(self, self._ctx, self.state<currentRule.args:{a | , <a.name>}>)
```

```
  self.enterRule(localctx, <currentRule.startState>, self.RULE_<currentRule.name>)
```

```
  <namedActions.init>
```

```
  <locals; separator="\n">
```

```
  try:
```

```
    <code>
```

```
    <postamble; separator="\n">
```

```
    <namedActions.after>
```

```
  <if(exceptions)>
```

```
  <exceptions; separator="\n">
```

```
  <else>
```

```
  except RecognitionException as re:
```

```
    localctx.exception = re
```

```
    self._errHandler.reportError(self, re)
```

```
    self._errHandler.recover(self, re)
```

```
  <endif>
```

```
  finally:
```

```
    <finallyAction>
```

```
    self.exitRule()
```

```
  return localctx
```

```
>>
```

```
LeftRecursiveRuleFunction(currentRule,args,code,locals,ruleCtx,altLabelCtxs,  
  namedActions,finallyAction,postamble) ::=
```

```
<<
```

```
<ruleCtx>
```

```
<altLabelCtxs:{l | <altLabelCtxs.(l)>}; separator="\n">
```

```
def <currentRule.name>(self, _p:int=0<if(currentRule.args)>, <args:{a | , <a>}><endif>):
```

```
  _parentctx = self._ctx
```

```
  _parentState = self.state
```

```
  localctx = <parser.name>.<currentRule.ctxType>(self, self._ctx, _parentState<args:{a | , <a.name>}>)
```

```
  _prevctx = localctx
```

```

    _startState = <currentRule.startState>
    self.enterRecursionRule(localctx, <currentRule.startState>, self.RULE_<currentRule.name>, _p)
    <namedActions.init>
    <locals; separator="\n">
    try:
        <code>
        <postamble; separator="\n">
        <namedActions.after>
    except RecognitionException as re:
        localctx.exception = re
        self._errHandler.reportError(self, re)
        self._errHandler.recover(self, re)
    finally:
        <finallyAction>
        self.unrollRecursionContexts(_parentctx)
    return localctx

>>

CodeBlockForOuterMostAlt(currentOuterMostAltCodeBlock, locals, preamble, ops) ::= <<
<if(currentOuterMostAltCodeBlock.altLabel)>localctx = <parser.name>.<currentOuterMostAltCodeBlock.altLabel;
format="cap">Context(self, localctx)<endif>
self.enterOuterAlt(localctx, <currentOuterMostAltCodeBlock.alt.altNum>)
<CodeBlockForAlt(currentAltCodeBlock=currentOuterMostAltCodeBlock, ...)>
>>

CodeBlockForAlt(currentAltCodeBlock, locals, preamble, ops) ::= <<
<locals; separator="\n">
<preamble; separator="\n">
<ops; separator="\n">
>>

LL1AltBlock(choice, preamble, alts, error) ::= <<
self.state = <choice.stateNumber>
self._errHandler.sync(self)
<if(choice.label)><labelref(choice.label)> = _input.LT(1)<endif>
<preamble; separator="\n">
token = self._input.LA(1)
<choice.altLook,alts:{look,alt| <cases(ttypes=look)>
    <alt>
    pass}; separator="\n1">
else:
    <error>

>>

LL1OptionalBlock(choice, alts, error) ::= <<
self.state = <choice.stateNumber>

```

```

self._errHandler.sync(self)
token = self._input.LA(1)
<choice.altLook,alts:{look,alt| <cases(ttypes=look)>
  <alt>
    pass }; separator="\n">
else:
  pass
>>

```

```

LL1OptionalBlockSingleAlt(choice, expr, alts, preamble, error, followExpr) ::= <<
self.state = <choice.stateNumber>
self._errHandler.sync(self)
<preamble; separator="\n">
if <expr>:
  <alts; separator="\n">

<!else if ( !(<followExpr> ) <error!>
>>

```

```

LL1StarBlockSingleAlt(choice, loopExpr, alts, preamble, iteration) ::= <<
self.state = <choice.stateNumber>
self._errHandler.sync(self)
<preamble; separator="\n">
while <loopExpr>:
  <alts; separator="\n">
  self.state = <choice.loopBackStateNumber>
  self._errHandler.sync(self)
  <iteration>

>>

```

```

LL1PlusBlockSingleAlt(choice, loopExpr, alts, preamble, iteration) ::= <<
self.state = <choice.blockStartStateNumber> <! alt block decision !>
self._errHandler.sync(self)
<preamble; separator="\n">
while True:
  <alts; separator="\n">
  self.state = <choice.stateNumber> <! loopback/exit decision !>
  self._errHandler.sync(self)
  <iteration>
  if not (<loopExpr>):
    break

>>

```

```

// LL(*) stuff

```

```

AltBlock(choice, preamble, alts, error) ::= <<
self.state = <choice.stateNumber>
self._errHandler.sync(self)
<if(choice.label)><labelref(choice.label)> = _input.LT(1)<endif>
<preamble; separator="\n">
la_ = self._interp.adaptivePredict(self._input,<choice.decision>,self._ctx)
<alts:{ alt |
if la_ == <i>:
    <alt>
    pass
}; separator="\nel">

>>

```

```

OptionalBlock(choice, alts, error) ::= <<
self.state = <choice.stateNumber>
self._errHandler.sync(self)
la_ = self._interp.adaptivePredict(self._input,<choice.decision>,self._ctx)
<alts:{ alt |
if la_ == <i><if(!choice.ast.greedy)>+1<endif>:
    <alt>
}; separator="\nel">

>>

```

```

StarBlock(choice, alts, sync, iteration) ::= <<
self.state = <choice.stateNumber>
self._errHandler.sync(self)
_alt = self._interp.adaptivePredict(self._input,<choice.decision>,self._ctx)
while _alt!=<choice.exitAlt> and _alt!=ATN.INVALID_ALT_NUMBER:
    if _alt==1<if(!choice.ast.greedy)>+1<endif>:
        <iteration>
        <alts> <! should only be one !>
        self.state = <choice.loopBackStateNumber>
        self._errHandler.sync(self)
        _alt = self._interp.adaptivePredict(self._input,<choice.decision>,self._ctx)

>>

```

```

PlusBlock(choice, alts, error) ::= <<
self.state = <choice.blockStartStateNumber> <! alt block decision !>
self._errHandler.sync(self)
_alt = 1<if(!choice.ast.greedy)>+1<endif>
while _alt!=<choice.exitAlt> and _alt!=ATN.INVALID_ALT_NUMBER:
    <alts:{ alt|
if _alt == <i><if(!choice.ast.greedy)>+1<endif>:
    <alt>
}; separator="\nel">

```

```

else:
  <error>
  self.state = <choice.loopBackStateNumber> <! loopback/exit decision !>
  self._errHandler.sync(self)
  _alt = self._interp.adaptivePredict(self._input,<choice.decision>,self._ctx)

>>

Sync(s) ::= "sync(<s.expecting.name>)"

ThrowNoViableAlt(t) ::= "raise NoViableAltException(self)"

TestSetInline(s) ::= <<
<s.bitsets:{bits | <if(rest(rest(bits.ttypes)))><bitsetBitfieldComparison(s, bits)><else><bitsetInlineComparison(s,
bits)><endif>} ; separator=" or ">
>>

// Java language spec 15.19 - shift operators mask operands rather than overflow to 0... need range test
testShiftInRange(shiftAmount) ::= <<
((<shiftAmount>) & ~0x3f) == 0
>>

// produces smaller bytecode only when bits.ttypes contains more than two items
bitsetBitfieldComparison(s, bits) ::= <%
(<testShiftInRange({<offsetShiftVar(s.varName, bits.shift)>})> and ((1 \<< <offsetShiftVar(s.varName,
bits.shift)>) & (<bits.ttypes:{ttype | (1 \<< <offsetShiftType(ttype, bits.shift)>)}); separator=" | ">)) != 0)
%>

isZero ::= [
"0":true,
default:false
]

offsetShiftVar(shiftAmount, offset) ::= <%
<if(!isZero.(offset))><shiftAmount> - <offset><else><shiftAmount><endif>
%>

offsetShiftType(shiftAmount, offset) ::= <%
<if(!isZero.(offset))><parser.name>.<shiftAmount> - <offset><else><parser.name>.<shiftAmount><endif>
%>

// produces more efficient bytecode when bits.ttypes contains at most two items
bitsetInlineComparison(s, bits) ::= <%
<bits.ttypes:{ttype | <s.varName>==<parser.name>.<ttype>} ; separator=" or ">
%>

cases(ttypes) ::= <<
if token in [<ttypes:{t | <parser.name>.<t>} ; separator=", ">]:

```

>>

```
InvokeRule(r, argExprsChunks) ::= <<
self.state = <r.stateNumber>
<if(r.labels)><r.labels:{1 | <labelref(1)> =
}><endif>self.<r.name>( <if(r.ast.options.p)><r.ast.options.p><if(argExprsChunks)>,<endif><endif><argExprsChu
nks>)
>>
```

```
MatchToken(m) ::= <<
self.state = <m.stateNumber>
<if(m.labels)><m.labels:{1 | <labelref(1)> = }><endif>self.match(<parser.name>.<m.name>)
>>
```

```
MatchSet(m, expr, capture) ::= "<CommonSetStuff(m, expr, capture, false)>"
```

```
MatchNotSet(m, expr, capture) ::= "<CommonSetStuff(m, expr, capture, true)>"
```

```
CommonSetStuff(m, expr, capture, invert) ::= <<
self.state = <m.stateNumber>
<if(m.labels)><m.labels:{1 | <labelref(1)> = }>self._input.LT(1)<endif>
<capture>
<if(invert)>if <m.varName> \<= 0 or <expr><else>if not(<expr>)<endif>:
<if(m.labels)><m.labels:{1 | <labelref(1)> = }><else> <endif>self._errHandler.recoverInline(self)
else:
    self._errHandler.reportMatch(self)
    self.consume()
>>
```

```
Wildcard(w) ::= <<
self.state = <w.stateNumber>
<if(w.labels)><w.labels:{1 | <labelref(1)> = }><endif>self.matchWildcard()
>>
```

```
// ACTION STUFF
```

```
Action(a, foo, chunks) ::= "<chunks>"
```

```
ArgAction(a, chunks) ::= "<chunks>"
```

```
SemPred(p, chunks, failChunks) ::= <<
self.state = <p.stateNumber>
if not <chunks>:
    from antlr4.error.Errors import FailedPredicateException
    raise FailedPredicateException(self, <p.predicate><if(failChunks)>, <failChunks><elseif(p.msg)>,
<p.msg><endif>)
>>
```

```

ExceptionClause(e, catchArg, catchAction) ::= <<
catch (<catchArg>) {
  <catchAction>
}
>>

// lexer actions are not associated with model objects

LexerSkipCommand() ::= "skip()"
LexerMoreCommand() ::= "more()"
LexerPopModeCommand() ::= "popMode()"

LexerTypeCommand(arg, grammar) ::= "_type = <arg>"
LexerChannelCommand(arg, grammar) ::= "_channel = <arg>"
LexerModeCommand(arg, grammar) ::= "_mode = <arg>"
LexerPushModeCommand(arg, grammar) ::= "pushMode(<arg>)"

ActionText(t) ::= "<t.text>"
ActionTemplate(t) ::= "<t.st>"
ArgRef(a) ::= "localctx.<a.name>"
LocalRef(a) ::= "localctx.<a.name>"
RetValRef(a) ::= "localctx.<a.name>"
QRetValRef(a) ::= "<ctx(a)>.<a.dict>.<a.name>"
/** How to translate $tokenLabel */
TokenRef(t) ::= "<ctx(t)>.<t.name>"
LabelRef(t) ::= "<ctx(t)>.<t.name>"
ListLabelRef(t) ::= "<ctx(t)>.<ListLabelName(t.name)>"
SetAttr(s,rhsChunks) ::= "<ctx(s)>.<s.name> = <rhsChunks>"

TokenLabelType() ::= "<file.TokenLabelType; null={Token}>"
InputSymbolType() ::= "<file.InputSymbolType; null={Token}>"

TokenPropertyRef_text(t) ::= "(None if <ctx(t)>.<t.label> is None else <ctx(t)>.<t.label>.text)"
TokenPropertyRef_type(t) ::= "(0 if <ctx(t)>.<t.label> is None else <ctx(t)>.<t.label>.type)"
TokenPropertyRef_line(t) ::= "(0 if <ctx(t)>.<t.label> is None else <ctx(t)>.<t.label>.line)"
TokenPropertyRef_pos(t) ::= "(0 if <ctx(t)>.<t.label> is None else <ctx(t)>.<t.label>.column)"
TokenPropertyRef_channel(t) ::= "(0 if (<ctx(t)>.<t.label> is None else <ctx(t)>.<t.label>.channel)"
TokenPropertyRef_index(t) ::= "(0 if <ctx(t)>.<t.label> is None else <ctx(t)>.<t.label>.tokenIndex)"
TokenPropertyRef_int(t) ::= "(0 if <ctx(t)>.<t.label> is None else int(<ctx(t)>.<t.label>.text))"

RulePropertyRef_start(r) ::= "(None if <ctx(r)>.<r.label> is None else <ctx(r)>.<r.label>.start)"
RulePropertyRef_stop(r) ::= "(None if <ctx(r)>.<r.label> is None else <ctx(r)>.<r.label>.stop)"
RulePropertyRef_text(r) ::= "(None if <ctx(r)>.<r.label> is None else
self._input.getText(<ctx(r)>.<r.label>.start,<ctx(r)>.<r.label>.stop))"
RulePropertyRef_ctx(r) ::= "<ctx(r)>.<r.label>"
RulePropertyRef_parser(r) ::= "self"

ThisRulePropertyRef_start(r) ::= "localctx.start"

```

```

ThisRulePropertyRef_stop(r) ::= "localctx.stop"
ThisRulePropertyRef_text(r) ::= "self._input.getText(localctx.start, self._input.LT(-1))"
ThisRulePropertyRef_ctx(r) ::= "localctx"
ThisRulePropertyRef_parser(r) ::= "self"

NonLocalAttrRef(s) ::= "getInvokingContext(<s.ruleIndex>).<s.name>"
SetNonLocalAttr(s, rhsChunks) ::= "getInvokingContext(<s.ruleIndex>).<s.name> = <rhsChunks>"

AddToLabelList(a) ::= "<ctx(a.label)>.<a.listName>.append(<labelref(a.label)>)"

TokenDecl(t) ::= "self.<t.name> = None # <TokenLabelType()>"
TokenTypeDecl(t) ::= "self.<t.name> = 0 # <TokenLabelType()> type"
TokenListDecl(t) ::= "self.<t.name> = list() # of <TokenLabelType()>s"
RuleContextDecl(r) ::= "self.<r.name> = None # <r.ctxName>"
RuleContextListDecl(rdecl) ::= "self.<rdecl.name> = list() # of <rdecl.ctxName>s"

ContextTokenGetterDecl(t) ::= <<
def <t.name>(self):
    return self.getToken(<parser.name>.<t.name>, 0)
>>

// should never be called
ContextTokenListGetterDecl(t) ::= <<
def <t.name>_list(self):
    return self.getTokens(<parser.name>.<t.name>)
>>

ContextTokenListIndexedGetterDecl(t) ::= <<
def <t.name>(self, i:int=None):
    if i is None:
        return self.getTokens(<parser.name>.<t.name>)
    else:
        return self.getToken(<parser.name>.<t.name>, i)
>>

ContextRuleGetterDecl(r) ::= <<
def <r.name>(self):
    return self.getTypedRuleContext(<parser.name>.<r.ctxName>,0)
>>

// should never be called
ContextRuleListGetterDecl(r) ::= <<
def <r.name>_list(self):
    return self.getTypedRuleContexts(<parser.name>.<r.ctxName>)
>>

```

```

ContextRuleListIndexedGetterDecl(r) ::= <<
def <r.name>(self, i:int=None):
  if i is None:
    return self.getTypedRuleContexts(<parser.name>.<r.ctxName>)
  else:
    return self.getTypedRuleContext(<parser.name>.<r.ctxName>,i)

>>

LexerRuleContext() ::= "RuleContext"

/** The rule context name is the rule followed by a suffix; e.g.,
 * r becomes rContext.
 */
RuleContextNameSuffix() ::= "Context"

ImplicitTokenLabel(tokenName) ::= "_<tokenName>"
ImplicitRuleLabel(ruleName) ::= "_<ruleName>"
ImplicitSetLabel(id) ::= "_tset<id>"
ListLabelName(label) ::= "<label>"

CaptureNextToken(d) ::= "<d.varName> = self._input.LT(1)"
CaptureNextTokenType(d) ::= "<d.varName> = self._input.LA(1)"

StructDecl(struct,ctorAttrs,attrs, getters,dispatchMethods,interfaces,extensionMembers) ::= <<
class <struct.name>(<if(contextSuperClass)><contextSuperClass><else>ParserRuleContext<endif>):

  def __init__(self, parser, parent:ParserRuleContext=None, invokingState:int=-1<struct.ctorAttrs:{ a | ,
<a.name><if(a.type)><a.type><endif>=None }>):
    super().__init__(parent, invokingState)
    self.parser = parser
    <attrs:{ a | <a> }; separator="\n">
    <struct.ctorAttrs:{ a | self.<a.name> = <a.name> }; separator="\n">

    <getters:{ g | <g> }; separator="\n\n">

  def getRuleIndex(self):
    return <parser.name>.RULE_<struct.derivedFromName>

<if(struct.provideCopyFrom)> <! don't need copy unless we have subclasses !>
  def copyFrom(self, ctx:ParserRuleContext):
    super().copyFrom(ctx)
    <struct.attrs:{ a | self.<a.name> = ctx.<a.name> }; separator="\n">

<endif>
  <dispatchMethods; separator="\n">
  <extensionMembers; separator="\n">

```

>>

```
AltLabelStructDecl(struct,attrs, getters,dispatchMethods) ::= <<
class <struct.name><(currentRule.name; format="cap">Context):
```

```
    def __init__(self, parser, ctx:ParserRuleContext): # actually a <parser.name>.<currentRule.name;
format="cap">Context
        super().__init__(parser)
        <attrs:{ a | <a>}; separator="\n">
        self.copyFrom(ctx)

    <getters:{ g | <g>}; separator="\n">

    <dispatchMethods; separator="\n">
```

>>

```
ListenerDispatchMethod(method) ::= <<
def <if(method.isEnter)>enter<else>exit<endif>Rule(self, listener:ParseTreeListener):
    if hasattr( listener, "<if(method.isEnter)>enter<else>exit<endif><struct.derivedFromName; format="cap">" ):
        listener.<if(method.isEnter)>enter<else>exit<endif><struct.derivedFromName; format="cap">(self)
```

>>

```
VisitorDispatchMethod(method) ::= <<
def accept(self, visitor:ParseTreeVisitor):
    if hasattr( visitor, "visit<struct.derivedFromName; format="cap">" ):
        return visitor.visit<struct.derivedFromName; format="cap">(self)
    else:
        return visitor.visitChildren(self)
```

>>

```
AttributeDecl(d) ::= "self.<d.name> = <if(d.initValue)><d.initValue><else>None<endif>"
```

```
/** If we don't know location of label def x, use this template */
```

```
labelref(x) ::= "<if(!x.isLocal)>localctx.<endif><x.name>"
```

```
/** For any action chunk, what is correctly-typed context struct ptr? */
```

```
ctx(actionChunk) ::= "localctx"
```

```
// used for left-recursive rules
```

```
recRuleAltPredicate(ruleName,opPrec) ::= "self.precpred(self._ctx, <opPrec>)"
```

```
recRuleSetReturnAction(src,name) ::= "$<name>=$<src>.<name>"
```

```
recRuleSetStopToken() ::= "self._ctx.stop = self._input.LT(-1)"
```

```
recRuleAltStartAction(ruleName, ctxName, label) ::= <<
```

```
localctx = <parser.name>.<ctxName>Context(self, _parentctx, _parentState)
```

```

<if(label)>localctx.<label> = _prevctx<endif>
self.pushNewRecursionContext(localctx, _startState, self.RULE_<ruleName>)
>>

recRuleLabeledAltStartAction(ruleName, currentAltLabel, label, isListLabel) ::= <<
localctx = <parser.name>.<currentAltLabel; format="cap">Context(self, <parser.name>.<ruleName;
format="cap">Context(self, _parentctx, _parentState))
<if(label)>
<if(isListLabel)>
localctx.<label>.append(_prevctx)
<else>
localctx.<label> = _prevctx
<endif>
<endif>
self.pushNewRecursionContext(localctx, _startState, self.RULE_<ruleName>)
>>

recRuleReplaceContext(ctxName) ::= <<
localctx = <parser.name>.<ctxName>Context(self, localctx)
self._ctx = localctx
_prevctx = localctx
>>

recRuleSetPrevCtx() ::= <<
if self._parseListeners is not None:
    self.triggerExitRuleEvent()
_prevctx = localctx
>>

LexerFile(lexerFile, lexer, namedActions) ::= <<
<fileHeader(lexerFile.grammarFileName, lexerFile.ANTLRVersion)>
from antlr4 import *
from io import StringIO
from typing.io import TextIO
import sys

<namedActions.header>

<lexer>
>>

Lexer(lexer, atn, actionFuncs, sempredFuncs, superClass) ::= <<
<if(superClass)>
if __name__ is not None and "." in __name__:
    from .<superClass> import <superClass>
else:
    from <superClass> import <superClass>

```

```

<endif>
<atn>

class <lexer.name>(<if(superClass)><superClass><else>Lexer<endif>):

    atn = ATNDeserializer().deserialize(serializedATN())

    decisionsToDFA = [ DFA(ds, i) for i, ds in enumerate(atn.decisionToState) ]

<if(lexer.channels)>
    <lexer.channels:{c| <c> = <lexer.channels.(c)>}; separator="\n">

<endif>
<if(rest(lexer.modes)>
    <rest(lexer.modes):{m| <m> = <i>}; separator="\n">

<endif>
<lexer.tokens:{k| <k> = <lexer.tokens.(k)>}; separator="\n", wrap, anchor>

    channelNames = [ u"DEFAULT_TOKEN_CHANNEL", u"HIDDEN"<if (lexer.channels)>, <lexer.channels:{c|
u"<c>">}; separator=", ", wrap, anchor><endif> ]

    modeNames = [ <lexer.modes:{m| "<m>">}; separator=", ", wrap, anchor> ]

    literalNames = [ "\<INVALID>",
        <lexer.literalNames:{t| <t>}; separator=", ", wrap, anchor> ]

    symbolicNames = [ "\<INVALID>",
        <lexer.symbolicNames:{t| <t>}; separator=", ", wrap, anchor> ]

    ruleNames = [ <lexer.ruleNames:{r| "<r>">}; separator=", ", wrap, anchor> ]

    grammarFileName = "<lexer.grammarFileName>"

    def __init__(self, input=None, output:TextIO = sys.stdout):
        super().__init__(input, output)
        self.checkVersion("<lexerFile.ANTLRVersion>")
        self._interp = LexerATNSimulator(self, self.atn, self.decisionsToDFA, PredictionContextCache())
        self._actions = None
        self._predicates = None

<namedActions.members>

<dumpActions(lexer, "", actionFuncs, sempredFuncs)>

>>

```

```
SerializedATN(model) ::= <<
<! only one segment, can be inlined !>
```

```
def serializedATN():
    with StringIO() as buf:
        buf.write("<model.serialized; wrap={ }<\n>    buf.write({ }>")
        return buf.getvalue()
```

```
>>
```

```
/** Using a type to init value map, try to init a type; if not in table
 * must be an object, default value is "null".
 */
```

```
initValue(typeName) ::= <<
<pythonTypeInitMap.(typeName)>
>>
```

```
codeFileExtension() ::= ".py"
```

Found in path(s):

```
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/tool/templates/codegen/Python3/Python3.stg
```

No license file was found, but licenses were detected in source scan.

```
* [The "BSD license"]
* All rights reserved.
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products
* derived from this software without specific prior written permission.
```

```
[The "BSD licence"]
```

```
All rights reserved.
```

```
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
```

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

Found in path(s):

* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/parse/ANTLRParser.g
No license file was found, but licenses were detected in source scan.

/*

* [The "BSD license"]

* Copyright (c) 2012-2016 Terence Parr

* Copyright (c) 2012-2016 Sam Harwell

* Copyright (c) 2014 Eric Vergnaud

* All rights reserved.

*

* Redistribution and use in source and binary forms, with or without

* modification, are permitted provided that the following conditions

* are met:

*

* 1. Redistributions of source code must retain the above copyright

* notice, this list of conditions and the following disclaimer.

* 2. Redistributions in binary form must reproduce the above copyright

* notice, this list of conditions and the following disclaimer in the

* documentation and/or other materials provided with the distribution.

* 3. The name of the author may not be used to endorse or promote products

* derived from this software without specific prior written permission.

*

* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR

* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES

* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.

* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,

* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT

* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,

* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY

* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT

* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF

* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

/** ANTLR tool checks output templates are compatible with tool code generation.

* For now, a simple string match used on x.y of x.y.z scheme.

* Must match Tool.VERSION during load to templates.

*

* REQUIRED.

*/

javascriptTypeInitMap ::= [

"bool": "false",

"int": "0",

"float": "0.0",

"str": "",

```

default: "{}" // anything other than a primitive type is an object
]

// args must be <object-model-object>, <fields-resulting-in-STs>

ParserFile(file, parser, namedActions, contextSuperClass) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
var antlr4 = require('antlr4/index');
<if(file.genListener)>
var <file.grammarName>Listener = require('./<file.grammarName>Listener').<file.grammarName>Listener;
<endif>
<if(file.genVisitor)>
var <file.grammarName>Visitor = require('./<file.grammarName>Visitor').<file.grammarName>Visitor;
<endif>

<namedActions.header>
<parser>
>>

ListenerFile(file, header, namedActions) ::= <<
<fileHeader(file.grammarFileName, file.ANTLRVersion)>
var antlr4 = require('antlr4/index');

// This class defines a complete listener for a parse tree produced by <file.parserName>.
function <file.grammarName>Listener() {
  antlr4.tree.ParseTreeListener.call(this);
  return this;
}

<file.grammarName>Listener.prototype = Object.create(antlr4.tree.ParseTreeListener.prototype);
<file.grammarName>Listener.prototype.constructor = <file.grammarName>Listener;

<file.listenerNames:{ lname |
// Enter a parse tree produced by <file.parserName>#<lname>.
<file.grammarName>Listener.prototype.enter<lname; format="cap"> = function(ctx) {
\};

// Exit a parse tree produced by <file.parserName>#<lname>.
<file.grammarName>Listener.prototype.exit<lname; format="cap"> = function(ctx) {
\};

}; separator="\n">

exports.<file.grammarName>Listener = <file.grammarName>Listener;
>>

VisitorFile(file, header, namedActions) ::= <<

```

```

<fileHeader(file.grammarFileName, file.ANTLRVersion)>
var antlr4 = require('antlr4/index');

// This class defines a complete generic visitor for a parse tree produced by <file.parserName>.

function <file.grammarName>Visitor() {
  antlr4.tree.ParseTreeVisitor.call(this);
  return this;
}

<file.grammarName>Visitor.prototype = Object.create(antlr4.tree.ParseTreeVisitor.prototype);
<file.grammarName>Visitor.prototype.constructor = <file.grammarName>Visitor;

<file.visitorNames:{lname |
// Visit a parse tree produced by <file.parserName>#<lname>.
<file.grammarName>Visitor.prototype.visit<lname; format="cap"> = function(ctx) {
  return this.visitChildren(ctx);
}
\};

}; separator="\n">

exports.<file.grammarName>Visitor = <file.grammarName>Visitor;
>>

fileHeader(grammarFileName, ANTLRVersion) ::= <<
// Generated from <grammarFileName; format="java-escape"> by ANTLR <ANTLRVersion>
// jshint ignore: start
>>

Parser(parser, funcs, atn, sempredFuncs, superClass) ::= <<
<if(superClass)>
var <superClass> = require('./<superClass>').<superClass>;
<endif>

var grammarFileName = "<parser.grammarFileName; format="java-escape">";

<atn>

var atn = new antlr4.atn.ATNDeserializer().deserialize(serializedATN);

var decisionsToDFA = atn.decisionToState.map( function(ds, index) { return new antlr4.dfa.DFA(ds, index); });

var sharedContextCache = new antlr4.PredictionContextCache();

var literalNames = [ <parser.literalNames:{t | <t>}; null="null", separator=", ", wrap, anchor > ];

var symbolicNames = [ <parser.symbolicNames:{t | <t>}; null="null", separator=", ", wrap, anchor > ];

```

```

var ruleNames = [ <parser.ruleNames:{r | "<r>"}; separator=", ", wrap, anchor > ];

function <parser.name> (input) {
  <superClass; null="antlr4.Parser">.call(this, input);
  this._interp = new antlr4.atn.ParserATNSimulator(this, atn, decisionsToDFA, sharedContextCache);
  this.ruleNames = ruleNames;
  this.literalNames = literalNames;
  this.symbolicNames = symbolicNames;
  <namedActions.members>
  return this;
}

<parser.name>.prototype = Object.create(<superClass; null="antlr4.Parser">.prototype);
<parser.name>.prototype.constructor = <parser.name>;

Object.defineProperty(<parser.name>.prototype, "atn", {
  get : function() {
    return atn;
  }
});

<parser.name>.EOF = antlr4.Token.EOF;
<if(parser.tokens)>
<parser.tokens:{k | <parser.name>.<k> = <parser.tokens.(k)>;}; separator="\n", wrap, anchor >
<endif>

<parser.rules:{r | <parser.name>.RULE_<r.name> = <r.index>;}; separator="\n", wrap, anchor >

<funcs; separator="\n">

<if(semanticFuncs)>
<parser.name>.prototype.sempred = function(localctx, ruleIndex, predIndex) {
  switch(ruleIndex) {
  <parser.sempredFuncs.values:{f | case <f.ruleIndex>:
  return this.<f.name>_sempred(localctx, predIndex);}; separator="\n">
  default:
    throw "No predicate with index:" + ruleIndex;
  }
};

<semanticFuncs.values; separator="\n">
<endif>

exports.<parser.name> = <parser.name>;

>>

```

```

dumpActions(recog, argFuncs, actionFuncs, sempredFuncs) ::= <<
<if(actionFuncs)>
<lexer.name>.prototype.action = function(localctx, ruleIndex, actionIndex) {
  switch (ruleIndex) {
    <recog.actionFuncs.values:{f}|
  case <f.ruleIndex>:
    this.<f.name>_action(localctx, actionIndex);
    break;}; separator="\n">
  default:
    throw "No registered action for:" + ruleIndex;
  }
};

<actionFuncs.values; separator="\n">
<endif>
<if(sempredFuncs)>
<lexer.name>.prototype.sempred = function(localctx, ruleIndex, predIndex) {
  switch (ruleIndex) {
    <recog.sempredFuncs.values:{f}| case <f.ruleIndex>:
    return this.<f.name>_sempred(localctx, predIndex);}; separator="\n">
  default:
    throw "No registered predicate for:" + ruleIndex;
  }
};

<sempredFuncs.values; separator="\n">
<endif>
>>

```

/* This generates a private method since the actionIndex is generated, making an
* overriding implementation impossible to maintain.
*/

```

RuleActionFunction(r, actions) ::= <<

<lexer.name>.prototype.<r.name>_action = function(localctx , actionIndex) {
  switch (actionIndex) {
    <actions:{index}|
  case <index>:
    <actions.(index)>
    break;}; separator="\n">
  default:
    throw "No registered action for:" + actionIndex;
  }
};

>>

```

/* This generates a private method since the predIndex is generated, making an

```

* overriding implementation impossible to maintain.
*/
RuleSempredFunction(r, actions) ::= <<
<if (r.factory.g.lexer)><lexer.name><else><parser.name><endif>.prototype.<r.name>_sempred = function(localctx,
predIndex) {
switch(predIndex) {
<actions:{index| case <index>:
return <actions.(index)>;}; separator="\n">
default:
throw "No predicate with index:" + predIndex;
}
};
>>

```

```

RuleFunction(currentRule,args,code,locals,ruleCtx,altLabelCtxs,namedActions,finallyAction,postamble,exceptions)
::= <<

```

```

<ruleCtx>

```

```

<altLabelCtxs:{l | <altLabelCtxs.(l)>}; separator="\n">

```

```

<! Define fields of this parser to export the context classes !>

```

```

<parser.name>.<currentRule.ctxType> = <currentRule.ctxType>;

```

```

<parser.name>.prototype.<currentRule.name> = function(<currentRule.args:{a | <a.name>}; separator=", ">) {

```

```

var localctx = new <currentRule.ctxType>(this, this._ctx, this.state<currentRule.args:{a | , <a.name>}>);

```

```

this.enterRule(localctx, <currentRule.startState>, <parser.name>.RULE_<currentRule.name>);

```

```

<namedActions.init>

```

```

<locals; separator="\n">

```

```

try {

```

```

    <code>

```

```

    <postamble; separator="\n">

```

```

    <namedActions.after>

```

```

<if(exceptions)>

```

```

<exceptions; separator="\n">

```

```

<else>

```

```

} catch (re) {

```

```

    if(re instanceof antlr4.error.RecognitionException) {

```

```

        localctx.exception = re;

```

```

        this._errHandler.reportError(this, re);

```

```

        this._errHandler.recover(this, re);

```

```

    } else {

```

```

        throw re;

```

```

    }

```

```

}<endif> finally {

```

```

    <finallyAction>

```

```

        this.exitRule();
    }
    return localctx;
};

>>

LeftRecursiveRuleFunction(currentRule,args,code,locals,ruleCtx,altLabelCtxs,
    namedActions,finallyAction,postamble) ::=
<<

<ruleCtx>
<altLabelCtxs:{l | <altLabelCtxs.(l)>}; separator="\n">

<parser.name>.prototype.<currentRule.name> = function(_p<if(currentRule.args)>, <args:{a | , <a}>><endif>) {
    if(_p===undefined) {
        _p = 0;
    }
    var _parentctx = this._ctx;
    var _parentState = this.state;
    var localctx = new <currentRule.ctxType>(this, this._ctx, _parentState<args:{a | , <a.name}>>);
    var _prevctx = localctx;
    var _startState = <currentRule.startState>;
    this.enterRecursionRule(localctx, <currentRule.startState>, <parser.name>.RULE_<currentRule.name>, _p);
    <namedActions.init>
    <locals; separator="\n">
    try {
        <code>
        <postamble; separator="\n">
        <namedActions.after>
    } catch( error) {
        if(error instanceof antlr4.error.RecognitionException) {
            localctx.exception = error;
            this._errHandler.reportError(this, error);
            this._errHandler.recover(this, error);
        } else {
            throw error;
        }
    } finally {
        <finallyAction>
        this.unrollRecursionContexts(_parentctx)
    }
    return localctx;
};

>>

```

```

CodeBlockForOuterMostAlt(currentOuterMostAltCodeBlock, locals, preamble, ops) ::= <<

```

```

<if(currentOuterMostAltCodeBlock.altLabel)>localctx = new <currentOuterMostAltCodeBlock.altLabel;
format="cap">Context(this, localctx);<endif>
this.enterOuterAlt(localctx, <currentOuterMostAltCodeBlock.alt.altNum>);
<CodeBlockForAlt(currentAltCodeBlock=currentOuterMostAltCodeBlock, ...)>
>>

```

```

CodeBlockForAlt(currentAltCodeBlock, locals, preamble, ops) ::= <<
<locals; separator="\n">
<preamble; separator="\n">
<ops; separator="\n">
>>

```

```

LL1AltBlock(choice, preamble, alts, error) ::= <<
this.state = <choice.stateNumber>;
this._errHandler.sync(this);
<if(choice.label)><labelref(choice.label)> = this._input.LT(1);<endif>
<preamble; separator="\n">
switch(this._input.LA(1)) {
<choice.altLook,alts:{look,alt| <cases(ttypes=look)>
  <alt>
  break;}; separator="\n">
default:
  <error>
}
>>

```

```

LL1OptionalBlock(choice, alts, error) ::= <<
this.state = <choice.stateNumber>;
this._errHandler.sync(this);
switch (this._input.LA(1)) {
<choice.altLook,alts:{look,alt| <cases(ttypes=look)>
  <alt>
  break;}; separator="\n">
default:
  break;
}
>>

```

```

LL1OptionalBlockSingleAlt(choice, expr, alts, preamble, error, followExpr) ::= <<
this.state = <choice.stateNumber>;
this._errHandler.sync(this);
<preamble; separator="\n">
if(<expr>) {
  <alts; separator="\n">
}
<!else if ( !(<followExpr> ) <error!>
>>

```

```

LL1StarBlockSingleAlt(choice, loopExpr, alts, preamble, iteration) ::= <<
this.state = <choice.stateNumber>;
this._errHandler.sync(this);
<preamble; separator="\n">
while(<loopExpr>) {
  <alts; separator="\n">
  this.state = <choice.loopBackStateNumber>;
  this._errHandler.sync(this);
  <iteration>
}
>>

```

```

LL1PlusBlockSingleAlt(choice, loopExpr, alts, preamble, iteration) ::= <<
this.state = <choice.blockStartStateNumber>; <! alt block decision !>
this._errHandler.sync(this);
<preamble; separator="\n">
do {
  <alts; separator="\n">
  this.state = <choice.stateNumber>; <! loopback/exit decision !>
  this._errHandler.sync(this);
  <iteration>
} while(<loopExpr>);
>>

```

```
// LL(*) stuff
```

```

AltBlock(choice, preamble, alts, error) ::= <<
this.state = <choice.stateNumber>;
this._errHandler.sync(this);
<if(choice.label)><labelref(choice.label)> = _input.LT(1)<endif>
<preamble; separator="\n">
var la_ = this._interp.adaptivePredict(this._input,<choice.decision>,this._ctx);
switch(la_) {
<alts:{alt |
case <i>:
  <alt>
  break;
}; separator="\n">
}
>>

```

```

OptionalBlock(choice, alts, error) ::= <<
this.state = <choice.stateNumber>;
this._errHandler.sync(this);
var la_ = this._interp.adaptivePredict(this._input,<choice.decision>,this._ctx);
<alts:{alt |
if(la_===<i><if(!choice.ast.greedy)>+1<endif>) {
  <alt>

```

```
}; separator="\n} else ">
}
>>
```

```
StarBlock(choice, alts, sync, iteration) ::= <<
this.state = <choice.stateNumber>;
this._errHandler.sync(this);
var _alt = this._interp.adaptivePredict(this._input,<choice.decision>,this._ctx)
while(_alt!=<choice.exitAlt> && _alt!=antlr4.atn.ATN.INVALID_ALT_NUMBER) {
  if(_alt===1<if(!choice.ast.greedy)>+1<endif>) {
    <iteration>
    <alts> <! should only be one !>
  }
  this.state = <choice.loopBackStateNumber>;
  this._errHandler.sync(this);
  _alt = this._interp.adaptivePredict(this._input,<choice.decision>,this._ctx);
}
>>
```

```
PlusBlock(choice, alts, error) ::= <<
this.state = <choice.blockStartStateNumber>; <! alt block decision !>
this._errHandler.sync(this);
var _alt = 1<if(!choice.ast.greedy)>+1<endif>;
do {
  switch (_alt) {
    <alts:{alt|
case <i><if(!choice.ast.greedy)>+1<endif>:
    <alt>
    break;}; separator="\n">
    default:
    <error>
  }
  this.state = <choice.loopBackStateNumber>; <! loopback/exit decision !>
  this._errHandler.sync(this);
  _alt = this._interp.adaptivePredict(this._input,<choice.decision>, this._ctx);
} while ( _alt!=<choice.exitAlt> && _alt!=antlr4.atn.ATN.INVALID_ALT_NUMBER );
>>
```

```
Sync(s) ::= "sync(<s.expecting.name>)"
```

```
ThrowNoViableAlt(t) ::= "throw new antlr4.error.NoViableAltException(this);"
```

```
TestSetInline(s) ::= <<
<s.bitsets:{bits | <if(rest(rest(bits.ttypes)))><bitsetBitfieldComparison(s, bits)><else><bitsetInlineComparison(s,
bits)><endif>}; separator=" || ">
>>
```

```

// Javascript language spec - shift operators are 32 bits long max
testShiftInRange(shiftAmount) ::= <<
((<shiftAmount>) & ~0x1f) == 0
>>

// produces smaller bytecode only when bits.ttypes contains more than two items
bitsetBitfieldComparison(s, bits) ::= <%
(<testShiftInRange({<offsetShiftVar(s.varName, bits.shift)>})> && ((1 \<< <offsetShiftVar(s.varName,
bits.shift)>) & (<bits.ttypes:{ttype | (1 \<< <offsetShiftType(ttype, bits.shift)>)}; separator=" | ">)) != 0)
%>

isZero ::= [
"0":true,
default:false
]

offsetShiftVar(shiftAmount, offset) ::= <%
<if(!isZero.(offset))><shiftAmount> - <offset><else><shiftAmount><endif>
%>

offsetShiftType(shiftAmount, offset) ::= <%
<if(!isZero.(offset))><parser.name>.<shiftAmount> - <offset><else><parser.name>.<shiftAmount><endif>
%>

// produces more efficient bytecode when bits.ttypes contains at most two items
bitsetInlineComparison(s, bits) ::= <%
<bits.ttypes:{ttype | <s.varName>===<parser.name>.<ttype>}; separator=" || ">
%>

cases(ttypes) ::= <<
<ttypes:{t | case <parser.name>.<t>:}; separator="\n">
>>

InvokeRule(r, argExprsChunks) ::= <<
this.state = <r.stateNumber>;
<if(r.labels)><r.labels:{1 | <labelref(l)> =
}><endif>this.<r.name><(if(r.ast.options.p)><r.ast.options.p><if(argExprsChunks)>,<endif><endif><argExprsChu
nks>);
>>

MatchToken(m) ::= <<
this.state = <m.stateNumber>;
<if(m.labels)><m.labels:{1 | <labelref(l)> = }><endif>this.match(<parser.name>.<m.name>);
>>

MatchSet(m, expr, capture) ::= "<CommonSetStuff(m, expr, capture, false)>"

MatchNotSet(m, expr, capture) ::= "<CommonSetStuff(m, expr, capture, true)>"

```

```

CommonSetStuff(m, expr, capture, invert) ::= <<
this.state = <m.stateNumber>;
<if(m.labels)><m.labels:{1 | <labelref(l)> = }>this._input.LT(1);<endif>
<capture>
<if(invert)>if(<m.varName>\<=0 || <expr><else>if(!(<expr>))<endif> {
  <if(m.labels)><m.labels:{1 | <labelref(l)> = }><endif>this._errHandler.recoverInline(this);
}
else {
  this._errHandler.reportMatch(this);
  this.consume();
}
>>

```

```

Wildcard(w) ::= <<
this.state = <w.stateNumber>;
<if(w.labels)><w.labels:{1 | <labelref(l)> = }><endif>this.matchWildcard();
>>

```

// ACTION STUFF

```
Action(a, foo, chunks) ::= "<chunks>"
```

```
ArgAction(a, chunks) ::= "<chunks>"
```

```

SemPred(p, chunks, failChunks) ::= <<
this.state = <p.stateNumber>;
if (!( <chunks>)) {
  throw new antlr4.error.FailedPredicateException(this, <p.predicate><if(failChunks)>,
<failChunks><elseif(p.msg)>, <p.msg><endif>);
}
>>

```

```

ExceptionClause(e, catchArg, catchAction) ::= <<
catch (<catchArg>) {
  <catchAction>
}
>>

```

// lexer actions are not associated with model objects

```

LexerSkipCommand() ::= "this.skip()"
LexerMoreCommand() ::= "this.more()"
LexerPopModeCommand() ::= "this.popMode()"
LexerTypeCommand(arg, grammar) ::= "this._type = <arg>"
LexerChannelCommand(arg, grammar) ::= "this._channel = <arg>"
LexerModeCommand(arg, grammar) ::= "this._mode = <arg>"
LexerPushModeCommand(arg, grammar) ::= "this.pushMode(<arg>)"

```

```

ActionText(t) ::= "<t.text>"
ActionTemplate(t) ::= "<t.st>"
ArgRef(a) ::= "localctx.<a.name>"
LocalRef(a) ::= "localctx.<a.name>"
RetValRef(a) ::= "localctx.<a.name>"
QRetValRef(a) ::= "<ctx(a)>.<a.dict>.<a.name>"
/** How to translate $tokenLabel */
TokenRef(t) ::= "<ctx(t)>.<t.name>"
LabelRef(t) ::= "<ctx(t)>.<t.name>"
ListLabelRef(t) ::= "<ctx(t)>.<ListLabelName(t.name)>"
SetAttr(s,rhsChunks) ::= "<ctx(s)>.<s.name> = <rhsChunks>"

TokenLabelType() ::= "<file.TokenLabelType; null={Token}>"
InputSymbolType() ::= "<file.InputSymbolType; null={Token}>"

TokenPropertyRef_text(t) ::= "(<ctx(t)>.<t.label>====null ? null : <ctx(t)>.<t.label>.text)"
TokenPropertyRef_type(t) ::= "(<ctx(t)>.<t.label> === null ? 0 : <ctx(t)>.<t.label>.type)"
TokenPropertyRef_line(t) ::= "(<ctx(t)>.<t.label> === null ? 0 : <ctx(t)>.<t.label>.line)"
TokenPropertyRef_pos(t) ::= "(<ctx(t)>.<t.label> === null ? 0 : <ctx(t)>.<t.label>.column)"
TokenPropertyRef_channel(t) ::= "(<ctx(t)>.<t.label> === null ? 0 : <ctx(t)>.<t.label>.channel)"
TokenPropertyRef_index(t) ::= "(<ctx(t)>.<t.label> === null ? 0 : <ctx(t)>.<t.label>.tokenIndex)"
TokenPropertyRef_int(t) ::= "(<ctx(t)>.<t.label> === null ? 0 : parseInt(<ctx(t)>.<t.label>.text))"

RulePropertyRef_start(r) ::= "(<ctx(r)>.<r.label>====null ? null : <ctx(r)>.<r.label>.start)"
RulePropertyRef_stop(r) ::= "(<ctx(r)>.<r.label>====null ? null : <ctx(r)>.<r.label>.stop)"
RulePropertyRef_text(r) ::= "(<ctx(r)>.<r.label>====null ? null : this._input.getText(new antlr4.Interval(<ctx(r)>.<r.label>.start,<ctx(r)>.<r.label>.stop)))"
RulePropertyRef_ctx(r) ::= "<ctx(r)>.<r.label>"
RulePropertyRef_parser(r) ::= "this"

ThisRulePropertyRef_start(r) ::= "localctx.start"
ThisRulePropertyRef_stop(r) ::= "localctx.stop"
ThisRulePropertyRef_text(r) ::= "this._input.getText(new antlr4.Interval(localctx.start, this._input.LT(-1)))"
ThisRulePropertyRef_ctx(r) ::= "localctx"
ThisRulePropertyRef_parser(r) ::= "this"

NonLocalAttrRef(s) ::= "getInvokingContext(<s.ruleIndex>).<s.name>"
SetNonLocalAttr(s, rhsChunks) ::= "getInvokingContext(<s.ruleIndex>).<s.name> = <rhsChunks>"

AddToLabelList(a) ::= "<ctx(a.label)>.<a.listName>.push(<labelref(a.label)>);"

TokenDecl(t) ::= "this.<t.name> = null; // <TokenLabelType()>"
TokenTypeDecl(t) ::= "var <t.name> = 0; // <TokenLabelType()> type"
TokenListDecl(t) ::= "this.<t.name> = []; // of <TokenLabelType()>s"
RuleContextDecl(r) ::= "this.<r.name> = null; // <r.ctxName>"
RuleContextListDecl(rdecl) ::= "this.<rdecl.name> = []; // of <rdecl.ctxName>s"

```

```

ContextTokenGetterDecl(t) ::= <<
<t.name> = function() {
  return this.getToken(<parser.name>.<t.name>, 0);
};
>>

// should never be called
ContextTokenListGetterDecl(t) ::= <<
def <t.name>_list(self):
  return self.getTokens(<parser.name>.<t.name>)
>>

ContextTokenListIndexedGetterDecl(t) ::= <<
<t.name> = function(i) {
  if(i===undefined) {
    i = null;
  }
  if(i===null) {
    return this.getTokens(<parser.name>.<t.name>);
  } else {
    return this.getToken(<parser.name>.<t.name>, i);
  }
};
>>

ContextRuleGetterDecl(r) ::= <<
<r.name> = function() {
  return this.getTypedRuleContext(<r.ctxName>,0);
};
>>

// should never be called
ContextRuleListGetterDecl(r) ::= <<
def <r.name>_list(self):
  return self.getTypedRuleContexts(<parser.name>.<r.ctxName>)
>>

ContextRuleListIndexedGetterDecl(r) ::= <<
<r.name> = function(i) {
  if(i===undefined) {
    i = null;
  }
  if(i===null) {
    return this.getTypedRuleContexts(<r.ctxName>);
  } else {
    return this.getTypedRuleContext(<r.ctxName>,i);
  }
};
>>

```

```
}  
};  
>>
```

```
LexerRuleContext() ::= "RuleContext"
```

```
/** The rule context name is the rule followed by a suffix; e.g.,  
* r becomes rContext.  
*/
```

```
RuleContextNameSuffix() ::= "Context"
```

```
ImplicitTokenLabel(tokenName) ::= "_<tokenName>"
```

```
ImplicitRuleLabel(ruleName) ::= "_<ruleName>"
```

```
ImplicitSetLabel(id) ::= "_tset<id>"
```

```
ListLabelName(label) ::= "<label>"
```

```
CaptureNextToken(d) ::= "<d.varName> = self._input.LT(1)"
```

```
CaptureNextTokenType(d) ::= "<d.varName> = this._input.LA(1);"
```

```
StructDecl(struct,ctorAttrs,attrs,getters,dispatchMethods,interfaces,extensionMembers) ::= <<
```

```
function <struct.name>(parser, parent, invokingState<struct.ctorAttrs:{a | , <a.name>}>) {
```

```
  if(parent===undefined) {
```

```
    parent = null;
```

```
  }
```

```
  if(invokingState===undefined || invokingState===null) {
```

```
    invokingState = -1;
```

```
  }
```

```
  <if(contextSuperClass)><contextSuperClass><else>antlr4.ParserRuleContext<endif>.call(this, parent,  
  invokingState);
```

```
    this.parser = parser;
```

```
    this.ruleIndex = <parser.name>.RULE_<struct.derivedFromName>;
```

```
    <attrs:{a | <a>} separator="\n">
```

```
    <struct.ctorAttrs:{a | this.<a.name> = <a.name> || null;} separator="\n">
```

```
    return this;
```

```
  }
```

```
<struct.name>.prototype =
```

```
Object.create(<if(contextSuperClass)><contextSuperClass><else>antlr4.ParserRuleContext<endif>.prototype);
```

```
<struct.name>.prototype.constructor = <struct.name>;
```

```
<getters:{g | <struct.name>.prototype.<g>} separator="\n\n">
```

```
<if(struct.provideCopyFrom)> <! don't need copy unless we have subclasses !>
```

```
<struct.name>.prototype.copyFrom = function(ctx) {
```

```
<if(contextSuperClass)><contextSuperClass><else>antlr4.ParserRuleContext<endif>.prototype.copyFrom.call(this,  
ctx);
```

```
  <struct.attrs:{a | this.<a.name> = ctx.<a.name>} separator="\n">
```

```
};
```

```

<endif>
<dispatchMethods; separator="\n">
<extensionMembers; separator="\n">

>>

AltLabelStructDecl(struct,attrs, getters,dispatchMethods) ::= <<
function <struct.name>(parser, ctx) {
  <currentRule.name; format="cap">Context.call(this, parser);
  <attrs:{a | <a>;}; separator="\n">
  <currentRule.name; format="cap">Context.prototype.copyFrom.call(this, ctx);
  return this;
}

<struct.name>.prototype = Object.create(<currentRule.name; format="cap">Context.prototype);
<struct.name>.prototype.constructor = <struct.name>;

<! Define fields of this parser to export this struct/context class !>
<parser.name>.<struct.name> = <struct.name>;

<getters:{g | <struct.name>.prototype.<g>;}; separator="\n\n">
<dispatchMethods; separator="\n">

>>

ListenerDispatchMethod(method) ::= <<
<struct.name>.prototype.<if(method.isEnter)>enter<else>exit<endif>Rule = function(listener) {
  if(listener instanceof <parser.grammarName>Listener ) {
    listener.<if(method.isEnter)>enter<else>exit<endif><struct.derivedFromName; format="cap">(this);
  }
};

>>

VisitorDispatchMethod(method) ::= <<
<struct.name>.prototype.accept = function(visitor) {
  if ( visitor instanceof <parser.grammarName>Visitor ) {
    return visitor.visit<struct.derivedFromName; format="cap">(this);
  } else {
    return visitor.visitChildren(this);
  }
};

>>

AttributeDecl(d) ::= "this.<d.name> = <if(d.initValue)><d.initValue><else>null<endif>"

/** If we don't know location of label def x, use this template */

```

```

labelref(x) ::= "<if(!x.isLocal)>localctx.<endif><x.name>"

/** For any action chunk, what is correctly-typed context struct ptr? */
ctx(actionChunk) ::= "localctx"

// used for left-recursive rules
recRuleAltPredicate(ruleName,opPrec) ::= "this.precpred(this._ctx, <opPrec>)"
recRuleSetReturnAction(src,name) ::= "$<name>=<src>.<name>"
recRuleSetStopToken() ::= "this._ctx.stop = this._input.LT(-1);"

recRuleAltStartAction(ruleName, ctxName, label) ::= <<
localctx = new <ctxName>Context(this, _parentctx, _parentState);
<if(label)>localctx.<label> = _prevctx;<endif>
this.pushNewRecursionContext(localctx, _startState, <parser.name>.RULE_<ruleName>);
>>

recRuleLabeledAltStartAction(ruleName, currentAltLabel, label, isListLabel) ::= <<
localctx = new <currentAltLabel; format="cap">Context(this, new <ruleName; format="cap">Context(this,
_parentctx, _parentState));
<if(label)>
<if(isListLabel)>
localctx.<label>.push(_prevctx);
<else>
localctx.<label> = _prevctx;
<endif>
<endif>
this.pushNewRecursionContext(localctx, _startState, <parser.name>.RULE_<ruleName>);
>>

recRuleReplaceContext(ctxName) ::= <<
localctx = new <ctxName>Context(this, localctx);
this._ctx = localctx;
_prevctx = localctx;
>>

recRuleSetPrevCtx() ::= <<
if(this._parseListeners!=null) {
    this.triggerExitRuleEvent();
}
_prevctx = localctx;
>>

LexerFile(lexerFile, lexer, namedActions) ::= <<
<fileHeader(lexerFile.grammarFileName, lexerFile.ANTLRVersion)>
var antlr4 = require('antlr4/index');

<namedActions.header>

```

```

<lexer>

>>

Lexer(lexer, atn, actionFuncs, sempredFuncs, superClass) ::= <<
<if(superClass)>
var <superClass> = require('./<superClass>').<superClass>;
<endif>

<atn>

var atn = new antlr4.atn.ATNDeserializer().deserialize(serializedATN);

var decisionsToDFA = atn.decisionToState.map( function(ds, index) { return new antlr4.dfa.DFA(ds, index); });

function <lexer.name>(input) {
  <if(superClass)><superClass><else>antlr4.Lexer<endif>.call(this, input);
  this._interp = new antlr4.atn.LexerATNSimulator(this, atn, decisionsToDFA, new
antlr4.PredictionContextCache());
  <namedActions.members>
  return this;
}

<lexer.name>.prototype = Object.create(<if(superClass)><superClass><else>antlr4.Lexer<endif>.prototype);
<lexer.name>.prototype.constructor = <lexer.name>;

Object.defineProperty(<lexer.name>.prototype, "atn", {
  get : function() {
    return atn;
  }
});

<lexer.name>.EOF = antlr4.Token.EOF;
<lexer.tokens>: { k | <lexer.name>.<k> = <lexer.tokens.(k)>; separator="\n", wrap, anchor>

<if(lexer.channels)>
<lexer.channels>: { c | <lexer.name>.<c> = <lexer.channels.(c)>; separator="\n">

<endif>
<if(rest(lexer.modes))>
<rest(lexer.modes)>: { m | <lexer.name>.<m> = <i>; separator="\n">

<endif>
<lexer.name>.prototype.channelNames = [ "DEFAULT_TOKEN_CHANNEL", "HIDDEN"<if (lexer.channels)>,
<lexer.channels>: { c | "<c>" }; separator=", ", wrap, anchor><endif> ];

<lexer.name>.prototype.modeNames = [ <lexer.modes>: { m | "<m>" }; separator=", ", wrap, anchor ]];

```

```

<lexer.name>.prototype.literalNames = [ <lexer.literalNames:{t | <t>}; null="null", separator=", ", wrap, anchor > ];

<lexer.name>.prototype.symbolicNames = [ <lexer.symbolicNames:{t | <t>}; null="null", separator=", ", wrap,
anchor > ];

<lexer.name>.prototype.ruleNames = [ <lexer.ruleNames:{r | "<r>"}; separator=", ", wrap, anchor > ];

<lexer.name>.prototype.grammarFileName = "<lexer.grammarFileName>";

<dumpActions(lexer, "", actionFuncs, sempredFuncs)>

exports.<lexer.name> = <lexer.name>;

>>

SerializedATN(model) ::= <<
<! only one segment, can be inlined !>

var serializedATN = ["<model.serialized; wrap={", <\n> " }>"].join("");

>>

/** Using a type to init value map, try to init a type; if not in table
 * must be an object, default value is "null".
 */
initValue(typeName) ::= <<
<javacriptTypeInitMap.(typeName)>
>>

codeFileExtension() ::= ".js"

Found in path(s):
* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-
jar/org/antlr/v4/tool/templates/codegen/JavaScript/JavaScript.stg
No license file was found, but licenses were detected in source scan.

/*
 * [The "BSD license"]
 * Copyright (c) 2012-2016 Terence Parr
 * Copyright (c) 2012-2016 Sam Harwell
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright

```

```

* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products
* derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

```

```

/** templates used to generate make-compatible dependencies */

```

```

/** Generate "f : x, y, z" dependencies for input
* dependencies and generated files. in and out
* are File objects. For example, you can say
* <f.canonicalPath>
*/
dependencies(grammarFileName,in,out) ::= <<
<if(in)><grammarFileName>: <in; separator=", "><endif>
<out:{f | <f> : <grammarFileName>; separator="\n">
>>

```

Found in path(s):

```

* /opt/cola/permits/1192255793_1629156080.64/0/antlr4-4-8-1-sources-1-jar/org/antlr/v4/tool/templates/depend.stg

```

1.170 jdbi3-bom 3.30.0

1.170.1 Available under license :

Jdbi 1 & 2

- Brian McCallister
- Martin Traverso
- Henning Schmiedehausen
- Jax Law
- maniax, for finding a nasty bug in named parameter substitution
- Robert Sfeir, for batch processing feature requests
- Henry Yandell, for `in` clause expansion feature request
- Patrick Burleson, for showing that `DBIException` should extend `RuntimeException`
- Thomas Risberg, for the Spring `DataAccessException` wrapping code

- Simone Gianni, for finding and solving a nasty `setNull` bug
- Thomas Dudziak

Jdbi 3

- Steven Schlansker
- Matthew Hall
- Artem Prigoda
- Marnick L'Eau

Maven Wrapper Jar

Copyright 2013-2022 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You

institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

JDBI

Copyright (C) 2004 - 2015 Brian McCallister

1.171 micrometer-bom 1.9.4

1.171.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2019 VMware, Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <https://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

```
* See the License for the specific language governing permissions and
* limitations under the License.
*/
/*
* Copyright 2012 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
* https://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
```

Found in path(s):

```
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/util/internal/logging/AbstractInternalLogger.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/util/internal/logging/Slf4JLoggerFactory.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/util/internal/logging/Slf4JLogger.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/util/internal/logging/InternalLogLevel.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/util/internal/logging/JdkLoggerFactory.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/util/internal/logging/InternalLoggerFactory.java
No license file was found, but licenses were detected in source scan.
```

```
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
# distributed under the License is distributed on an "AS IS" BASIS,
```

Found in path(s):

```
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/META-INF/native-
image/io.micrometer/micrometer-core/native-image.properties
No license file was found, but licenses were detected in source scan.
```

```
/*
* Copyright 2019 VMware, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
```

* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <https://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.

*/

/*

* Copyright 2017 The Netty Project

*

* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:

*

* <https://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.

*/

Found in path(s):

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/util/internal/logging/LocationAwareSlf4JLogger.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2022 VMware, Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <https://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.

*/

Found in path(s):

- * /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/okhttp3/package-info.java
- * /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/hystrix/package-info.java
- * /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/config/validate/package-info.java
- * /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/jpa/package-info.java
- * /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/commonspool2/package-info.java
- * /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/lang/package-info.java
- * /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/aop/package-info.java
- * /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/package-info.java
- * /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/tomcat/package-info.java
- * /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/jvm/package-info.java
- * /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/logging/package-info.java
- * /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/kafka/package-info.java
- * /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/annotation/package-info.java
- * /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/system/package-info.java
- * /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/jersey/server/package-info.java
- * /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/ipc/http/package-info.java
- * /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/httpcomponents/package-info.java
- * /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/db/package-info.java
- * /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/jetty/package-info.java
- * /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/mongodb/package-info.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2019 VMware, Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at
*
* <https://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.

*/

/*

* Copyright 2013 The Netty Project

*

* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:

*

* <https://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.

*/

Found in path(s):

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/util/internal/logging/package-info.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2017 VMware, Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <https://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/internal/DefaultLongTaskTimer.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/util/StringEscapeUtils.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/composite/CompositeMeterRegistry.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/distribution/HistogramSnapshot.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/composite/CompositeFunctionCounter.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/db/DatabaseTableMetrics.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/jvm/ClassLoaderMetrics.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/aop/TimedAspect.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/distribution/ValueAtPercentile.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/noop/NoopGauge.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/FunctionCounter.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/config/MeterRegistryConfig.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/search/RequiredSearch.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/noop/NoopDistributionSummary.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/distribution/CountAtBucket.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/cache/GuavaCacheMetrics.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/annotation/Incubating.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/dropwizard/DropwizardClock.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/step/StepRegistryConfig.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/system/FileDescriptorMetrics.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/dropwizard/package-info.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/jersey/server/MetricsApplicationEventListener.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/distribution/HistogramGauges.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/jersey/server/JerseyTagsProvider.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/annotation/Counted.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/jersey/server/DefaultJerseyTagsProvider.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/Tags.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/dropwizard/DropwizardFunctionCounter.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/internal/DefaultGauge.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/cumulative/CumulativeDistributionSummary.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/jvm/ExecutorServiceMetrics.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/cache/package-info.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/step/StepDouble.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/internal/Mergeable.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/internal/TimedExecutor.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/distribution/TimeWindowFixedBoundaryHistogram.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/Counter.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/annotation/Timed.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/MockClock.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/noop/NoopTimer.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/jpa/HibernateMetrics.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/composite/package-info.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/package-info.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/cumulative/CumulativeTimer.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/system/ProcessorMetrics.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/aop/CountedAspect.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/distribution/NoopHistogram.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/Gauge.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/tomcat/TomcatMetrics.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/hystrix/MicrometerMetricsPublisherCommand.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/noop/NoopCounter.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/distribution/pause/ClockDriftPauseDetector.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/step/StepTimer.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/Clock.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/step/StepFunctionCounter.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/noop/NoopFunctionCounter.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/config/MissingRequiredConfigurationException.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/annotation/TimedSet.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/composite/CompositeMeter.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/distribution/DistributionStatisticConfig.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/dropwizard/DropwizardDistributionSummary.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/simple/package-info.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/cumulative/CumulativeFunctionTimer.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/composite/CompositeLongTaskTimer.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/Metrics.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/search/Search.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/step/package-info.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/cache/CaffeineCacheMetrics.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/cumulative/package-info.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/composite/CompositeCustomMeter.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/util/MeterEquivalence.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/cache/EhCache2Metrics.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/cache/HazelcastCacheMetrics.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/internal/DefaultMeter.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/step/StepDistributionSummary.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/AbstractMeter.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/cache/CacheMeterBinder.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/distribution/pause/package-info.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/db/PostgreSQLDatabaseMetrics.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/dropwizard/DropwizardFunctionTimer.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/config/NamingConvention.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/composite/CompositeTimeGauge.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/step/StepMeasurement.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/AbstractDistributionSummary.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/dropwizard/DropwizardRate.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/util/package-info.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/internal/TimedExecutorService.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/jersey/server/TimedFinder.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/config/package-info.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/okhttp3/OkHttpMetricsEventListener.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/step/StepCounter.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/dropwizard/DropwizardTimer.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/noop/NoopLongTaskTimer.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/search/MeterNotFoundException.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/simple/SimpleConfig.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/simple/SimpleMeterRegistry.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/jvm/DiskSpaceMetrics.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/step/StepMeterRegistry.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/config/MeterFilterReply.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/distribution/PercentileHistogramBuckets.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/Measurement.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/Statistic.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/TimeGauge.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/jersey/server/AnnotationFinder.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/system/UptimeMetrics.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/push/package-info.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/cumulative/CumulativeFunctionCounter.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/composite/CompositeCounter.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/hystrix/HystrixMetricsBinder.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/logging/Log4j2Metrics.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/system/DiskSpaceMetrics.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/distribution/AbstractTimeWindowHistogram.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/distribution/pause/NoPauseDetector.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/jersey/server/MetricsRequestEventListener.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/BaseUnits.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/distribution/Histogram.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/util/StringUtils.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/cumulative/CumulativeCounter.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/hystrix/MicrometerMetricsPublisher.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/util/HierarchicalNameMapper.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/composite/CompositeTimer.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/jvm/JvmGcMetrics.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/ImmutableTag.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/internal/package-info.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/util/MeterPartition.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/dropwizard/DropwizardCounter.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/search/package-info.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/logging/LogbackMetrics.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/distribution/TimeWindowPercentileHistogram.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/MeterRegistry.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/jvm/JvmMemoryMetrics.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/util/IOUtils.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/step/StepLong.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/cache/JCacheMetrics.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/noop/NoopTimeGauge.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/dropwizard/DropwizardGauge.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/AbstractTimer.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/LongTaskTimer.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/distribution/HistogramSupport.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/config/InvalidConfigurationException.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/step/StepFunctionTimer.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/composite/CompositeFunctionTimer.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/util/TimeUtils.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/DistributionSummary.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/distribution/pause/PauseDetector.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/MeterBinder.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/noop/NoopMeter.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/distribution/TimeWindowMax.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/FunctionTimer.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/dropwizard/DropwizardConfig.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/noop/package-info.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/config/MeterFilter.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/logging/package-info.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/jetty/JettyStatisticsMetrics.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/Tag.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/Meter.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/jersey/server/JerseyTags.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/composite/AbstractCompositeMeter.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/util/DoubleFormat.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/simple/CountingMode.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/hytrix/MicrometerMetricsPublisherThreadPool.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/jvm/JvmCompilationMetrics.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/jvm/JvmThreadMetrics.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/distribution/package-info.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/dropwizard/DropwizardMeterRegistry.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/composite/CompositeGauge.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/util/AbstractPartition.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/composite/CompositeDistributionSummary.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/noop/NoopFunctionTimer.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2012-2019 the original author or authors.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <https://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/internal/OnlyOnceLoggingDenyMeterFilter.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2021 VMware, Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <https://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/mongodb/DefaultMongoConnectionPoolTagsProvider.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/grpc/MetricCollectingClientInterceptor.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/cache/CaffeineStatsCounter.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/jvm/JvmInfoMetrics.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/grpc/AbstractMetricCollectingInterceptor.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/grpc/MetricCollectingServerInterceptor.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/mongodb/DefaultMongoCommandTagsProvider.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/grpc/MetricCollectingServerCall.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/grpc/package-info.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/mongodb/MongoCommandTagsProvider.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/grpc/MetricCollectingClientCallListener.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/grpc/MetricCollectingServerCallListener.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/grpc/MetricCollectingClientCall.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2019 VMware, Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <https://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/jetty/JettyConnectionMetrics.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/Timer.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/jetty/TimerHandler.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/jetty/OnCompletionAsyncListener.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-

```
jar/io/micrometer/core/util/internal/logging/WarnThenDebugLogger.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/instrument/binder/httpcomponents/MicrometerHttpRequestExecutor.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/instrument/binder/mongodb/MongoMetricsCommandListener.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/instrument/binder/mongodb/MongoMetricsConnectionPoolListener.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/instrument/internal/TimerRunnable.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/instrument/binder/jvm/JvmMemory.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/instrument/internal/TimerCallable.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/instrument/binder/jetty/JettyServerThreadPoolMetrics.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/instrument/binder/jetty/InstrumentedQueuedThreadPool.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/instrument/internal/TimerScheduledExecutorService.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/instrument/binder/httpcomponents/PoolingHttpClientConnectionManagerMetricsBinder.java
No license file was found, but licenses were detected in source scan.
```

```
/*
* Copyright 2018 VMware, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* https://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):

```
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/instrument/logging/LoggingMeterRegistry.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/instrument/StrongReferenceGaugeFunction.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/ipc/http/URLConnectionSender.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/ipc/http/OkHttpSender.java
```

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/push/PushMeterRegistry.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/push/PushRegistryConfig.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/ipc/http/HttpStatusClass.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/kafka/KafkaConsumerMetrics.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/ipc/http/HttpSender.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/util/NamedThreadFactory.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/MultiGauge.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/logging/LoggingRegistryConfig.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2020 VMware, Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <https://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/kafka/KafkaStreamsMetrics.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/config/MeterRegistryConfigValidator.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/http/Outcome.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/jetty/JettyClientMetrics.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/binder/jetty/JettySslHandshakeMetrics.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/instrument/config/validate/InvalidReason.java

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-

jar/io/micrometer/core/instrument/binder/http/DefaultHttpServletRequestTagsProvider.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/instrument/binder/http/HttpServletRequestTagsProvider.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/instrument/internal/CumulativeHistogramLongTaskTimer.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/instrument/binder/db/JooqExecuteListener.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/instrument/binder/kafka/KafkaClientMetrics.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/instrument/binder/jvm/JvmHeapPressureMetrics.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/instrument/config/validate/DurationValidator.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/instrument/binder/okhttp3/OkHttpConnectionPoolMetrics.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/instrument/binder/http/package-info.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/instrument/binder/kafka/KafkaMetrics.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/instrument/binder/http/HttpRequestTags.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/instrument/binder/db/MetricsDSLContext.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/instrument/distribution/TimeWindowSum.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/instrument/binder/jetty/JettyClientTagsProvider.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/instrument/config/validate/Validated.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/instrument/binder/httpcomponents/MicrometerHttpClientInterceptor.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/instrument/binder/commonspool2/CommonsObjectPool2Metrics.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/instrument/binder/httpcomponents/DefaultUriMapper.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/instrument/config/validate/ValidationException.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/instrument/binder/jetty/JettyClientTags.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/instrument/config/validate/PropertyValidator.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/instrument/binder/httpcomponents/HttpContextUtils.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/instrument/binder/jpa/HibernateQueryMetrics.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/instrument/binder/cache/HazelcastIMapAdapter.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-

jar/io/micrometer/core/instrument/AbstractTimerBuilder.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/instrument/distribution/FixedBoundaryVictoriaMetricsHistogram.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/instrument/binder/mongodb/MongoConnectionPoolTagsProvider.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/instrument/step/StepTuple2.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/instrument/step/StepValue.java
No license file was found, but licenses were detected in source scan.

```
/*  
* Copyright 2002-2017 the original author or authors.  
* Copyright 2017-2021 VMware, Inc.  
*  
* Licensed under the Apache License, Version 2.0 (the "License");  
* you may not use this file except in compliance with the License.  
* You may obtain a copy of the License at  
*  
* https://www.apache.org/licenses/LICENSE-2.0  
*  
* Unless required by applicable law or agreed to in writing, software  
* distributed under the License is distributed on an "AS IS" BASIS,  
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
* See the License for the specific language governing permissions and  
* limitations under the License.  
*/
```

Found in path(s):

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/lang/NonNullApi.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/lang/NonNull.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/lang/Nullable.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/lang/NonNullFields.java
No license file was found, but licenses were detected in source scan.

```
/*  
* Copyright 2019 VMware, Inc.  
*  
* Licensed under the Apache License, Version 2.0 (the "License");  
* you may not use this file except in compliance with the License.  
* You may obtain a copy of the License at  
*  
* https://www.apache.org/licenses/LICENSE-2.0  
*  
*/
```

* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.

*/

/*

* Copyright 2012 The Netty Project

*

* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:

*

* <https://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.

*/

/**

* Copyright (c) 2004-2011 QOS.ch

* All rights reserved.

*

* Permission is hereby granted, free of charge, to any person obtaining
* a copy of this software and associated documentation files (the
* "Software"), to deal in the Software without restriction, including
* without limitation the rights to use, copy, modify, merge, publish,
* distribute, sublicense, and/or sell copies of the Software, and to
* permit persons to whom the Software is furnished to do so, subject to
* the following conditions:

*

* The above copyright notice and this permission notice shall be
* included in all copies or substantial portions of the Software.

*

* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
* NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
* LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
* OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
* WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

*

*/

Found in path(s):

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-

jar/io/micrometer/core/util/internal/logging/InternalLogger.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-
jar/io/micrometer/core/util/internal/logging/JdkLogger.java
No license file was found, but licenses were detected in source scan.

```
/*  
* Copyright 2019 VMware, Inc.  
*  
* Licensed under the Apache License, Version 2.0 (the "License");  
* you may not use this file except in compliance with the License.  
* You may obtain a copy of the License at  
*  
* https://www.apache.org/licenses/LICENSE-2.0  
*  
* Unless required by applicable law or agreed to in writing, software  
* distributed under the License is distributed on an "AS IS" BASIS,  
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
* See the License for the specific language governing permissions and  
* limitations under the License.  
*/
```

```
/*  
* Copyright 2013 The Netty Project  
*  
* The Netty Project licenses this file to you under the Apache License,  
* version 2.0 (the "License"); you may not use this file except in compliance  
* with the License. You may obtain a copy of the License at:  
*  
* https://www.apache.org/licenses/LICENSE-2.0  
*  
* Unless required by applicable law or agreed to in writing, software  
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT  
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the  
* License for the specific language governing permissions and limitations  
* under the License.  
*/
```

```
/**  
* Copyright (c) 2004-2011 QOS.ch  
* All rights reserved.  
*  
* Permission is hereby granted, free of charge, to any person obtaining  
* a copy of this software and associated documentation files (the  
* "Software"), to deal in the Software without restriction, including  
* without limitation the rights to use, copy, modify, merge, publish,  
* distribute, sublicense, and/or sell copies of the Software, and to  
* permit persons to whom the Software is furnished to do so, subject to  
* the following conditions:  
*  
* The above copyright notice and this permission notice shall be
```

* included in all copies or substantial portions of the Software.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
* NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
* LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
* OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
* WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
*
*/

Found in path(s):

* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/util/internal/logging/FormattingTuple.java
* /opt/cola/permits/1473460188_1668565716.6098356/0/micrometer-core-1-9-4-sources-jar/io/micrometer/core/util/internal/logging/MessageFormatter.java

1.172 micronaut-oracle-cloud 2.2.1

1.172.1 Available under license :

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,

including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf

of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.173 kafka-avro-serializer 5.5.5

1.173.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
<project xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://maven.apache.org/POM/4.0.0"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">

  <modelVersion>4.0.0</modelVersion>

  <parent>
    <groupId>io.confluent</groupId>
    <artifactId>kafka-schema-registry-parent</artifactId>
    <version>5.5.5</version>
```

```
</parent>

<licenses>
  <license>
    <name>Apache License 2.0</name>
    <url>http://www.apache.org/licenses/LICENSE-2.0.html</url>
    <distribution>repo</distribution>
  </license>
</licenses>

<artifactId>kafka-avro-serializer</artifactId>
<packaging>jar</packaging>
<name>kafka-avro-serializer</name>

<dependencies>
  <dependency>
    <groupId>org.apache.kafka</groupId>
    <artifactId>kafka_${kafka.scala.version}</artifactId>
    <scope>provided</scope>
  </dependency>
  <dependency>
    <groupId>org.apache.avro</groupId>
    <artifactId>avro</artifactId>
  </dependency>
  <dependency>
    <groupId>io.confluent</groupId>
    <artifactId>kafka-schema-serializer</artifactId>
  </dependency>
  <dependency>
    <groupId>io.confluent</groupId>
    <artifactId>kafka-schema-registry-client</artifactId>
  </dependency>
  <dependency>
    <groupId>io.confluent</groupId>
    <artifactId>common-config</artifactId>
  </dependency>

  <dependency>
    <groupId>com.google.guava</groupId>
    <artifactId>guava</artifactId>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <scope>test</scope>
  </dependency>
</dependencies>
```

```

<build>
  <plugins>
    <plugin>
      <groupId>org.apache.avro</groupId>
      <artifactId>avro-maven-plugin</artifactId>
      <executions>
        <execution>
          <phase>generate-sources</phase>
          <goals>
            <goal>schema</goal>
          </goals>
          <configuration>
            <testSourceDirectory>${project.basedir}/src/test/avro</testSourceDirectory>
          </configuration>
        </execution>
      </executions>
    </plugin>
  </plugins>
</build>
</project>

```

Found in path(s):

* /opt/cola/permits/1341640010_1654785818.176366/0/kafka-avro-serializer-5-5-5-jar/META-INF/maven/io.confluent/kafka-avro-serializer/pom.xml

1.174 micronaut-cache 3.5.0

1.174.1 Available under license :

Apache License

Version 2.0, January 2004

<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the

direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of

this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and

wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor

has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.175 metrics-core 4.2.12

1.175.1 Available under license :

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0

Automatic-Module-Name: com.codahale.metrics

Build-Jdk-Spec: 11

Bundle-Description: Metrics is a Java library which gives you unparalleled insight into what your code does in production. Metrics provides a powerful toolkit of ways to measure the behavior of critical components in your production environment.

Bundle-License: <https://www.apache.org/licenses/LICENSE-2.0.html>

Bundle-ManifestVersion: 2

Bundle-Name: Metrics Core

Bundle-SymbolicName: io.dropwizard.metrics.core

Bundle-Version: 4.2.12

Created-By: Apache Maven Bundle Plugin 5.1.8

Export-Package: com.codahale.metrics;uses:="org.slf4j";version="4.2.12"

"

Implementation-Title: Metrics Core

Implementation-Version: 4.2.12

Import-Package: org.slf4j;version="[1.6.0,2.0.0)"

Require-Capability: osgi.ee;filter:="(&(osgi.ee=JavaSE)(version=1.8))"

Tool: Bnd-6.3.1.202206071316

Found in path(s):

* /opt/cola/permits/1470279060_1668107933.5636022/0/metrics-core-4-2-12-jar/META-INF/MANIFEST.MF

1.176 javax-annotation-api 1.3.2

1.176.1 Available under license :

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

1. Definitions.

1.1. Contributor. means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version. means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software. means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable. means the Covered Software in any form other than Source Code.

1.5. Initial Developer. means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work. means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License. means this document.

1.8. Licensable. means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications. means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software. means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims. means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code. means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You. (or .Your.) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, .You. includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, .control. means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the

applicable version of this License or the recipients. rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient.s rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN .AS IS. BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION,

WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as .Participant.) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a .commercial item., as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of .commercial computer software. (as that term is defined at 48 C.F.R. ? 252.227-7014(a)(1)) and .commercial

computer software documentation. as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding

source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous

contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.

Copyright (C)

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author

Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program

is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

"CLASSPATH" EXCEPTION TO THE GPL VERSION 2

Certain source files distributed by Sun Microsystems, Inc. are subject to the following clarification and special exception to the GPL Version 2, but only where Sun has expressly included in the particular source file's header the words

"Sun designates this particular file as subject to the "Classpath" exception as provided by Sun in the License file that accompanied this code."

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License Version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module.? An independent module is a module which is not derived from or based on this library.? If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so.? If you do not wish to do so, delete this exception statement from your version.

1.177 fabric8-::-kubernetes-model-::-apps

4.13.3

1.177.1 Available under license :

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0

Bnd-LastModified: 1619068485705

Build-Jdk-Spec: 1.8

Bundle-Description: Java client for Kubernetes and OpenShift

Bundle-DocURL: <http://redhat.com>

Bundle-License: <http://www.apache.org/licenses/LICENSE-2.0.txt>

Bundle-ManifestVersion: 2

Bundle-Name: Fabric8 :: Kubernetes Model :: Apps

Bundle-SymbolicName: io.fabric8.kubernetes-model-apps

Bundle-Vendor: Red Hat

Bundle-Version: 4.13.3

Created-By: Apache Maven Bundle Plugin

Export-Package: io.fabric8.kubernetes.api.model.apps;uses:="com.fasterxml.jackson.annotation,com.fasterxml.jackson.databind,com.fasterxml.jackson.databind.annotation,io.fabric8.kubernetes.api.builder,io.fabric8.kubernetes.api.model,io.fabric8.kubernetes.model.annotation";versi

on="4.13.3"

Implementation-Title: Fabric8 :: Kubernetes Model :: Apps

Implementation-Vendor: Red Hat

Implementation-Version: 4.13.3

Import-Package: com.fasterxml.jackson.annotation;version="[2.11,3)",com.fasterxml.jackson.databind;version="[2.11,3)",com.fasterxml.jackson.databind.annotation;version="[2.11,3)",io.fabric8.kubernetes.api.builder;version="[4.13,5)",io.fabric8.kubernetes.api.model,io.fabric8.kubernetes.model.annotation;version="[4.13,5)"

Require-Capability: osgi.ee;filter="(&(osgi.ee=JavaSE)(version=1.8))"

Specification-Title: Fabric8 :: Kubernetes Model :: Apps

Specification-Vendor: Red Hat

Specification-Version: 4.13

Tool: Bnd-5.1.1.202006162103

Found in path(s):

* /opt/cola/permits/1288519826_1647861654.16/0/kubernetes-model-apps-4-13-3-jar/META-INF/MANIFEST.MF

No license file was found, but licenses were detected in source scan.

Copyright (C) 2015 Red Hat, Inc.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE>

2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Found in path(s):

* /opt/cola/permits/1288519826_1647861654.16/0/kubernetes-model-apps-4-13-3-jar/META-INF/maven/io.fabric8/kubernetes-model-apps/pom.xml

1.178 caffeine-cache 3.0.3

1.178.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/MpscGrowableArrayQueue.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2014 Ben Manes. All Rights Reserved.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/Ticker.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/stats/CacheStats.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/LinkedDeque.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/Policy.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/RemovalListener.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/stats/DisabledStatsCounter.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/Weigher.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/AsyncLoadingCache.java

* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/BoundedLocalCache.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/Cache.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/RemovalCause.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/WriteOrderDeque.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/UnboundedLocalCache.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/AccessOrderDeque.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/stats/StatsCounter.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/stats/ConcurrentStatsCounter.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/Caffeine.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/LoadingCache.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/AbstractLinkedDeque.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/CacheLoader.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2015 Ben Manes. All Rights Reserved.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

/*

* Written by Doug Lea with assistance from members of JCP JSR-166

* Expert Group and released to the public domain, as explained at

* <http://creativecommons.org/publicdomain/zero/1.0/>

*/

Found in path(s):

* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/StripedBuffer.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2017 Ben Manes. All Rights Reserved.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/Expiry.java

* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/TimerWheel.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2018 Ben Manes. All Rights Reserved.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/AsyncCache.java

* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/LocalAsyncCache.java

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2016 Ben Manes. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/AsyncCacheLoader.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/CaffeineSpec.java
```

No license file was found, but licenses were detected in source scan.

```
// Copyright 2021 Ben Manes. All Rights Reserved.
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
// http://www.apache.org/licenses/LICENSE-2.0
// distributed under the License is distributed on an "AS IS" BASIS,
```

Found in path(s):

```
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLMWAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILMWWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILSMSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WLSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSMSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
```

jar/com/github/benmanes/caffeine/cache/WILW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLSMWW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLMSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SIMWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SIS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WIMSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILSMWA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FWAWRMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SISAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FWAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILMWA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FSARMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLSMWA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FWAWRMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PSAWMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PDMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SISAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FSARMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WISA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PDARMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-

jar/com/github/benmanes/caffeine/cache/FWRMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILMSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILSMWWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSMSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PDAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FDRMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSSMSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FDAWRMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SISA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PDRMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLMWA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PWW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WISWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSSMWW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PDR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SISMSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLMWWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FDARMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SISMWA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WLSMWAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILSMWW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WLSMSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-

jar/com/github/benmanes/caffeine/cache/WLSLMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WISAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SIMSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLSMSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSSMSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WIMSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PWAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSMSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SIAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PD.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSMSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FDAWMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FSRMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WISMWWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILSMSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-

jar/com/github/benmanes/caffeine/cache/PSARMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FDAMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WLSMWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSSMWW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSMWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILSMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WISMSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FWAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSSMSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSMSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSSMWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SISMSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FDW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PSMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSMWA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PSAWRMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILSMWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSMSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PDAWMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILMSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLMWAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-

jar/com/github/benmanes/caffeine/cache/FDWMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PSMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FSAWRMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILMSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FDA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLMSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WISMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PDAMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SIMWAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILMWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SIMSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FSAWMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FDAWRMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILSMSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SIMSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SISMWAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLMWWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILMSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILMWAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SISMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WISMWAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILMWW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-

jar/com/github/benmanes/caffeine/cache/SILSMWAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSSMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PDMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SIMWAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILSMSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WISMSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLSMWAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WISR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLMWAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FDAWMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WIMWAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILMWWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FDMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SIMSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSSMSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSSMSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PSWRMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLMSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-

jar/com/github/benmanes/caffeine/cache/PSAWRMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WLSMWAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSMWAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PSWMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FWARMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSMSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSMWAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SIA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SISMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PDAWRMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WIMWW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FWAMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILSMSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSL.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLSMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WISMWA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLMSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLMSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FDWMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-

jar/com/github/benmanes/caffeine/cache/SISW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSSMWAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FDAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FWWRMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SISMWW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SIW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SISMSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WISMSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PWAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WLSMSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILMSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSSMSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILMWAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WIMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSMWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILMWAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PSAWMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SISMWAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WIMSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-

jar/com/github/benmanes/caffeine/cache/WISMSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PDARMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PSARMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SIWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WIW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSMSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLSMWA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSSMWAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PSRMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WIMSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WISAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PDWMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SISMSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILSMSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PDAWMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WISMSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSSMSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSMWW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILSMWA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-

jar/com/github/benmanes/caffeine/cache/SSMWAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSL.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLSMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PWAWMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FDAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FSAWRMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLSMSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLMSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SIMSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSSMSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILSMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLSMWWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLSMSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SIL.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FWWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SISMSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FWAWMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILMWAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-

jar/com/github/benmanes/caffeine/cache/WIMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSMSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLMSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FDMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILSMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLMWAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLSMWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSMWA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILMWW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSMWWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PDAMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSMSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WISMWAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FDARMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILMSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSSMSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FSWMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-

jar/com/github/benmanes/caffeine/cache/SISR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FWAWMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILMSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WLSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLMWW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/LocalCacheFactory.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FWWMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLMSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FSWRMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSMWAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILMSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PSAMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FWA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WISMWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WLSMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSSMSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSSMWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WLSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FSMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-

jar/com/github/benmanes/caffeine/cache/FSMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILMSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PWMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WISMSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSSMWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLMWAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SIR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLMSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WLSMSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLMSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WLSMWAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILSMWAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSSMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILMWAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSSMWAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SIMSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SISMSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSMWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-

jar/com/github/benmanes/caffeine/cache/SSLWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLMSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILSMSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSSMWA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILSMSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FDRMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILSMWWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WLSLA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSMSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SIAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SIMWA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSSMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSMSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSSMWAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FD.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SI.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILSMWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-

jar/com/github/benmanes/caffeine/cache/WIS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WI.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PSWRMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PDW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSSMWA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLMSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILMSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WIMWA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLMWAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FWMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WIAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILMSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PWARMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WLSMWW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PSRMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PDAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WLSMSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSSMSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-

jar/com/github/benmanes/caffeine/cache/FWAMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SISMWAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WISW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SIMWWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WIMWWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PWA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLMSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PWRMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PDA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FWW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSSMSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSSMWAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLMWA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILSMSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILSMWAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FDR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLSMSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLSMSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WIAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-

jar/com/github/benmanes/caffeine/cache/SISMSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILSMWW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WIAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILSMSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FDAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLMWAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILMSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FSWRMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PWAMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FDWRMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FSAMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILSMSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLMSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLSMWAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WISMWW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FSAMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FWARMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WISAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-

jar/com/github/benmanes/caffeine/cache/WSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PWWRMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PDWRMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WLSLR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SISMWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSMSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FSWMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILSMWAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PWAMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PSWMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSMWAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FDWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FSAWMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WIL.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FDWRMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILSMSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PDWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PWAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WIMSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FWMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-

jar/com/github/benmanes/caffeine/cache/WSSMWAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SIAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WIMWAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILSMWAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PWAWRMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PSAMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILSMSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SISAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WISMWAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WISMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SISWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FDAMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PWWMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WISMSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSMSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FWWMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/NodeFactory.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PWAWRMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILSMWAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSSMWWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SISMWWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-

jar/com/github/benmanes/caffeine/cache/WIR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PWWRMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WIA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLMSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILMSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FWRMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILMSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PWARMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILMVAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLMSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WIWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLMWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSMWW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WIMSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PWWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLSMWWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PDRMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PWMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PWWMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-

jar/com/github/benmanes/caffeine/cache/SILMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PDWRMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WIMWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SIMWW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PDAWRMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PDAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PWRMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLMWW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLMWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SILMWA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SIMWAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SIMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FWAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILMWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-

jar/com/github/benmanes/caffeine/cache/PDWMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SIMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WIMSW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FSRMS.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILMSWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WIMWAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FWWRMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/PWAWMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSLSMSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WLSMSA.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SIMSAR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/FSR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSSMSAW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSMWAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSMSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WILSMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SSSAWR.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSSMW.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/WSLMSR.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2019 Ben Manes. All Rights Reserved.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

- * Unless required by applicable law or agreed to in writing, software
- * distributed under the License is distributed on an "AS IS" BASIS,
- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.
- */

Found in path(s):

- * /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/Scheduler.java
- * /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/Pacer.java

No license file was found, but licenses were detected in source scan.

/*

- * Copyright 2015 Ben Manes. All Rights Reserved.
- *
- * Licensed under the Apache License, Version 2.0 (the "License");
- * you may not use this file except in compliance with the License.
- * You may obtain a copy of the License at
- *
- * <http://www.apache.org/licenses/LICENSE-2.0>
- *
- * Unless required by applicable law or agreed to in writing, software
- * distributed under the License is distributed on an "AS IS" BASIS,
- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.
- */

Found in path(s):

- * /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/FrequencySketch.java
- * /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/Buffer.java
- * /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/Async.java
- * /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/LocalAsyncLoadingCache.java
- * /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/References.java
- * /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/stats/package-info.java
- * /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/LocalLoadingCache.java
- * /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-jar/com/github/benmanes/caffeine/cache/WriteThroughEntry.java
- * /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-

jar/com/github/benmanes/caffeine/cache/Node.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/BoundedBuffer.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/package-info.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/SerializationProxy.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/LocalCache.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/stats/GuardedStatsCounter.java
* /opt/cola/permits/1341639961_1654809995.3473003/0/caffeine-3-0-3-sources-
jar/com/github/benmanes/caffeine/cache/LocalManualCache.java

1.179 metrics-integration-with-jmx 4.1.29

1.179.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work,

where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or

for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason

of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 2010-2013 Coda Hale and Yammer, Inc., 2014-2020 Dropwizard Team

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.180 openapi/swagger-support 4.5.2

1.180.1 Available under license :

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all

other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and

subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed

as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the

Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.181 micronaut-views 3.6.0

1.181.1 Available under license :

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or

agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.182 apache-groovy 4.0.1

1.182.1 Available under license :

This directory contains generated LICENSE files and snippets used to generate those files. See the assemble.gradle file (updateLicenses task) for details on how this is done. Snippets have predefined suffix values in their name to determine which files they go into. LICENSE (the one for source), LICENSE-DOC and LICENSE-JARJAR get snippets containing SRC, DOC and JARJAR respectively. LICENSE-BINZIP gets JARJAR and BINZIP snippets. In addition, LICENSE files are generated for these subprojects: groovy-docgenerator, groovy-groovydoc, groovy-groovysh, groovy-jsr223
Copyright (c) Nicolas Gallagher and Jonathan Neal

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and

attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the

appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

JSR223 License

The following classes within this product:

```
org.codehaus.groovy.jsr223.GroovyCompiledScript  
org.codehaus.groovy.jsr223.GroovyScriptEngineFactory  
org.codehaus.groovy.jsr223.GroovyScriptEngineImpl
```

were derived from reference implementation files developed by Sun in
collaboration with the Groovy community. The reference implementation
has a BSD-style license. Details can be found in: licenses/jsr223-license.txt

Apache Commons Lang

Copyright 2001-2015 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

This product includes software from the Spring Framework,
under the Apache License 2.0 (see: `StringUtils.containsWhitespace()`)
Copyright jQuery Foundation and other contributors, <https://jquery.org/>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

XStream License (optional dependency when serializing AST as XML)

This product bundles the XStream jar, which is available under a "3-clause BSD" license. For details, see licenses/xstream-license.txt.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of,

the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

ANTLR 4 License

Antlr4 is released under a BSD 3-clause license.
See [licenses/antlr4-license.txt](#) for details.

ASM License

ASM uses a 3-clause BSD license. For details, see [licenses/asm-license.txt](#).
Apache Groovy
Copyright 2003-2022 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

This product includes/uses ANTLR4 (<https://github.com/antlr/antlr4>)
Copyright (c) 2012-2017 The ANTLR Project. All rights reserved.
ANTLR 4 License

[The "BSD 3-clause license"]
Copyright (c) 2012-2017 The ANTLR Project. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors

may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NORMALIZE.CSS LICENSE

The following file is used with documentation:

`org/codehaus/groovy/tools/stylesheet.css`

Copyright (c) Nicolas Gallagher and Jonathan Neal

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,

and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the

Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,

whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

ANTLR 4 License

Antlr4 is released under a BSD 3-clause license.
See [licenses/antlr4-license.txt](#) for details.

ASM License

ASM uses a 3-clause BSD license. For details, see [licenses/asm-license.txt](#).

Hamcrest License (needed when using optional JUnit dependency)

This product bundles the Hamcrest jar, which is available under a BSD license. For details, see [licenses/hamcrest-license.txt](#).

JLine2 License (optional dependency used with groovysh)

This product bundles the JLine2 jar, which is available under a BSD License. For details, see [licenses/jline2-license.txt](#).

JSR166y License (optionally used by the optional GPar dependency)

This product bundles the jsr166y jar (containing works from the JSR-166 EG, Doug Lea, and Jason T. Greene) made available in the public domain. For details, see [licenses/jsr166y-license.txt](#).

JSR223 License

The following classes within this product:

```
org.codehaus.groovy.jsr223.GroovyCompiledScript  
org.codehaus.groovy.jsr223.GroovyScriptEngineFactory  
org.codehaus.groovy.jsr223.GroovyScriptEngineImpl
```

were derived from reference implementation files developed by Sun in collaboration with the Groovy community. The reference implementation

has a BSD-style license. Details can be found in: licenses/jsr223-license.txt

JUnit Licenses (optional dependencies when using Groovy for testing)

This product bundles the JUnit 4 jar, which is available under the Eclipse Public License v1.0. For details, see licenses/junit4-license.txt.

This product bundles several JUnit 5 jars, which are available under the Eclipse Public License v2.0. For details, see licenses/junit5-license.txt.

normalize.css License

The stylesheet.css file (originally normalize.css) is used by the groovydoc and docgenerator components for groovy-jdk/gapi documentation. It is made available under a MIT License:
licenses/normalize-stylesheet-license.txt

XStream License (optional dependency when serializing AST as XML)

This product bundles the XStream jar, which is available under a "3-clause BSD" license. For details, see licenses/xstream-license.txt.

This convenience zip embeds Groovy's src and doc zips.
See also src/LICENSE and doc/LICENSE files for additional license information.

////////////////////////////////////

Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to you under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY

KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

////////////////////////////////////

= License

This work is licensed under the <http://www.apache.org/licenses/LICENSE-2.0>[Apache License, Version 2.0].

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate

as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify

the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include

the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

AsciiDoc License

This product uses the style.css from asciidoctor.org within documentation. The file is available under the MIT License. For details, see licenses/asciidoc-style-license.txt.

jQuery License

The following file is used within documentation:

`src/spec/assets/css/jquery-2.1.1.min.js`

This file is made available under the following MIT license:
licenses/jquery-js-license.txt

JSR223 License

The following classes within this product:

`org.codehaus.groovy.jsr223.GroovyCompiledScript`
`org.codehaus.groovy.jsr223.GroovyScriptEngineFactory`
`org.codehaus.groovy.jsr223.GroovyScriptEngineImpl`

were derived from reference implementation files developed by Sun in collaboration with the Groovy community. The reference implementation has a BSD-style license. Details can be found in: licenses/jsr223-license.txt

normalize.css License

The stylesheet.css file (originally normalize.css) is used by the groovydoc and docgenerator components for groovy-jdk/gapi documentation. It is made available under a MIT License:
licenses/normalize-stylesheet-license.txt
normalize.css License

The stylesheet.css file (originally normalize.css) is used by the groovydoc and docgenerator components for groovy-jdk/gapi documentation. It is made available under a MIT License:
licenses/normalize-stylesheet-license.txt
Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

- a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
- b) in the case of each subsequent Contributor:
 - i) changes to the Program, and
 - ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

- a) it complies with the terms and conditions of this Agreement; and
- b) its license agreement:
 - i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
 - ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
 - iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
 - iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

- a) it must be made available under this Agreement; and
- b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes

the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF

LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

JSR223 License

The following classes within this product:

```
org.codehaus.groovy.jsr223.GroovyCompiledScript
org.codehaus.groovy.jsr223.GroovyScriptEngineFactory
org.codehaus.groovy.jsr223.GroovyScriptEngineImpl
```

were derived from reference implementation files developed by Sun in collaboration with the Groovy community. The reference implementation has a BSD-style license. Details can be found in: licenses/jsr223-license.txt
Revised BSD license

This is a specific instance of the Open Source Initiative (OSI) BSD license template
<http://www.opensource.org/licenses/bsd-license.php>

Copyright 2004-2009 Brent Fulgham
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of "The Computer Language Benchmarks Game" nor the name of "The Computer Language Shootout Benchmarks" nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN

IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

JSR166y License (optionally used by the optional GPar dependency)

This product bundles the jsr166y jar (containing works from the JSR-166 EG, Doug Lea, and Jason T. Greene) made available in the public domain. For details, see licenses/jsr166y-license.txt.

JLine2 License (optional dependency used with groovysh)

This product bundles the JLine2 jar, which is available under a BSD License. For details, see licenses/jline2-license.txt.

ANTLR 4 License

Antlr4 is released under a BSD 3-clause license.

See licenses/antlr4-license.txt for details.

Apache Groovy

Copyright 2003-2022 The Apache Software Foundation

This product includes software developed at The Apache Software Foundation (<http://www.apache.org/>).

The Java source files in src/main/java/org/apache/groovy/util/concurrent/concurrentlinkedhashmap/ are from <https://github.com/ben-manes/concurrentlinkedhashmap> and the following notice applies: Copyright 2010-2012 Google Inc. All Rights Reserved.

The Java source file src/main/java/org/apache/groovy/util/concurrent/ConcurrentReferenceHashMap is from <https://github.com/hazelcast/hazelcast> and the following notice applies: Copyright (c) 2008-2020, Hazelcast, Inc. All Rights Reserved.

This product bundles icons from the famfamfam.com silk icons set

<http://www.famfamfam.com/lab/icons/silk/>

Licensed under the Creative Commons Attribution Licence v2.5

<http://creativecommons.org/licenses/by/2.5/>

Eclipse Public License - v 2.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE (AGREEMENT). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. Definitions

Contribution means:

- a) in the case of the initial Contributor, the initial content Distributed under this Agreement, and
- b) in the case of each subsequent Contributor:
 - i) changes to the Program, and
 - ii) additions to the Program; where such changes and/or additions to the Program originate from and are Distributed by that particular Contributor. A Contribution originates from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include changes or additions to the Program that are not Modified Works.

Contributor means any person or entity that Distributes the Program.

Licensed Patents mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

Program means the Contributions Distributed in accordance with this Agreement.

Recipient means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

Derivative Works shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

Modified Works shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

Distribute means the acts of a) distributing or b) making available in any manner that enables the transfer of a copy.

Source Code means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

Secondary License means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.

2. Grant of Rights

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

e) Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

3. Requirements

3.1 If a Contributor Distributes the Program in any form, then:

a) the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and

b) the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:

- i) effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
- ii) effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
- iii) does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and
- iv) requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.

3.2 When the Program is Distributed as Source Code:

a) it must be made available under this Agreement, or if the Program (i) is combined with other material in a separate file or files made available under a Secondary License, and (ii) the initial Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available under the terms of such Secondary Licenses, and

b) a copy of this Agreement must be included with each copy of the Program.

3.3 Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability (notices) contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

4. Commercial Distribution

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor (Commercial Contributor) hereby agrees to defend and indemnify every other Contributor (Indemnified Contributor) against any losses, damages and costs (collectively Losses) arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify

the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. No Warranty

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN AS IS BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. Disclaimer of Liability

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. General

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the

Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.

Exhibit A - Form of Secondary Licenses Notice

This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: {name license(s), version(s), and exceptions or additional permissions here}.

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the

outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable

copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and

do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

ANTLR 4 License

Antlr4 is released under a BSD 3-clause license.
See licenses/antlr4-license.txt for details.

ASM License

ASM uses a 3-clause BSD license. For details, see licenses/asm-license.txt.

Hamcrest License (needed when using optional JUnit dependency)

This product bundles the Hamcrest jar, which is available under a BSD license. For details, see licenses/hamcrest-license.txt.

JLine2 License (optional dependency used with groovysh)

This product bundles the JLine2 jar, which is available under a BSD License. For details, see licenses/jline2-license.txt.

JSR166y License (optionally used by the optional GParas dependency)

This product bundles the jsr166y jar (containing works from the JSR-166 EG, Doug Lea, and Jason T. Greene) made available in the public domain. For details, see licenses/jsr166y-license.txt.

JSR223 License

The following classes within this product:

org.codehaus.groovy.jsr223.GroovyCompiledScript
org.codehaus.groovy.jsr223.GroovyScriptEngineFactory
org.codehaus.groovy.jsr223.GroovyScriptEngineImpl

were derived from reference implementation files developed by Sun in collaboration with the Groovy community. The reference implementation has a BSD-style license. Details can be found in: licenses/jsr223-license.txt

JUnit Licenses (optional dependencies when using Groovy for testing)

This product bundles the JUnit 4 jar, which is available under the Eclipse Public License v1.0. For details, see licenses/junit4-license.txt.

This product bundles several JUnit 5 jars, which are available under the Eclipse Public License v2.0. For details, see licenses/junit5-license.txt.

normalize.css License

The stylesheet.css file (originally normalize.css) is used by the groovydoc and docgenerator components for groovy-jdk/gapi documentation. It is made available under a MIT License:
licenses/normalize-stylesheet-license.txt

XStream License (optional dependency when serializing AST as XML)

This product bundles the XStream jar, which is available under a "3-clause BSD" license. For details, see licenses/xstream-license.txt.
COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

1. Definitions.

1.1. Contributor means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable means the Covered Software in any form other than Source Code.

1.5. Initial Developer means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License means this document.

1.8. Licensable means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications means the Source Code and Executable form of any of the following: A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous

Modifications; B. Any new file that contains any part of the Original Software or previous Modification; or C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You (or Your) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant. Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof);

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License;

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant. Conditioned upon Your compliance with Section 3.1 below

and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark)

Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code. Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications. The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices. You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove

or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms. You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions. You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipients rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works. You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions. Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions. You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available

under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions. When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY. COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor)

shall survive termination.

7. **LIMITATION OF LIABILITY.** UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTYS NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. **U.S. GOVERNMENT END USERS.** The Covered Software is a commercial item, as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of commercial computer software (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer software documentation as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. **MISCELLANEOUS.** This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdictions conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. **RESPONSIBILITY FOR CLAIMS.** As between Initial Developer and the Contributors, each party is responsible for claims and damages arising,

directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

For Covered Software in this distribution, this License shall be governed by the laws of the State of California (excluding conflict-of-law provisions).

Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

BSD License

Copyright (c) 2000-2015 www.hamcrest.org
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of Hamcrest nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The person or persons who have associated work with this document (the "Dedicator" or "Certifier") hereby either (a) certifies that, to the best of

his knowledge, the work of authorship identified is in the public domain of the country from which the work is published, or (b) hereby dedicates whatever copyright the dedicators holds in the work of authorship identified below (the "Work") to the public domain. A certifier, moreover, dedicates any copyright interest he may have in the associated work, and for these purposes, is described as a "dedicator" below.

A certifier has taken reasonable steps to verify the copyright status of this work. Certifier recognizes that his good faith efforts may not shield him from liability if in fact the work certified is not in the public domain.

Dedicator makes this dedication for the benefit of the public at large and to the detriment of the Dedicator's heirs and successors. Dedicator intends this dedication to be an overt act of relinquishment in perpetuity of all present and future rights under copyright law, whether vested or contingent, in the Work. Dedicator understands that such relinquishment of all rights includes the relinquishment of all rights to enforce (by lawsuit or otherwise) those copyrights in the Work.

Dedicator recognizes that, once placed in the public domain, the Work may be freely reproduced, distributed, transmitted, used, modified, built upon, or otherwise exploited by anyone for any purpose, commercial or non-commercial, and in any way, including by methods that have not yet been invented or conceived.

Apache Groovy
Copyright 2003-2022 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

This product embeds the OpenBeans jar within its grooid jar artifacts
OpenBeans includes/uses files from Apache Harmony and the following notice applies
Copyright 2006, 2010 The Apache Software Foundation.

Portions of Apache Harmony were originally developed by Intel Corporation and are licensed to the Apache Software Foundation under the "Software Grant and Corporate Contribution License Agreement" and for which the following copyright notices apply

- (C) Copyright 2005 Intel Corporation
- (C) Copyright 2005-2006 Intel Corporation
- (C) Copyright 2006 Intel Corporation

Apache Groovy
Copyright 2003-2022 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

This product includes/uses ANTLR4 (<https://github.com/antlr/antlr4>)
Copyright (c) 2012-2017 The ANTLR Project. All rights reserved.

This product bundles the JUnit4 jar (junit.org)
which is available under the terms of the Eclipse Public License v1.0

This product bundles several of the JUnit5 jars (junit.org)
which are available under the terms of the Eclipse Public License v2.0

This product embeds the OpenBeans jar within its grooid jar artifacts
OpenBeans includes/uses files from Apache Harmony and the following notice applies
Copyright 2006, 2010 The Apache Software Foundation.

Portions of Apache Harmony were originally developed by Intel Corporation and are
licensed to the Apache Software Foundation under the "Software Grant and Corporate
Contribution License Agreement" and for which the following copyright notices apply

- (C) Copyright 2005 Intel Corporation
- (C) Copyright 2005-2006 Intel Corporation
- (C) Copyright 2006 Intel Corporation

This product bundles icons from the famfamfam.com silk icons set
<http://www.famfamfam.com/lab/icons/silk/>
Licensed under the Creative Commons Attribution Licence v2.5
<http://creativecommons.org/licenses/by/2.5/>

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation

source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable

(except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and

may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify,

defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

NORMALIZE.CSS LICENSE

The following file is used with documentation:

org/codehaus/groovy/tools/stylesheet.css

Copyright (c) Nicolas Gallagher and Jonathan Neal

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache Groovy

Copyright 2003-2022 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

This product includes/uses ANTLR4 (<https://github.com/antlr/antlr4>)
Copyright (c) 2012-2017 The ANTLR Project. All rights reserved.

This product bundles the JUnit4 jar (junit.org)
which is available under the terms of the Eclipse Public License v1.0

This product bundles several of the JUnit5 jars (junit.org)
which are available under the terms of the Eclipse Public License v2.0

This product embeds the OpenBeans jar within its grooid jar artifacts
OpenBeans includes/uses files from Apache Harmony and the following notice applies
Copyright 2006, 2010 The Apache Software Foundation.

Portions of Apache Harmony were originally developed by Intel Corporation and are licensed to the Apache Software Foundation under the "Software Grant and Corporate Contribution License Agreement" and for which the following copyright notices apply

(C) Copyright 2005 Intel Corporation

(C) Copyright 2005-2006 Intel Corporation

(C) Copyright 2006 Intel Corporation

This product bundles icons from the famfamfam.com silk icons set
<http://www.famfamfam.com/lab/icons/silk/>
Licensed under the Creative Commons Attribution Licence v2.5
<http://creativecommons.org/licenses/by/2.5/>

This convenience zip embeds Groovy's src and doc zips.
See also src/NOTICE and doc/NOTICE files for additional notice information.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and

attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the

appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

AsciiDoc License

This product uses the style.css from asciidoctor.org within
documentation. The file is available under the MIT License.
For details, see licenses/asciidoc-style-license.txt.

JQuery License

The following file is used within documentation:

`src/spec/assets/css/jquery-2.1.1.min.js`

This file is made available under the following MIT license:
licenses/jquery-js-license.txt

normalize.css License

The stylesheet.css file (originally normalize.css) is used by the
groovydoc and docgenerator components for groovy-jdk/gapi documentation.
It is made available under a MIT License:
licenses/normalize-stylesheet-license.txt

ASM License

ASM: a very small and fast Java bytecode manipulation framework
Copyright (c) 2000-2011 INRIA, France Telecom
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The MIT License

Copyright (C) 2012-2015 Dan Allen, Ryan Waldron and the AsciiDoctor Project

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2003-2006, Joe Walnes

Copyright (c) 2006-2009, 2011 XStream Committers

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of

conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of XStream nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache Groovy

Copyright 2003-2022 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

This product bundles icons from the famfamfam.com silk icons set

<http://www.famfamfam.com/lab/icons/silk/>

Licensed under the Creative Commons Attribution Licence v2.5

<http://creativecommons.org/licenses/by/2.5/>

Apache Groovy

Copyright 2003-2022 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

This product includes/uses ANTLR4 (<https://github.com/antlr/antlr4>)

Copyright (c) 2012-2017 The ANTLR Project. All rights reserved.

This product embeds the OpenBeans jar within its grooid jar artifacts

OpenBeans includes/uses files from Apache Harmony and the following notice applies

Copyright 2006, 2010 The Apache Software Foundation.

Portions of Apache Harmony were originally developed by Intel Corporation and are licensed to the Apache Software Foundation under the "Software Grant and Corporate Contribution License Agreement" and for which the following copyright notices apply

(C) Copyright 2005 Intel Corporation

(C) Copyright 2005-2006 Intel Corporation

(C) Copyright 2006 Intel Corporation

Hamcrest License (needed when using optional JUnit dependency)

This product bundles the Hamcrest jar, which is available under a BSD license. For details, see licenses/hamcrest-license.txt.

NORMALIZE.CSS LICENSE

The following file is used with documentation:

`org/codehaus/groovy/tools/groovydoc/gstringTemplates/topLevel/styleSheet.css`

Copyright (c) Nicolas Gallagher and Jonathan Neal

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

JUnit Licenses (optional dependencies when using Groovy for testing)

This product bundles the JUnit 4 jar, which is available under the Eclipse Public License v1.0. For details, see licenses/junit4-license.txt.

This product bundles several JUnit 5 jars, which are available under the Eclipse Public License v2.0. For details, see licenses/junit5-license.txt.

JQuery License

The following file is used within documentation:

`src/spec/assets/css/jquery-2.1.1.min.js`

This file is made available under the following MIT license:

licenses/jquery-js-license.txt

////////////////////////////////////

Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to you under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

////////////////////////////////////

= Contributors

The Groovy team would like to thank the contributors of this documentation (in alphabetical order of last/surname):

- * <https://github.com/mojavelinux>[Dan Allen]
- * <https://github.com/and-dmitry>[Dmitry Andreychuk]
- * <http://hamletdarcy.blogspot.fr/>[Hamlet D'Arcy]
- * <https://github.com/anshbansal>[Aseem Bansal]
- * <https://github.com/bura>[Andrey Bloschetsov]
- * <https://github.com/JBrownVisualSpection>[J Brown]
- * <https://github.com/jeffbrown>[Jeff Scott Brown]
- * <http://twitter.com/CedricChampeau>[Cdric Champeau]
- * <https://github.com/tobia>[Tobia Conforto]
- * <https://github.com/ddimitrov>[Dimitar Dimitrov]
- * <http://twitter.com/werdnagreb>[Andrew Eisenberg]
- * <https://github.com/erdi>[Marcin Erdmann]
- * <https://github.com/christoph-frick>[Christoph Frick]
- * <http://twitter.com/marioggar>[Mario Garca]
- * <https://github.com/davidmichaelkarr>[David Michael Karr]
- * http://twitter.com/paulk_asert[Paul King]
- * <http://twitter.com/glaforge>[Guillaume Laforge]
- * <http://twitter.com/pledbrook>[Peter Ledbrook]
- * <http://grantmcconnaughey.github.io/>[Grant McConnaughey]
- * <https://github.com/eric-milles>[Eric Milles]
- * <https://github.com/dnahodil>[David Nahodil]
- * <https://github.com/jnorthr>[James Northrop]
- * <https://github.com/marcpa00>[Marc Paquette]
- * <https://github.com/michaelss>[Michael Schuenck]
- * <https://github.com/PascalSchumacher>[Pascal Schumacher]

- * <https://github.com/shils>[Shil Sinha]
- * <https://github.com/stavytskyi>[Maksym Stavytskyi]
- * <https://twitter.com/asteingr>[Andr Steingre]
- * https://twitter.com/daniel_sun[Daniel Sun]
- * <https://github.com/EPadronU>[Edinson Padrn Urdaneta]
- * <https://github.com/keeganwitt>[Keegan Witt]

ASM License

ASM uses a 3-clause BSD license. For details, see [licenses/asm-license.txt](#).

Asciidoc License

This product uses the style.css from [asciidoc.org](#) within documentation. The file is available under the MIT License.

For details, see [licenses/asciidoc-style-license.txt](#).

Copyright (c) 2002-2012, the original author or authors.

All rights reserved.

<http://www.opensource.org/licenses/bsd-license.php>

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of JLine nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or

agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

NORMALIZE.CSS LICENSE

The following file is used with documentation:

<org/codehaus/groovy/tools/groovydoc/gstringTemplates/topLevel/stylesheet.css>

Copyright (c) Nicolas Gallagher and Jonathan Neal

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache Groovy

Copyright 2003-2022 The Apache Software Foundation

This product includes software developed at

The Apache Software Foundation (<http://www.apache.org/>).

Copyright (c) 2006, Sun Microsystems, Inc.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the Sun Microsystems, Inc. nor the names of contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the

outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable

copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and

do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.183 fabric8-:::kubernetes-model-:::core

4.13.3

1.183.1 Available under license :

No license file was found, but licenses were detected in source scan.

Copyright (C) 2015 Red Hat, Inc.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE>

2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Found in path(s):

* /opt/cola/permits/1288520097_1648099091.58/0/kubernetes-client-4-13-3-jar/META-
INF/maven/io.fabric8/kubernetes-client/pom.xml

No license file was found, but licenses were detected in source scan.

#

Copyright (C) 2015 Red Hat, Inc.

#

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

#

<http://www.apache.org/licenses/LICENSE-2.0>

#

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

#

io.fabric8.kubernetes.client.AppsAPIGroupExtensionAdapter

io.fabric8.kubernetes.client.AdmissionRegistrationAPIGroupExtensionAdapter

io.fabric8.kubernetes.client.V1AdmissionRegistrationAPIGroupExtensionAdapter

io.fabric8.kubernetes.client.V1beta1AdmissionRegistrationAPIGroupExtensionAdapter

io.fabric8.kubernetes.client.AutoscalingAPIGroupExtensionAdapter

io.fabric8.kubernetes.client.ApiextensionsAPIGroupExtensionAdapter

io.fabric8.kubernetes.client.AuthorizationAPIGroupExtensionAdapter

io.fabric8.kubernetes.client.V1AutoscalingAPIGroupExtensionAdapter

io.fabric8.kubernetes.client.V2beta1AutoscalingAPIGroupExtensionAdapter

io.fabric8.kubernetes.client.V2beta2AutoscalingAPIGroupExtensionAdapter

io.fabric8.kubernetes.client.BatchAPIGroupExtensionAdapter

io.fabric8.kubernetes.client.ExtensionsAPIGroupExtensionAdapter

io.fabric8.kubernetes.client.MetricAPIGroupExtensionAdapter

io.fabric8.kubernetes.client.NetworkAPIGroupExtensionAdapter

io.fabric8.kubernetes.client.PolicyAPIGroupExtensionAdapter
io.fabric8.kubernetes.client.RbacAPIGroupExtensionAdapter
io.fabric8.kubernetes.client.SchedulingAPIGroupExtensionAdapter
io.fabric8.kubernetes.client.SettingsAPIGroupExtensionAdapter
io.fabric8.kubernetes.client.StorageAPIGroupExtensionAdapter
io.fabric8.kubernetes.client.V1APIGroupExtensionAdapter
io.fabric8.kubernetes.client.V1ApiextensionsAPIGroupExtensionAdapter
io.fabric8.kubernetes.client.V1beta1ApiextensionsAPIGroupExtensionAdapter
io.fabric8.kubernetes.client.V1AuthorizationAPIGroupExtensionAdapter
io.fabric8.kubernetes.client.V1beta1AuthorizationAPIGroupExtensionAdapter
io.fabric8.kubernetes.client.V1NetworkAPIGroupExtensionAdapter
io.fabric8.kubernetes.client.V1beta1NetworkAPIGroupExtensionAdapter

Found in path(s):

* /opt/cola/permits/1288520097_1648099091.58/0/kubernetes-client-4-13-3-jar/META-INF/services/io.fabric8.kubernetes.client.ExtensionAdapter

No license file was found, but licenses were detected in source scan.

```
#  
# Copyright (C) 2015 Red Hat, Inc.  
#  
# Licensed under the Apache License, Version 2.0 (the "License");  
# you may not use this file except in compliance with the License.  
# You may obtain a copy of the License at  
#  
#   http://www.apache.org/licenses/LICENSE-2.0  
#  
# Unless required by applicable law or agreed to in writing, software  
# distributed under the License is distributed on an "AS IS" BASIS,  
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
# See the License for the specific language governing permissions and  
# limitations under the License.  
#
```

io.fabric8.kubernetes.client.URLFromEnvVarsImpl
io.fabric8.kubernetes.client.URLFromIngressImpl
io.fabric8.kubernetes.client.URLFromNodePortImpl
io.fabric8.kubernetes.client.URLFromClusterIPImpl

Found in path(s):

* /opt/cola/permits/1288520097_1648099091.58/0/kubernetes-client-4-13-3-jar/META-INF/services/io.fabric8.kubernetes.client.ServiceToURLProvider

1.184 jackson-datatype-guava 2.14.0

1.184.1 Available under license :

This copy of Jackson JSON processor `jackson-datatype-guava` module is licensed under the Apache (Software) License, version 2.0 ("the License").

See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

<http://www.apache.org/licenses/LICENSE-2.0>

1.185 json-smart 2.4.7

1.185.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2011-2014 JSON-SMART authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1183889917_1627493840.74/0/json-smart-2-4-7-sources-1-
jar/net/minidev/json/writer/DefaultMapper.java
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2011 JSON-SMART authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
```

- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.
- */

Found in path(s):

- * /opt/cola/permits/1183889917_1627493840.74/0/json-smart-2-4-7-sources-1-jar/net/minidev/json/JSONStyle.java
- * /opt/cola/permits/1183889917_1627493840.74/0/json-smart-2-4-7-sources-1-jar/net/minidev/json/parser/JSONParserByteArray.java
- * /opt/cola/permits/1183889917_1627493840.74/0/json-smart-2-4-7-sources-1-jar/net/minidev/json/JSONStreamAware.java
- * /opt/cola/permits/1183889917_1627493840.74/0/json-smart-2-4-7-sources-1-jar/net/minidev/json/writer/JsonReaderI.java
- * /opt/cola/permits/1183889917_1627493840.74/0/json-smart-2-4-7-sources-1-jar/net/minidev/json/JStylerObj.java
- * /opt/cola/permits/1183889917_1627493840.74/0/json-smart-2-4-7-sources-1-jar/net/minidev/json/parser/JSONParserStream.java
- * /opt/cola/permits/1183889917_1627493840.74/0/json-smart-2-4-7-sources-1-jar/net/minidev/json/parser/JSONParser.java
- * /opt/cola/permits/1183889917_1627493840.74/0/json-smart-2-4-7-sources-1-jar/net/minidev/json/JSONAwareEx.java
- * /opt/cola/permits/1183889917_1627493840.74/0/json-smart-2-4-7-sources-1-jar/net/minidev/json/parser/JSONParserString.java
- * /opt/cola/permits/1183889917_1627493840.74/0/json-smart-2-4-7-sources-1-jar/net/minidev/json/JSONStreamAwareEx.java
- * /opt/cola/permits/1183889917_1627493840.74/0/json-smart-2-4-7-sources-1-jar/net/minidev/json/JSONUtil.java
- * /opt/cola/permits/1183889917_1627493840.74/0/json-smart-2-4-7-sources-1-jar/net/minidev/json/writer/CompressorMapper.java
- * /opt/cola/permits/1183889917_1627493840.74/0/json-smart-2-4-7-sources-1-jar/net/minidev/json/parser/ParseException.java
- * /opt/cola/permits/1183889917_1627493840.74/0/json-smart-2-4-7-sources-1-jar/net/minidev/json/JSONValue.java
- * /opt/cola/permits/1183889917_1627493840.74/0/json-smart-2-4-7-sources-1-jar/net/minidev/json/parser/JSONParserMemory.java
- * /opt/cola/permits/1183889917_1627493840.74/0/json-smart-2-4-7-sources-1-jar/net/minidev/json/writer/DefaultMapperCollection.java
- * /opt/cola/permits/1183889917_1627493840.74/0/json-smart-2-4-7-sources-1-jar/net/minidev/json/JSONNavi.java
- * /opt/cola/permits/1183889917_1627493840.74/0/json-smart-2-4-7-sources-1-jar/net/minidev/json/writer/BeansMapper.java
- * /opt/cola/permits/1183889917_1627493840.74/0/json-smart-2-4-7-sources-1-jar/net/minidev/json/writer/CollectionMapper.java
- * /opt/cola/permits/1183889917_1627493840.74/0/json-smart-2-4-7-sources-1-jar/net/minidev/json/writer/ArraysMapper.java
- * /opt/cola/permits/1183889917_1627493840.74/0/json-smart-2-4-7-sources-1-jar/net/minidev/json/JSONArray.java
- * /opt/cola/permits/1183889917_1627493840.74/0/json-smart-2-4-7-sources-1-jar/net/minidev/json/writer/FakeMapper.java
- * /opt/cola/permits/1183889917_1627493840.74/0/json-smart-2-4-7-sources-1-jar/net/minidev/json/parser/JSONParserBase.java

* /opt/cola/permits/1183889917_1627493840.74/0/json-smart-2-4-7-sources-1-jar/net/minidev/json/JSONAware.java
* /opt/cola/permits/1183889917_1627493840.74/0/json-smart-2-4-7-sources-1-jar/net/minidev/json/writer/JsonReader.java
* /opt/cola/permits/1183889917_1627493840.74/0/json-smart-2-4-7-sources-1-jar/net/minidev/json/JSONObject.java
* /opt/cola/permits/1183889917_1627493840.74/0/json-smart-2-4-7-sources-1-jar/net/minidev/json/parser/JSONParserInputStream.java
* /opt/cola/permits/1183889917_1627493840.74/0/json-smart-2-4-7-sources-1-jar/net/minidev/json/writer/DefaultMapperOrdered.java
* /opt/cola/permits/1183889917_1627493840.74/0/json-smart-2-4-7-sources-1-jar/net/minidev/json/parser/JSONParserReader.java
No license file was found, but licenses were detected in source scan.

<url><http://www.apache.org/licenses/LICENSE-2.0.txt></url>

Found in path(s):

* /opt/cola/permits/1183889917_1627493840.74/0/json-smart-2-4-7-sources-1-jar/META-INF/maven/net.minidev/json-smart/pom.xml

1.186 micronaut-flyway 5.4.1

1.186.1 Available under license :

Apache License

Version 2.0, January 2004

<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed

as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this

License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.187 kotlin-stdlib-common 1.6.21

1.187.1 Available under license :

Apache-2.0

1.188 jackson-module-scala 2.14.0

1.188.1 Available under license :

This copy of Jackson JSON processor Scala module is licensed under the Apache (Software) License, version 2.0 ("the License").

See the License for details about distribution rights, and the specific rights regarding derivative works.

You may obtain a copy of the License at:

<http://www.apache.org/licenses/LICENSE-2.0>

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes

of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You

meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor,

except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.189 netty-project 4.1.84.Final

1.189.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
# The Netty Project licenses this file to you under the Apache License,  
# version 2.0 (the "License"); you may not use this file except in compliance  
# with the License. You may obtain a copy of the License at:  
# distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
```

Found in path(s):

```
* /opt/cola/permits/1470281385_1668115980.505427/0/netty-codec-http-4-1-84-final-jar/META-INF/native-  
image/io.netty.codec-http/native-image.properties
```

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0

Implementation-Title: Netty/Codec/HTTP

Bundle-Description: Netty is an asynchronous event-driven network appl
ication framework for rapid development of maintainable high perfo
rmance protocol servers and clients.

Automatic-Module-Name: io.netty.codec.http

Bundle-License: <https://www.apache.org/licenses/LICENSE-2.0>

Bundle-SymbolicName: io.netty.codec-http

Implementation-Version: 4.1.84.Final

Built-By: chris

Bnd-LastModified: 1665536154725

Bundle-ManifestVersion: 2

Implementation-Vendor-Id: io.netty

Bundle-DocURL: <https://netty.io/>

Bundle-Vendor: The Netty Project

Import-Package: com.aayushatharva.brotli4j.encoder;resolution:=optiona

l,com.jcraft.jzlib;resolution:=optional,io.netty.buffer;version="[4.1

,5)",io.netty.channel;version="[4.1,5)",io.netty.channel.embedded;version="[4.1,5)",io.netty.handler.codec,io.netty.handler.codec.base64;version="[4.1,5)",io.netty.handler.codec.compression;version="[4.1,5)",io.netty.handler.ssl;version="[4.1,5)",io.netty.handler.stream;version="[4.1,5)",io.netty.util;version="[4.1,5)",io.netty.util.concurrent;version="[4.1,5)",io.netty.util.internal;version="[4.1,5)",io.netty.util.internal.logging;version="[4.1,5)",sun.nio.ch;resolution:=optional,org.eclipse.jetty.npn;version="[1,2)";resolution:=optional,org.eclipse.jetty.alpn;version="[1,2)";resolution:=optional
Require-Capability: osgi.ee;filter="(&(osgi.ee=JavaSE)(version=1.6))"
Tool: Bnd-2.4.1.201501161923
Implementation-Vendor: The Netty Project
Export-Package: io.netty.handler.codec.http;uses:="io.netty.buffer,io.netty.channel,io.netty.channel.embedded,io.netty.handler.codec,io.netty.handler.codec.compression,io.netty.handler.codec.http.cookie,io.netty.handler.stream,io.netty.util";version="4.1.84",io.netty.handler.codec.http.cookie;version="4.1.84",io.netty.handler.codec.http.cors;uses:="io.netty.channel,io.netty.handler.codec.http";version="4.1.84",io.netty.handler.codec.http.multipart;uses:="io.netty.buffer,io.netty.channel,io.netty.handler.codec,io.netty.handler.codec.http,io.netty.handler.stream,io.netty.util";version="4.1.84",io.netty.handler.codec.http.websocketx;uses:="io.netty.buffer,io.netty.channel,io.netty.handler.codec,io.netty.handler.codec.http,io.netty.handler.stream,io.netty.util,io.netty.util.internal.logging";version="4.1.84",io.netty.handler.codec.http.websocketx.extensions;uses:="io.netty.channel,io.netty.handler.codec,io.netty.handler.codec.http.websocketx";version="4.1.84",io.netty.handler.codec.http.websocketx.extensions.compression;uses:="io.netty.channel,io.netty.handler.codec.http.websocketx.extensions";version="4.1.84",io.netty.handler.codec.rtsp;uses:="io.netty.buffer,io.netty.channel,io.netty.handler.codec.http,io.netty.util";version="4.1.84",io.netty.handler.codec.spdy;uses:="io.netty.buffer,io.netty.channel,io.netty.handler.codec,io.netty.handler.codec.http,io.netty.util";version="4.1.84"
Bundle-Name: Netty/Codec/HTTP
Bundle-Version: 4.1.84.Final
Created-By: Apache Maven Bundle Plugin
Build-Jdk: 1.8.0_312
Implementation-URL: <https://netty.io/netty-codec-http/>

Found in path(s):

* /opt/cola/permits/1470281385_1668115980.505427/0/netty-codec-http-4-1-84-final-jar/META-INF/MANIFEST.MF

No license file was found, but licenses were detected in source scan.

<!--

~ Copyright 2012 The Netty Project

~

~ The Netty Project licenses this file to you under the Apache License,

~ version 2.0 (the "License"); you may not use this file except in compliance
~ with the License. You may obtain a copy of the License at:
~
~ <https://www.apache.org/licenses/LICENSE-2.0>
~
~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
~ License for the specific language governing permissions and limitations
~ under the License.
-->

Found in path(s):

* /opt/cola/permits/1470281385_1668115980.505427/0/netty-codec-http-4-1-84-final-jar/META-INF/maven/io.netty/netty-codec-http/pom.xml

1.190 scala-logging_2.12 3.9.4

1.190.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation

source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable

(except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and

may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify,

defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.191 micronaut-micrometer 4.6.1

1.191.1 Available under license :

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special,

incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.192 jackson-datatype-jdk8 2.14.0

1.192.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain

separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the

origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

This copy of Jackson JSON processor streaming parser/generator is licensed under the
Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the
specific rights regarding derivate works.

You may obtain a copy of the License at:

<http://www.apache.org/licenses/LICENSE-2.0>

1.193 junit-jupiter-junit-jupiter-api 5.7.1

1.193.1 Available under license :

Eclipse Public License - v 2.0

=====

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC
LICENSE (AGREEMENT). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM
CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. Definitions

Contribution means:

* **a)** in the case of the initial Contributor, the initial content Distributed under this Agreement, and

* **b)** in the case of each subsequent Contributor:

* **i)** changes to the Program, and

* **j)** additions to the Program;

where such changes and/or additions to the Program originate from and are Distributed by that particular
Contributor. A Contribution originates from a Contributor if it was added to the Program by such Contributor itself
or anyone acting on such Contributor's behalf. Contributions do not include changes or additions to the Program that
are not Modified Works.

Contributor means any person or entity that Distributes the Program.

Licensed Patents mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale

of its Contribution alone or when combined with the Program.

Program means the Contributions Distributed in accordance with this Agreement.

Recipient means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

Derivative Works shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

Modified Works shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

Distribute means the acts of ****a)**** distributing or ****b)**** making available in any manner that enables the transfer of a copy.

Source Code means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

Secondary License means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.

2. Grant of Rights

****a)**** Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

****b)**** Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

****c)**** Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

****d)**** Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any,

to grant the copyright license set forth in this Agreement.

****e)**** Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

3. Requirements

****3.1)**** If a Contributor Distributes the Program in any form, then:

****a)**** the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and

****b)**** the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:

****i)**** effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

****ii)**** effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

****iii)**** does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and

****iv)**** requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.

****3.2)**** When the Program is Distributed as Source Code:

****a)**** it must be made available under this Agreement, or if the Program ****i)**** is combined with other material in a separate file or files made available under a Secondary License, and ****ii)**** the initial Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available under the terms of such Secondary Licenses, and

****b)**** a copy of this Agreement must be included with each copy of the Program.

****3.3)**** Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability (notices) contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

4. Commercial Distribution

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor (Commercial Contributor) hereby agrees to defend and indemnify every other Contributor (Indemnified Contributor) against any losses, damages and costs (collectively Losses) arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial

product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: ****a)**** promptly notify the Commercial Contributor in writing of such claim, and ****b)**** allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. No Warranty

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN AS IS BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. Disclaimer of Liability

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. General

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of

such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.

Exhibit A - Form of Secondary Licenses Notice

> This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: {name license(s), version(s), and exceptions or additional permissions here }.

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Open Source Licenses

=====

This product may include a number of subcomponents with separate copyright notices and license terms. Your use of the source code for these subcomponents is subject to the terms and conditions of the subcomponent's license, as noted in the LICENSE-<subcomponent>.md files.

1.194 project-lombok 1.18.8

1.194.1 Available under license :

Copyright (C) 2009-2015 The Project Lombok Authors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.195 antlr 3.5.2

1.195.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*

* [The "BSD license"]

* Copyright (c) 2007-2008 Johannes Luber

* Copyright (c) 2005-2007 Kunle Odutola

* Copyright (c) 2011 Sam Harwell

* Copyright (c) 2011 Terence Parr

* All rights reserved.

*

* Redistribution and use in source and binary forms, with or without

* modification, are permitted provided that the following conditions

* are met:

* 1. Redistributions of source code must retain the above copyright

* notice, this list of conditions and the following disclaimer.

* 2. Redistributions in binary form must reproduce the above copyright

* notice, this list of conditions and the following disclaimer in the

* documentation and/or other materials provided with the distribution.

* 3. The name of the author may not be used to endorse or promote products

* derived from this software without specific prior written permission.

*

* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR

* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES

* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

```
@outputFile.imports() ::= <<  
<@super.imports()>
```

```
<if(!TREE_PARSER)>  
<! tree parser would already have imported !>  
using Antlr.Runtime.Tree;  
using RewriteRuleITokenStream = Antlr.Runtime.Tree.RewriteRuleTokenStream;  
<endif>  
>>
```

```
@genericParser.members() ::= <<  
<@super.members()>  
<parserMembers()>  
>>
```

```
parserCtorBody() ::= <%  
<super.parserCtorBody()><\n>  
TreeAdaptor =  
<if(actions.(actionScope).treeAdaptorInitializer)>  
<actions.(actionScope).treeAdaptorInitializer>  
<else>  
new <actions.(actionScope).treeAdaptorType; null="CommonTreeAdaptor">()  
<end>  
;  
>%>
```

/** Add an adaptor property that knows how to build trees */

```
parserMembers() ::= <<  
private <treeAdaptorType()> adaptor;  
  
public <treeAdaptorType()> TreeAdaptor  
{  
get  
{  
return adaptor;  
}  
  
set
```

```

{
  this.adaptor = value;
  <grammar.directDelegates:{g|<g:delegateName().TreeAdaptor = this.adaptor;}>
}
}
>>

treeAdaptorType() ::= <<
<actions.(actionScope).treeAdaptorType; null="ITreeAdaptor">
>>

ruleReturnBaseType() ::= <%
Ast<if(TREE_PARSER)>Tree<else>Parser<endif>RuleReturnScope\<<ASTLabelType>, <labelType>>
%>

/** Add a variable to track rule's return AST */
ruleDeclarations() ::= <<
<super.ruleDeclarations()>
<ASTLabelType> root_0 = default(<ASTLabelType>);<\n>
>>

ruleLabelDefs() ::= <<
<super.ruleLabelDefs()>
<[ruleDescriptor.tokenLabels,ruleDescriptor.wildcardTreeLabels,ruleDescriptor.wildcardTreeListLabels]
:{it|<ASTLabelType> <it.label.text>_tree = default(<ASTLabelType>);}; separator="\n">
<ruleDescriptor.tokenListLabels:{it|<ASTLabelType> <it.label.text>_tree = default(<ASTLabelType>);};
separator="\n">
<ruleDescriptor.allTokenRefsInAltsWithRewrites
:{it|RewriteRule<rewriteElementType>Stream stream_<it>=new
RewriteRule<rewriteElementType>Stream(adaptor,"token <it>");}; separator="\n">
<ruleDescriptor.allRuleRefsInAltsWithRewrites
:{it|RewriteRuleSubtreeStream stream_<it>=new RewriteRuleSubtreeStream(adaptor,"rule <it>");};
separator="\n">
>>

/** When doing auto AST construction, we must define some variables;
 * These should be turned off if doing rewrites. This must be a "mode"
 * as a rule could have both rewrite and AST within the same alternative
 * block.
 */
@alt.declarations() ::= <<
<if(autoAST)>
<if(outerAlt)>
<if(!rewriteMode)>
root_0 = (<ASTLabelType>)adaptor.Nil();
<endif>
<endif>
<endif>

```

```

>>

// Tracking Rule Elements

/** ID and track it for use in a rewrite rule */
tokenRefTrack(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRefBang(...)> <! Track implies no auto AST construction!>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) <endif>stream_<token>.Add(<label>);<\n>
>>

/** ids+=ID and track it for use in a rewrite rule; adds to ids *and*
 * to the tracking list stream_ID for use in the rewrite.
 */
tokenRefTrackAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRefTrack(...)>
<listLabelElem(elem=label,elemType=labelType,...)>
>>

/** ^(ID ...) track for rewrite */
tokenRefRuleRootTrack(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRefBang(...)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) <endif>stream_<token>.Add(<label>);
>>

/** Match ^(label+=TOKEN ...) track for rewrite */
tokenRefRuleRootTrackAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRefRuleRootTrack(...)>
<listLabelElem(elem=label,elemType=labelType,...)>
>>

/** rule when output=AST and tracking for rewrite */
ruleRefTrack(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) <endif>stream_<rule.name>.Add(<label>.Tree);
>>

/** x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefTrack(...)>
<listLabelElem(elem={<label>.Tree},elemType=ASTLabelType,...)>
>>

/** ^(rule ...) rewrite */
ruleRefRuleRootTrack(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) <endif>stream_<rule>.Add(<label>.Tree);
>>

```

```

/** ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRootTrack(...)>
<listLabelElem(elem={<label>.Tree},elemType=ASTLabelType,...)>
>>

// R e w r i t e

rewriteCode(
alts, description,
referencedElementsDeep, // ALL referenced elements to right of ->
referencedTokenLabels,
referencedTokenListLabels,
referencedRuleLabels,
referencedRuleListLabels,
referencedWildcardLabels,
referencedWildcardListLabels,
rewriteBlockLevel, enclosingTreeLevel, treeLevel) ::= <<
<\n>{
// AST REWRITE
// elements: <referencedElementsDeep; separator=", ">
// token labels: <referencedTokenLabels; separator=", ">
// rule labels: <referencedRuleLabels; separator=", ">
// token list labels: <referencedTokenListLabels; separator=", ">
// rule list labels: <referencedRuleListLabels; separator=", ">
// wildcard labels: <[referencedWildcardLabels,referencedWildcardListLabels]; separator=", ">
<if(backtracking)>
if (<actions.(actionScope).synpredgate>) {
<endif>
<prevRuleRootRef(>).Tree = root_0;
<rewriteCodeLabels(>
root_0 = (<ASTLabelType>)adaptor.Nil();
<alts:rewriteAlt(); separator="else ">
<! if tree parser and rewrite=true !>
<if(TREE_PARSER&&rewriteMode)>
<prevRuleRootRef(>).Tree = (<ASTLabelType>)adaptor.RulePostProcessing(root_0);
if (<prevRuleRootRef(>).Tree != null)
input.ReplaceChildren(adaptor.GetParent(retval.Start), adaptor.GetChildIndex(retval.Start),
adaptor.GetChildIndex(_last), retval.Tree);
<endif>
<! if parser or tree-parser && rewrite!=true, we need to set result !>
<if(!TREE_PARSER||!rewriteMode)>
<prevRuleRootRef(>).Tree = root_0;
<endif>
<endif>
<if(backtracking)>
}
<endif>
}

```

```

>>

rewriteCodeLabels() ::= <<
<referencedTokenLabels
  :{it|RewriteRule<rewriteElementType>Stream stream_<it>=new
RewriteRule<rewriteElementType>Stream(adaptor,"token <it>",<it>)};
  separator="\n"
>
<referencedTokenListLabels
  :{it|RewriteRule<rewriteElementType>Stream stream_<it>=new
RewriteRule<rewriteElementType>Stream(adaptor,"token <it> ", list_<it>)};
  separator="\n"
>
<referencedWildcardLabels
  :{it|RewriteRuleSubtreeStream stream_<it>=new RewriteRuleSubtreeStream(adaptor,"wildcard <it>",<it>)};
  separator="\n"
>
<referencedWildcardListLabels
  :{it|RewriteRuleSubtreeStream stream_<it>=new RewriteRuleSubtreeStream(adaptor,"wildcard <it> ",list_<it>)};
  separator="\n"
>
<referencedRuleLabels
  :{it|RewriteRuleSubtreeStream stream_<it>=new RewriteRuleSubtreeStream(adaptor,"rule
<it>",<it>!=null?<it>.Tree:null)};
  separator="\n"
>
<referencedRuleListLabels
  :{it|RewriteRuleSubtreeStream stream_<it>=new RewriteRuleSubtreeStream(adaptor,"token <it> ",list_<it>)};
  separator="\n"
>
>>

/** Generate code for an optional rewrite block; note it uses the deep ref'd element
 * list rather shallow like other blocks.
 */
rewriteOptionalBlock(
  alt,rewriteBlockLevel,
  referencedElementsDeep, // all nested refs
  referencedElements, // elements in immediately block; no nested blocks
  description) ::=
<<
// <fileName>:<description>
if (<referencedElementsDeep: {el | stream_<el>.HasNext}; separator="||">)
{
  <alt>
}
<referencedElementsDeep: {el | stream_<el>.Reset();<n>}>

```

```

>>

rewriteClosureBlock(
  alt,rewriteBlockLevel,
  referencedElementsDeep, // all nested refs
  referencedElements, // elements in immediately block; no nested blocks
  description) ::=
<<
// <fileName>:<description>
while ( <referencedElements:{el | stream_<el>.HasNext}; separator="||"> )
{
  <alt>
}
<referencedElements:{el | stream_<el>.Reset();<\n>}>
>>

```

```

rewritePositiveClosureBlock(
  alt,rewriteBlockLevel,
  referencedElementsDeep, // all nested refs
  referencedElements, // elements in immediately block; no nested blocks
  description) ::=
<<
if (!(<referencedElements:{el | stream_<el>.HasNext}; separator="||">))
{
  throw new RewriteEarlyExitException();
}
while ( <referencedElements:{el | stream_<el>.HasNext}; separator="||"> )
{
  <alt>
}
<referencedElements:{el | stream_<el>.Reset();<\n>}>
>>

```

```

rewriteAlt(a) ::= <<
// <a.description>
<if(a.pred)>
if (<a.pred>)
{
  <a.alt>
}
<else>
{
  <a.alt>
}
<endif>
>>

```

```

/** For empty rewrites: "r : ... -> ;" */

```

```

rewriteEmptyAlt() ::= "root_0 = null;"

rewriteTree(root,children,description,enclosingTreeLevel,treeLevel) ::= <<
// <fileName>:<description>
{
<ASTLabelType> root_<treeLevel> = (<ASTLabelType>)adaptor.Nil();
<root:rewriteElement()>
<children:rewriteElement()>
adaptor.AddChild(root_<enclosingTreeLevel>, root_<treeLevel>);
}<\n>
>>

rewriteElementList(elements) ::= "<elements:rewriteElement()>"

rewriteElement(e) ::= <%
<@pregen()>
DebugLocation(<e.line>, <e.pos>);<\n>
<e.el>
%>

/** Gen ID or ID[args] */
rewriteTokenRef(token,elementIndex,args,terminalOptions={}) ::= <<
adaptor.AddChild(root_<treeLevel>, <createRewriteNodeFromElement(...)>);<\n>
>>

/** Gen $label ... where defined via label=ID */
rewriteTokenLabelRef(label,elementIndex) ::= <<
adaptor.AddChild(root_<treeLevel>, stream_<label>.NextNode());<\n>
>>

/** Gen $label ... where defined via label+=ID */
rewriteTokenListLabelRef(label,elementIndex) ::= <<
adaptor.AddChild(root_<treeLevel>, stream_<label>.NextNode());<\n>
>>

/** Gen ^($label ...) */
rewriteTokenLabelRefRoot(label,elementIndex) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(stream_<label>.NextNode(), root_<treeLevel>);<\n>
>>

/** Gen ^($label ...) where label+=... */
rewriteTokenListLabelRefRoot ::= rewriteTokenLabelRefRoot

/** Gen ^(ID ...) or ^(ID[args] ...) */
rewriteTokenRefRoot(token,elementIndex,args,terminalOptions={}) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(<createRewriteNodeFromElement(...)>,
root_<treeLevel>);<\n>
>>

```

```

rewriteImaginaryTokenRef(args,token,elementIndex,terminalOptions={}) ::= <<
adaptor.AddChild(root_<treeLevel>, <createImaginaryNode(tokenType=token, ...)>);<\n>
>>

```

```

rewriteImaginaryTokenRefRoot(args,token,elementIndex,terminalOptions={}) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(<createImaginaryNode(tokenType=token, ...)>,
root_<treeLevel>);<\n>
>>

```

```

/** plain -> {foo} action */
rewriteAction(action) ::= <<
root_0 = <action>;<\n>
>>

```

```

/** What is the name of the previous value of this rule's root tree? This
* let's us refer to $rule to mean previous value. I am reusing the
* variable 'tree' sitting in retval struct to hold the value of root_0 right
* before I set it during rewrites. The assign will be to retval.tree.
*/
prevRuleRootRef() ::= "retval"

```

```

rewriteRuleRef(rule) ::= <<
adaptor.AddChild(root_<treeLevel>, stream_<rule>.NextTree());<\n>
>>

```

```

rewriteRuleRefRoot(rule) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(stream_<rule>.NextNode(), root_<treeLevel>);<\n>
>>

```

```

rewriteNodeAction(action) ::= <<
adaptor.AddChild(root_<treeLevel>, <action>);<\n>
>>

```

```

rewriteNodeActionRoot(action) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(<action>, root_<treeLevel>);<\n>
>>

```

```

/** Gen $ruleLabel ... where defined via ruleLabel=rule */
rewriteRuleLabelRef(label) ::= <<
adaptor.AddChild(root_<treeLevel>, stream_<label>.NextTree());<\n>
>>

```

```

/** Gen $ruleLabel ... where defined via ruleLabel+=rule */
rewriteRuleListLabelRef(label) ::= <<
adaptor.AddChild(root_<treeLevel>, stream_<label>.NextTree());<\n>
>>

```

```

/** Gen ^($ruleLabel ...) where ruleLabel=rule */
rewriteRuleLabelRefRoot(label) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(stream_<label>.NextNode(), root_<treeLevel>);<\n>
>>

```

```

/** Gen ^($ruleLabel ...) where ruleLabel+=rule */
rewriteRuleListLabelRefRoot(label) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(stream_<label>.NextNode(), root_<treeLevel>);<\n>
>>

```

```

rewriteWildcardLabelRef(label) ::= <<
adaptor.AddChild(root_<treeLevel>, stream_<label>.NextTree());<\n>
>>

```

```

createImaginaryNode(tokenType,args,terminalOptions={ }) ::= <%
<if(terminalOptions.node)>
<! new MethodNode(IDLabel, args) !>
new <terminalOptions.node><(tokenType)<if(args)>, <args; separator=", "><endif>
<else>
(<ASTLabelType>)adaptor.Create(<tokenType>, <args; separator=", "><if(!args)>"<tokenType>"<endif>)
<endif>
%>

```

```

createRewriteNodeFromElement(token,args,terminalOptions={ }) ::= <%
<if(terminalOptions.node)>
new <terminalOptions.node>(stream_<token>.NextToken()<if(args)>, <args; separator=", "><endif>)
<else>
<if(args)> <! must create new node from old !>
adaptor.Create(<token>, <args; separator=", ">)
<else>
stream_<token>.NextNode()
<endif>
<endif>
%>

```

Found in path(s):

```

* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-
jar/org/antlr/codegen/templates/CSharp2/AST.stg

```

No license file was found, but licenses were detected in source scan.

[The "BSD license"]

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright

notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

Found in path(s):

* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/Python3/Dbg.stg

* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/Python/AST.stg

* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/Python/Dbg.stg

* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/Python3/AST.stg

* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/Java/ASTParser.stg

* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/tool/serialize.g

* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/ActionScript/ASTParser.stg

No license file was found, but licenses were detected in source scan.

/*

[The "BSD license"]

Copyright (c) 2010 Matthew Lloyd

<http://linkedin.com/in/matthewl>

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF

THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```
*/
scalaTypeInitMap ::= [
  "Int":"0",
  "Long":"0",
  "Float":"0.0f",
  "Double":"0.0",
  "Boolean":"false",
  "Byte":"0",
  "Short":"0",
  "Char":"0",
  default:"null" // anything other than an atomic type
]

/** The overall file structure of a recognizer; stores methods for rules
 * and cyclic DFAs plus support code.
 */
outputFile(LEXER,PARSER,TREE_PARSER, actionScope, actions,
  docComment, recognizer,
  name, tokens, tokenNames, rules, cyclicDFAs,
  bitsets, buildTemplate, buildAST, rewriteMode, profile,
  backtracking, synpreeds, memoize, numRules,
  fileName, ANTLRVersion, generatedTimestamp, trace,
  scopes, superClass, literals) ::=
<<
// $ANTLR <ANTLRVersion> <fileName> <generatedTimestamp>
<actions.(actionScope).header>

<@imports>
import org.antlr.runtime._
<if(TREE_PARSER)>
import org.antlr.runtime.tree._
<endif>
<@end>

<docComment>
<recognizer>
>>

lexer(grammar, name, tokens, scopes, rules, numRules, filterMode, labelType="CommonToken",
  superClass="Lexer") ::= <<
object <grammar.recognizerName> {
  <tokens:{it | val <it.name> = <it.type>} ; separator="\n">

  <cyclicDFAs:cyclicDFA()> <! dump tables for all DFA !>
}

class <grammar.recognizerName>(input: CharStream, state<grammar.delegators:{g|, <g.recognizerName>
```

```

<g:delegateName(>}>: RecognizerSharedState) extends <@superClassName><superClass><@end>(input,
state<grammar.delegators:{g|, <g.recognizerName> <g:delegateName(>}>}) {
  import <grammar.recognizerName>._
  <actions.lexer.members>

  // delegates
  <grammar.delegates:
    {g|<g.recognizerName> <g:delegateName(>}; separator="\n">
  // delegators
  <grammar.delegators:
    {g|<g.recognizerName> <g:delegateName(>}; separator="\n">
  <last(grammar.delegators):{g|public <g.recognizerName> gParent;}>

  <scopes:{it | <if(it.isDynamicGlobalScope)><globalAttributeScope(><endif>}>

  def this(input<grammar.delegators:{g|, <g.recognizerName> <g:delegateName(>}>: CharStream) =
    this(input, new RecognizerSharedState(<grammar.delegators:{g|, <g:delegateName(>}>))

  <if(memoize)>
  <if(grammar.grammarIsRoot)>
    state.ruleMemo = new Array[java.util.Map[_,_]](<numRules>+1)<\n><! index from 1..n !>
  <endif>
  <endif>
    <grammar.directDelegates:
      {g|<g:delegateName(> = new <g.recognizerName>(input, state<trunc(g.delegators):{p|,
  <p:delegateName(>}>, this); separator="\n">
    <grammar.delegators:
      {g|this.<g:delegateName(> = <g:delegateName(>}; separator="\n">
    <last(grammar.delegators):{g|gParent = <g:delegateName(>}>

  override def getGrammarFileName = "<fileName>"

  <if(filterMode)>
    <filteringNextToken(>
  <endif>
    <rules; separator="\n\n">

    <synpreds:{p | <lexerSynpred(p)>}>
    <cyclicDFAs:{dfa | private val dfa<dfa.decisionNumber> = new
  <grammar.recognizerName>.DFA<dfa.decisionNumber>(this); separator="\n">
  }
  >>

  /** A override of Lexer.nextToken() that backtracks over mTokens() looking
  * for matches. No error can be generated upon error; just rewind, consume
  * a token and then try again. backtracking needs to be set as well.
  * Make rule memoization happen only at levels above 1 as we start mTokens
  * at backtracking==1.

```

```

*/
filteringNextToken() ::= <<
override def nextToken(): Token = {
  while (true) {
    if ( input.LA(1)==CharStream.EOF ) {
      var eof: Token = new CommonToken((CharStream)input,Token.EOF,
        Token.DEFAULT_CHANNEL,
        input.index(),input.index())
      eof.setLine(getLine())
      eof.setCharPositionInLine(getCharPositionInLine())
      return eof
    }
    state.token = null
state.channel = Token.DEFAULT_CHANNEL
    state.tokenStartCharIndex = input.index()
    state.tokenStartCharPositionInLine = input.getCharPositionInLine()
    state.tokenStartLine = input.getLine()
state.text = null
    try {
      val m = input.mark()
      state.backtracking=1 <! means we won't throw slow exception !>
      state.failed=false
      mTokens()
      state.backtracking=0
      <! mTokens backtracks with synpred at backtracking==2
        and we set the synpredgate to allow actions at level 1. !>
      if ( state.failed ) {
        input.rewind(m)
        input.consume() <! advance one char and try again !>
      }
      else {
        emit()
        return state.token
      }
    }
    catch {
      case re: RecognitionException =>
        // shouldn't happen in backtracking mode, but...
        reportError(re)
        recover(re)
    }
  }
}
}

override def memoize(input: IntStream,
  ruleIndex: Int,
  ruleStartIndex: Int) = {
  if ( state.backtracking>1 ) super.memoize(input, ruleIndex, ruleStartIndex)
}

```

```

}

override def alreadyParsedRule(input: IntStream, ruleIndex: Int): Boolean {
if ( state.backtracking>1 ) return super.alreadyParsedRule(input, ruleIndex)
return false
}
>>

actionGate() ::= "state.backtracking==0"

filteringActionGate() ::= "state.backtracking==1"

/** How to generate a parser */
genericParser(grammar, name, scopes, tokens, tokenNames, rules, numRules,
    bitsets, inputStreamType, superClass,
    labelType, members, rewriteElementType,
    filterMode, ASTLabelType="Object") ::= <<
object <grammar.recognizerName> {
<if(grammar.grammarIsRoot)>
    val tokenNames = Array(
        "\<invalid>", "\<EOR>", "\<DOWN>", "\<UP>", <tokenNames; separator=", ">
    )<\n>
<endif>

    <tokens:{it | val <it.name> = <it.type>}; separator="\n">

    <cyclicDFAs:cyclicDFA(> <! dump tables for all DFA !>

    <bitsets:{it | <bitset(name={FOLLOW_<it.name>_in_<it.inName><it.tokenIndex>},
        words64=it.bits)>}>
}

class <grammar.recognizerName>(input: <inputStreamType>, state<grammar.delegators:{ g|, <g.recognizerName>
<g:delegateName(>>}: RecognizerSharedState) extends <@superClassName><superClass><@end>(input, state) {
import <grammar.recognizerName>._
// delegates
<grammar.delegates:
    {g|public <g.recognizerName> <g:delegateName(>}; separator="\n">
// delegators
<grammar.delegators:
    {g|public <g.recognizerName> <g:delegateName(>}; separator="\n">
<last(grammar.delegators):{g|public <g.recognizerName> gParent;}>

<scopes:{it | <if(it.isDynamicGlobalScope)><globalAttributeScope(><endif>}>

<@members>
<! WARNING. bug in ST: this is cut-n-paste into Dbg.stg !>
def this(input<grammar.delegators:{g|, <g.recognizerName> <g:delegateName(>}: <inputStreamType>) =

```

```

this(input, new RecognizerSharedState(<grammar.delegates:{g|, <g:delegateName()>>))

<parserCtorBody()>
<grammar.directDelegates:
  {g|<g:delegateName()> = new <g.recognizerName>(input, state<trunc(g.delegates):{p|,
<p:delegateName()>>, this)}; separator="\n">
  <grammar.indirectDelegates:{g | <g:delegateName()> = <g.delegate:delegateName()>.<g:delegateName()>};
separator="\n">
  <last(grammar.delegates):{g|gParent = <g:delegateName()>}>
<@end>

override def getTokenNames: Array[String] = tokenNames
override def getGrammarFileName = "<fileName>"

<members>

<rules; separator="\n\n">

<! generate rule/method definitions for imported rules so they
appear to be defined in this recognizer. !>
// Delegated rules
<grammar.delegatedRules:{ruleDescriptor|
  @throws(classOf[RecognitionException])
  def <ruleDescriptor.name>(<ruleDescriptor.parameterScope:parameterScope()>): <returnType()> = \{
<if(ruleDescriptor.hasReturnValue)>return
<endif><ruleDescriptor.grammar:delegateName()>.<ruleDescriptor.name>(<ruleDescriptor.parameterScope.attribute:
es:{a|<a.name>}; separator=", ">) \}}; separator="\n">

  <synpreds:{p | <synpred(p)>}>

  <cyclicDFAs:{dfa | private val dfa<dfa.decisionNumber> = new
<grammar.recognizerName>.DFA<dfa.decisionNumber>(this)}; separator="\n">
}
>>

parserCtorBody() ::= <<
<if(memoize)>
<if(grammar.grammarIsRoot)>
this.state.ruleMemo = new Array[java.util.Map[_,_]](<length(grammar.allImportedRules)>+1)<\n> <! index from
1..n !>
<endif>
<endif>
<grammar.delegates:
{g|this.<g:delegateName()> = <g:delegateName()>}; separator="\n">
>>

parser(grammar, name, scopes, tokens, tokenNames, rules, numRules, bitsets,
  ASTLabelType="Object", superClass="Parser", labelType="Token",

```

```

    members={ <actions.parser.members> } ) ::= <<
<genericParser(inputStreamType="TokenStream", rewriteElementType="Token", ...)>
>>

/** How to generate a tree parser; same as parser except the input
 * stream is a different type.
 */
treeParser(grammar, name, scopes, tokens, tokenNames, globalAction, rules,
    numRules, bitsets, filterMode, labelType={ <ASTLabelType> }, ASTLabelType="Object",
superClass={ <if(filterMode)><if(buildAST)>TreeRewriter<else>TreeFilter<endif><else>TreeParser<endif> },
members={ <actions.treeparser.members> }
    ) ::= <<
<genericParser(inputStreamType="TreeNodeStream", rewriteElementType="Node", ...)>
>>

/** A simpler version of a rule template that is specific to the imaginary
 * rules created for syntactic predicates. As they never have return values
 * nor parameters etc..., just give simplest possible method. Don't do
 * any of the normal memoization stuff in here either; it's a waste.
 * As predicates cannot be inlined into the invoking rule, they need to
 * be in a rule by themselves.
 */
synpredRule(ruleName, ruleDescriptor, block, description, nakedBlock) ::=
<<
// $ANTLR start <ruleName>
@throws(classOf[RecognitionException])
def <ruleName>_fragment(<ruleDescriptor.parameterScope:parameterScope()>): Unit = {
    <ruleLabelDefs()>
<if(trace)>
    traceIn("<ruleName>_fragment", <ruleDescriptor.index>)
    try {
        <block>
    }
    finally {
        traceOut("<ruleName>_fragment", <ruleDescriptor.index>);
    }
<else>
    <block>
<endif>
}
// $ANTLR end <ruleName>
>>

synpred(name) ::= <<
final def <name>(): Boolean = {
    state.backtracking+=1
    <@start()>
    val start = input.mark()

```

```

try {
  <name>_fragment() // can never throw exception
} catch {
  case re: RecognitionException =>
    System.err.println("impossible: "+re)
}
val success = !state.failed
input.rewind(start)
<@stop()>
state.backtracking-=1
state.failed=false
success
}<\n>
>>

lexerSynpred(name) ::= <<
<synpred(name)>
>>

ruleMemoization(name) ::= <<
<if(memoize)>
if ( state.backtracking>0 && alreadyParsedRule(input, <ruleDescriptor.index> ) ) { return <ruleReturnValue()> }
<endif>
>>

/** How to test for failure and return from rule */
checkRuleBacktrackFailure() ::= <<
<if(backtracking)>if (state.failed) return <ruleReturnValue()><endif>
>>

/** This rule has failed, exit indicating failure during backtrack */
ruleBacktrackFailure() ::= <<
<if(backtracking)>if (state.backtracking>0) { state.failed=true; return <ruleReturnValue()>}<endif>
>>

/** How to generate code for a rule. This includes any return type
 * data aggregates required for multiple return values.
 */
rule(ruleName,ruleDescriptor,block,emptyRule,description,exceptions,finally,memoize) ::= <<
<ruleAttributeScope(scope=ruleDescriptor.ruleScope)>
<returnScope(scope=ruleDescriptor.returnScope)>

// $ANTLR start "<ruleName>"
// <fileName>:<description>
@throws(classOf[RecognitionException])
final def <ruleName>( <ruleDescriptor.parameterScope:parameterScope()>): <returnType()> = {
  <if(trace)>traceIn("<ruleName>", <ruleDescriptor.index>)<endif>
  <ruleScopeSetUp()>

```

```

<ruleDeclarations()>
<ruleLabelDefs()>
<ruleDescriptor.actions.init>
<@preamble()>
try {
  <ruleMemoization(name=ruleName)>
  <block>
  <ruleCleanup()>
  <(ruleDescriptor.actions.after):execAction()>
}
<if(exceptions)>
  <exceptions:{e|<catch(decl=e.decl,action=e.action)><\n>}>
<else>
<if(!emptyRule)>
<if(actions.(actionScope).rulecatch)>
  <actions.(actionScope).rulecatch>
<else>
  catch {
    case re: RecognitionException =>
      reportError(re)
      recover(input,re)
  }
  <@setErrorReturnValue()>
  }<\n>
<endif>
<endif>
<endif>
  finally {
    <if(trace)>traceOut("<ruleName>", <ruleDescriptor.index>);<endif>
    <memoize()>
    <ruleScopeCleanup()>
    <finally>
  }
  <@postamble()>
  return <ruleReturnValue()>
}
// $ANTLR end "<ruleName>"
>>

```

```

catch(decl,action) ::= <<
catch (<e.decl>) {
  <e.action>
}
>>

```

```

ruleDeclarations() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
val retval = new <returnType()>()
retval.start = input.LT(1)<\n>

```

```

<else>
<ruleDescriptor.returnScope.attributes:{ a |
var <a.name>: <a.type> = <if(a.initValue)><a.initValue><else><initValue(a.type)><endif>
}>
<endif>
<if(memoize)>
val <ruleDescriptor.name>_startIndex = input.index()
<endif>
>>

ruleScopeSetUp() ::= <<
<ruleDescriptor.useScopes:{it | <it>_stack.push(new <it>_scope()); separator="\n">
<ruleDescriptor.ruleScope:{it | <it.name>_stack.push(new <it.name>_scope()); separator="\n">
>>

ruleScopeCleanUp() ::= <<
<ruleDescriptor.useScopes:{it | <it>_stack.pop(); separator="\n">
<ruleDescriptor.ruleScope:{it | <it.name>_stack.pop(); separator="\n">
>>

ruleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,ruleDescriptor.tokenListLabels,
ruleDescriptor.wildcardTreeLabels,ruleDescriptor.wildcardTreeListLabels]
: {it | var <it.label.text>: <labelType> = null}; separator="\n"
>
<[ruleDescriptor.tokenListLabels,ruleDescriptor.ruleListLabels,ruleDescriptor.wildcardTreeListLabels]
: {it | var list_<it.label.text>: java.util.List=null}; separator="\n"
>
<ruleDescriptor.ruleLabels:ruleLabelDef(); separator="\n">
<ruleDescriptor.ruleListLabels:{ll|var <ll.label.text>: RuleReturnScope = null}; separator="\n">
>>

lexerRuleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,
ruleDescriptor.tokenListLabels,
ruleDescriptor.ruleLabels]
: {it | var <it.label.text>: <labelType>=null}; separator="\n"
>
<ruleDescriptor.charLabels:{it | int <it.label.text>;}; separator="\n">
<[ruleDescriptor.tokenListLabels,
ruleDescriptor.ruleListLabels]
: {it | var list_<it.label.text>: java.util.List=null}; separator="\n"
>
>>

ruleReturnValue() ::= <<
<if(!ruleDescriptor.isSynPred)>

```

```

<if(ruleDescriptor.hasReturnValue)>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnName>
<else>
retval
<endif>
<endif>
<endif>
>>

ruleCleanUp() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(!TREE_PARSER)>
retval.stop = input.LT(-1)<\n>
<endif>
<endif>
>>

memoize() ::= <<
<if(memoize)>
<if(backtracking)>
if ( state.backtracking>0 ) { memoize(input, <ruleDescriptor.index>, <ruleDescriptor.name>_StartIndex) }
<endif>
<endif>
>>

/** How to generate a rule in the lexer; naked blocks are used for
 * fragment rules.
 */
lexerRule(ruleName,nakedBlock,ruleDescriptor,block,memoize) ::= <<
// $ANTLR start "<ruleName>"
@throws(classOf[RecognitionException])
final def m<ruleName>(<ruleDescriptor.parameterScope:parameterScope()): Unit = {
  <if(trace)>traceIn("<ruleName>", <ruleDescriptor.index>)<endif>
  <ruleScopeSetUp()>
  <ruleDeclarations()>
  try {
<if(nakedBlock)>
  <ruleMemoization(name=ruleName)>
  <lexerRuleLabelDefs()>
  <ruleDescriptor.actions.init>
  try <block><\n>
<else>
  var _type = <ruleName>
  var _channel = BaseRecognizer.DEFAULT_TOKEN_CHANNEL
  <ruleMemoization(name=ruleName)>
  <lexerRuleLabelDefs()>
  <ruleDescriptor.actions.init>

```

```

    try <block>
    <ruleCleanup()>
    state.`type` = _type
    state.channel = _channel
    <(ruleDescriptor.actions.after):execAction()>
<endif>
}
finally {
    <if(trace)>traceOut("<ruleName>", <ruleDescriptor.index>)<endif>
    <ruleScopeCleanup()>
    <memoize()>
}
}
// $ANTLR end "<ruleName>"
>>

```

```

/** How to generate code for the implicitly-defined lexer grammar rule
 * that chooses between lexer rules.
 */

```

```

tokensRule(ruleName,nakedBlock,args,block,ruleDescriptor) ::= <<
@throws(classOf[RecognitionException])
def mTokens(): Unit = {
    <block><\n>
}
>>

```

```

// S U B R U L E S

```

```

/** A (...) subrule with multiple alternatives */

```

```

block(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
// <fileName>:<description>
var alt<decisionNumber> = <maxAlt>
<decls>
<@predecision()>
<decision>
<@postdecision()>
<@prebranch()>
alt<decisionNumber> match {
    <alts:{ a | <altSwitchCase(i,a)> }>
    case _ =>
}
<@postbranch()>
>>

```

```

/** A rule block with multiple alternatives */

```

```

ruleBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
// <fileName>:<description>
var alt<decisionNumber> = <maxAlt>

```

```

<decls>
<@predecision()>
<decision>
<@postdecision()>
alt<decisionNumber> match {
  <alts:{a | <altSwitchCase(i,a)>}>
  case _ =>
}
>>

ruleBlockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
// <fileName>:<description>
<decls>
<@prealt()>
<alts>
<@postalt()>
>>

/** A special case of a (...) subrule with a single alternative */
blockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
// <fileName>:<description>
<decls>
<@prealt()>
<alts>
<@postalt()>
>>

/** A (..)+ block with 1 or more alternatives */
positiveClosureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
// <fileName>:<description>
var cnt<decisionNumber>: Int = 0
<decls>
<@preloop()>
var loop<decisionNumber>_quitflag = false
while (!loop<decisionNumber>_quitflag) {
  var alt<decisionNumber>:Int = <maxAlt>
  <@predecision()>
  <decision>
  <@postdecision()>
  alt<decisionNumber> match {
    <alts:{a | <altSwitchCase(i,a)>}>
  case _ =>
    if ( cnt<decisionNumber> >= 1 ) loop<decisionNumber>_quitflag = true
    else {
      <ruleBacktrackFailure()>
      val eee = new EarlyExitException(<decisionNumber>, input)
      <@earlyExitException()>
    }
  }
  cnt<decisionNumber> += 1
}
>>

```

```

        throw eee
    }
}
cnt<decisionNumber>+=1
}
<@postloop()>
>>

```

```

positiveClosureBlockSingleAlt ::= positiveClosureBlock

```

```

/** A (.)* block with 1 or more alternatives */

```

```

closureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::=
<<
// <fileName>:<description>
<decls>
<@preloop()>
var loop<decisionNumber>_quitflag = false
while (!loop<decisionNumber>_quitflag) {
    var alt<decisionNumber>:Int = <maxAlt>
    <@predecision()>
    <decision>
    <@postdecision()>
    alt<decisionNumber> match {
        <alts:{a | <altSwitchCase(i,a)>}>
    case _ => loop<decisionNumber>_quitflag = true
    }
}
<@postloop()>
>>

```

```

closureBlockSingleAlt ::= closureBlock

```

```

/** Optional blocks (x)? are translated to (x|) by before code generation
 * so we can just use the normal block template
 */

```

```

optionalBlock ::= block

```

```

optionalBlockSingleAlt ::= block

```

```

/** A case in a switch that jumps to an alternative given the alternative
 * number. A DFA predicts the alternative and then a simple switch
 * does the jump to the code that actually matches that alternative.
 */

```

```

altSwitchCase(altNum, alt) ::= <<
case <altNum> =>
    <@prealt()>
    <alt>
>>

```

```

/** An alternative is just a list of elements; at outermost level */
alt(elements,altNum,description,autoAST,outerAlt,treeLevel,rew) ::= <<
// <fileName>:<description>
{
<@declarations()>
<elements:element()>
<rew>
<@cleanup()>
}
>>

/** What to emit when there is no rewrite. For auto build
* mode, does nothing.
*/
noRewrite(rewriteBlockLevel, treeLevel) ::= ""

// E L E M E N T S

/** Dump the elements one per line */
element(e) ::= <<
<@prematch()>
<e.el><\n>
>>

/** match a token optionally with a label in front */
tokenRef(token,label,elementIndex,terminalOptions) ::= <<
<if(label)><label>=<endif>`match`(input,<token>,FOLLOW_<token>_in_<ruleName><elementIndex>)<if(label)>
.asInstanceOf[<labelType>]<endif>
<checkRuleBacktrackFailure()>
>>

/** ids+=ID */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

listLabel(label,elem) ::= <<
if (list_<label>==null) list_<label>=new java.util.ArrayList()
list_<label>.add(<elem>)<\n>
>>

/** match a character */
charRef(char,label) ::= <<
<if(label)>
<label> = input.LA(1)<\n>
<endif>

```

```

`match`(<char>)
<checkRuleBacktrackFailure()>
>>

/** match a character range */
charRangeRef(a,b,label) ::= <<
<if(label)>
<label> = input.LA(1)<\n>
<endif>
matchRange(<a>,<b>); <checkRuleBacktrackFailure()>
>>

/** For now, sets are interval tests and must be tested inline */
matchSet(s,label,elementIndex,terminalOptions,postmatchCode="") ::= <<
<if(label)>
<if(LEXER)>
<label>= input.LA(1)<\n>
<else>
<label>=input.LT(1).asInstanceOf[<labelType>]<\n>
<endif>
<endif>
if ( <s> ) {
    input.consume()
    <postmatchCode>
<if(!LEXER)>
    state.errorRecovery=false<\n>
<endif>
    <if(backtracking)>state.failed=false<endif>
}
else {
    <ruleBacktrackFailure()>
    val mse = new MismatchedSetException(null,input)
    <@mismatchedSetException()>
<if(LEXER)>
    recover(mse)
    throw mse
<else>
    throw mse
    <! use following code to make it recover inline; remove throw mse;
    recoverFromMismatchedSet(input,mse,FOLLOW_set_in_<ruleName><elementIndex>)
    !>
<endif>
}<\n>
>>

matchRuleBlockSet ::= matchSet

matchSetAndListLabel(s,label,elementIndex,postmatchCode) ::= <<

```

```

<matchSet(...)>
<listLabel(elem=label,...)>
>>

/** Match a string literal */
lexerStringRef(string,label,elementIndex="0") ::= <<
<if(label)>
val <label>Start = getCharIndex()
`match`(<string>)
<checkRuleBacktrackFailure()>
val <label>StartLine<elementIndex> = getLine()
val <label>StartCharPos<elementIndex> = getCharPositionInLine()
<label> = new <labelType>(input, Token.INVALID_TOKEN_TYPE, Token.DEFAULT_CHANNEL, <label>Start,
getCharIndex()-1)
<label>.setLine(<label>StartLine<elementIndex>)
<label>.setCharPositionInLine(<label>StartCharPos<elementIndex>)
<else>
`match`(<string>)
<checkRuleBacktrackFailure()><\n>
<endif>
>>

wildcard(token,label,elementIndex,terminalOptions) ::= <<
<if(label)>
<label>=input.LT(1).asInstanceOf[<labelType>]<\n>
<endif>
matchAny(input)
<checkRuleBacktrackFailure()>
>>

wildcardAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<wildcard(...)>
<listLabel(elem=label,...)>
>>

/** Match . wildcard in lexer */
wildcardChar(label, elementIndex) ::= <<
<if(label)>
<label> = input.LA(1)<\n>
<endif>
matchAny()
<checkRuleBacktrackFailure()>
>>

wildcardCharListLabel(label, elementIndex) ::= <<
<wildcardChar(...)>
<listLabel(elem=label,...)>
>>

```

```

/** Match a rule reference by invoking it possibly with arguments
 * and a return value or values. The 'rule' argument was the
 * target rule name, but now is type Rule, whose toString is
 * same: the rule name. Now though you can access full rule
 * descriptor stuff.
 */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
pushFollow(FOLLOW_<rule.name>_in_<ruleName><elementIndex>)
<if(label)><label>=<endif><if(scope)><scope.delegateName()>.<endif><rule.name><args; separator=", "><\n>
state._fsp-=1
<checkRuleBacktrackFailure()>
>>

/** ids+=r */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(elem=label,...)>
>>

/** A lexer rule reference.
 *
 * The 'rule' argument was the target rule name, but now
 * is type Rule, whose toString is same: the rule name.
 * Now though you can access full rule descriptor stuff.
 */
lexerRuleRef(rule,label,args,elementIndex,scope) ::= <<
<if(label)>
val <label>Start<elementIndex> = getCharIndex()
val <label>StartLine<elementIndex> = getLine()
val <label>StartCharPos<elementIndex> = getCharPositionInLine()
<if(scope)><scope.delegateName()>.<endif>m<rule.name><args; separator=", ">
<checkRuleBacktrackFailure()>
<label> = new <labelType>(input, Token.INVALID_TOKEN_TYPE, Token.DEFAULT_CHANNEL,
<label>Start<elementIndex>, getCharIndex()-1)
<label>.setLine(<label>StartLine<elementIndex>)
<label>.setCharPositionInLine(<label>StartCharPos<elementIndex>)
<else>
<if(scope)><scope.delegateName()>.<endif>m<rule.name><args; separator=", ">
<checkRuleBacktrackFailure()>
<endif>
>>

/** i+=INT in lexer */
lexerRuleRefAndListLabel(rule,label,args,elementIndex,scope) ::= <<
<lexerRuleRef(...)>
<listLabel(elem=label,...)>
>>

```

```

/** EOF in the lexer */
lexerMatchEOF(label,elementIndex) ::= <<
<if(label)>
val <label>Start<elementIndex> = getCharIndex()
val <label>StartLine<elementIndex> = getLine()
val <label>StartCharPos<elementIndex> = getCharPositionInLine()
`match`(EOF)
<checkRuleBacktrackFailure()>
val <label> = new <labelType>(input, EOF, Token.DEFAULT_CHANNEL, <label>Start<elementIndex>,
getCharIndex()-1)
<label>.setLine(<label>StartLine<elementIndex>)
<label>.setCharPositionInLine(<label>StartCharPos<elementIndex>)
<else>
`match`(EOF)
<checkRuleBacktrackFailure()>
<endif>
>>

// used for left-recursive rules
recRuleDefArg()          ::= "int <recRuleArg()>"
recRuleArg()             ::= "_p"
recRuleAltPredicate(ruleName,opPrec) ::= "<recRuleArg()> |<= <opPrec>"
recRuleSetResultAction() ::= "root_0=$<ruleName>_primary.tree;"
recRuleSetReturnAction(src,name)    ::= "$<name>=$<src>.<name>;"

/** match ^(root children) in tree parser */
tree(root, actionsAfterRoot, children, nullableChildList,
enclosingTreeLevel, treeLevel) ::= <<
<root:element()>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if ( input.LA(1)==Token.DOWN ) {
`match`(input, Token.DOWN, null)
<checkRuleBacktrackFailure()>
<children:element()>
`match`(input, Token.UP, null)
<checkRuleBacktrackFailure()>
}
<else>
`match`(input, Token.DOWN, null)
<checkRuleBacktrackFailure()>
<children:element()>
`match`(input, Token.UP, null)
<checkRuleBacktrackFailure()>
<endif>
>>

```

```

/** Every predicate is used as a validating predicate (even when it is
 * also hoisted into a prediction expression).
 */
validateSemanticPredicate(pred,description) ::= <<
if ( !(<evalPredicate(...)> ) ) {
    <ruleBacktrackFailure()>
    throw new FailedPredicateException(input, "<ruleName>", "<description>")
}
>>

// F i x e d D F A (if-then-else)

dfaState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
val LA<decisionNumber>_<stateNumber> = input.LA(<k>)<\n>
<edges; separator="\nelse ">
else {
<if(eotPredictsAlt)>
    alt<decisionNumber>=<eotPredictsAlt>
<else>
    <ruleBacktrackFailure()>
    val nvae = new NoViableAltException("<description>", <decisionNumber>, <stateNumber>, input)<\n>
    <@noViableAltException()>
    throw nvae<\n>
<endif>
}
>>

/** Same as a normal DFA state except that we don't examine lookahead
 * for the bypass alternative. It delays error detection but this
 * is faster, smaller, and more what people expect. For (X)? people
 * expect "if ( LA(1)==X ) match(X);" and that's it.
 */
dfaOptionalBlockState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
val LA<decisionNumber>_<stateNumber> = input.LA(<k>)<\n>
<edges; separator="\nelse ">
>>

/** A DFA state that is actually the loopback decision of a closure
 * loop. If end-of-token (EOT) predicts any of the targets then it
 * should act like a default clause (i.e., no error can be generated).
 * This is used only in the lexer so that for ('a')* on the end of a rule
 * anything other than 'a' predicts exiting.
 */
dfaLoopbackState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
val LA<decisionNumber>_<stateNumber> = input.LA(<k>)<\n>
<edges; separator="\nelse "><\n>
<if(eotPredictsAlt)>
<if(!edges)>

```

```

alt<decisionNumber>=<eotPredictsAlt> <! if no edges, don't gen ELSE !>
<else>
else {
  alt<decisionNumber>=<eotPredictsAlt>
}<\n>
<endif>
<endif>
>>

/** An accept state indicates a unique alternative has been predicted */
dfaAcceptState(alt) ::= "alt<decisionNumber>=<alt>"

/** A simple edge with an expression. If the expression is satisfied,
 * enter to the target state. To handle gated productions, we may
 * have to evaluate some predicates for this edge.
 */
dfaEdge(labelExpr, targetState, predicates) ::= <<
if ( (<labelExpr> <if(predicates)>&& (<predicates>)<endif>) {
  <targetState>
}
>>

// F i x e d D F A (switch case)

/** A DFA state where a SWITCH may be generated. The code generator
 * decides if this is possible: CodeGenerator.canGenerateSwitch().
 */
dfaStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
input.LA(<k>) match {
<edges; separator="\n">
case _ =>
<if(eotPredictsAlt)>
  alt<decisionNumber>=<eotPredictsAlt>
<else>
  <ruleBacktrackFailure()>
  val nvae = new NoViableAltException("<description>", <decisionNumber>, <stateNumber>, input)<\n>
  <@noViableAltException()>
  throw nvae<\n>
<endif>
}<\n>
>>

dfaOptionalBlockStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
input.LA(<k>) match {
  <edges; separator="\n">
  case _ =>
}<\n>
>>

```

```

dfaLoopbackStateSwitch(k, edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
input.LA(<k>) match {
<edges; separator="\n"><\n>
case _ =>
<if(eotPredictsAlt)>
  alt<decisionNumber>=<eotPredictsAlt>;
<endif>
}<\n>
>>

```

```

dfaEdgeSwitch(labels, targetState) ::= <<
case <labels:{it | <it>}>; separator=" | ">=>
{
  <targetState>
}
>>

```

```
// C y c l i c D F A
```

```

/** The code to initiate execution of a cyclic DFA; this is used
 * in the rule to predict an alt just like the fixed DFA case.
 * The <name> attribute is inherited via the parser, lexer, ...
 */

```

```

dfaDecision(decisionNumber,description) ::= <<
alt<decisionNumber> = dfa<decisionNumber>.predict(input)
>>

```

```

/* Dump DFA tables as run-length-encoded Strings of octal values.
 * Can't use hex as compiler translates them before compilation.
 * These strings are split into multiple, concatenated strings.
 * Java puts them back together at compile time thankfully.
 * Java cannot handle large static arrays, so we're stuck with this
 * encode/decode approach. See analysis and runtime DFA for
 * the encoding methods.
 */

```

```

cyclicDFA(dfa) ::= <<
val DFA<dfa.decisionNumber>_eotS =
  "<dfa.javaCompressedEOT; wrap="+\n \>"
val DFA<dfa.decisionNumber>_eofS =
  "<dfa.javaCompressedEOF; wrap="+\n \>"
val DFA<dfa.decisionNumber>_minS =
  "<dfa.javaCompressedMin; wrap="+\n \>"
val DFA<dfa.decisionNumber>_maxS =
  "<dfa.javaCompressedMax; wrap="+\n \>"
val DFA<dfa.decisionNumber>_acceptS =
  "<dfa.javaCompressedAccept; wrap="+\n \>"
val DFA<dfa.decisionNumber>_specialS =

```

```

"<dfa.javaCompressedSpecial; wrap="\n \>">"}>"
val DFA<dfa.decisionNumber>_transitionS: Array[String] = Array(
  <dfa.javaCompressedTransition: {s|<s; wrap="\n \>">"; separator=",\n">
)

val DFA<dfa.decisionNumber>_eot: Array[Short] = DFA.unpackEncodedString(DFA<dfa.decisionNumber>_eotS)
val DFA<dfa.decisionNumber>_eof: Array[Short] = DFA.unpackEncodedString(DFA<dfa.decisionNumber>_eofS)
val DFA<dfa.decisionNumber>_min: Array[Char] =
DFA.unpackEncodedStringToUnsignedChars(DFA<dfa.decisionNumber>_minS)
val DFA<dfa.decisionNumber>_max: Array[Char] =
DFA.unpackEncodedStringToUnsignedChars(DFA<dfa.decisionNumber>_maxS)
val DFA<dfa.decisionNumber>_accept: Array[Short] =
DFA.unpackEncodedString(DFA<dfa.decisionNumber>_acceptS)
val DFA<dfa.decisionNumber>_special: Array[Short] =
DFA.unpackEncodedString(DFA<dfa.decisionNumber>_specialS)
val DFA<dfa.decisionNumber>_transition = new
Array[Array[Short]](DFA<dfa.decisionNumber>_transitionS.length)

for (i \<- DFA<dfa.decisionNumber>_transition.indices) {
  DFA<dfa.decisionNumber>_transition(i) =
DFA.unpackEncodedString(DFA<dfa.decisionNumber>_transitionS(i))
}

class DFA<dfa.decisionNumber> extends DFA {

  def this(recognizer: BaseRecognizer) = {
    this()
    this.recognizer = recognizer
    this.decisionNumber = <dfa.decisionNumber>
    this.eot = DFA<dfa.decisionNumber>_eot
    this.eof = DFA<dfa.decisionNumber>_eof
    this.min = DFA<dfa.decisionNumber>_min
    this.max = DFA<dfa.decisionNumber>_max
    this.accept = DFA<dfa.decisionNumber>_accept
    this.special = DFA<dfa.decisionNumber>_special
    this.transition = DFA<dfa.decisionNumber>_transition
  }
  override def getDescription = "<dfa.description>"
  <@errorMethod()>
  <if(dfa.specialStateSTs)>
  @throws(classOf[NoViableAltException])
  override def specialStateTransition(s: Int, _input: IntStream):Int = {
    <if(LEXER)>
    val input = _input
    <endif>
    <if(PARSER)>
    val input = _input.asInstanceOf[TokenStream]
    <endif>

```

```

<if(TREE_PARSER)>
val input = _input.asInstanceOf[TreeNodeStream]
<endif>
val _s = s
s match {
  <dfa.specialStateSTs:{state |
case <i0> => <! compressed special state numbers 0..n-1 !>
  <state>}; separator="\n">
case _ =>
  }
<if(backtracking)>
  if (state.backtracking>0) {state.failed=true; return -1 }<\n>
<endif>
  val nvae = new NoViableAltException(getDescription(), <dfa.decisionNumber>, _s, input)
  error(nvae)
  throw nvae
}<\n>
<endif>
}<\n>
>>

/** A state in a cyclic DFA; it's a special state and part of a big switch on
 * state.
 */
cyclicDFAState(decisionNumber,stateNumber,edges,needErrorClause,semPredState) ::= <<
val LA<decisionNumber>_<stateNumber>: Int = input.LA(1)<\n>
<if(semPredState)> <! get next lookahead symbol to test edges, then rewind !>
val index<decisionNumber>_<stateNumber>: Int = input.index()
input.rewind()<\n>
<endif>
s = -1
<edges; separator="\nelse ">
<if(semPredState)> <! return input cursor to state before we rewound !>
input.seek(index<decisionNumber>_<stateNumber>)<\n>
<endif>
if ( s>=0 ) return s
>>

/** Just like a fixed DFA edge, test the lookahead and indicate what
 * state to jump to next if successful.
 */
cyclicDFAEdge(labelExpr, targetStateNumber, edgeNumber, predicates) ::= <<
if ( (<labelExpr> <if(predicates)>&& (<predicates>)<endif>) {s = <targetStateNumber>}<\n>
>>

/** An edge pointing at end-of-token; essentially matches any char;
 * always jump to the target.
 */

```

```

eotDFAEdge(targetStateNumber,edgeNumber, predicates) ::= <<
s = <targetStateNumber><\n>
>>

// D F A E X P R E S S I O N S

andPredicates(left,right) ::= "<left>&&<right>"

orPredicates(operands) ::= "<operands; separator=\\\"\\\">"

notPredicate(pred) ::= "!(<evalPredicate(pred,\\\"\\\")>)"

evalPredicate(pred,description) ::= "<pred>"

evalSynPredicate(pred,description) ::= "<pred>()"

lookaheadTest(atom,k,atomAsInt) ::= "LA<decisionNumber>_<stateNumber>==<atom>"

/** Sometimes a lookahead test cannot assume that LA(k) is in a temp variable
 * somewhere. Must ask for the lookahead directly.
 */
isolatedLookaheadTest(atom,k,atomAsInt) ::= "input.LA(<k>)==<atom>"

lookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= <<
(LA<decisionNumber>_<stateNumber> >= <lower> && LA<decisionNumber>_<stateNumber> \<= <upper>)
>>

isolatedLookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= "(input.LA(<k>) >=<lower>
&& input.LA(<k>) \<= <upper>)"

setTest(ranges) ::= "<ranges; separator=\\\"\\\">"

// A T T R I B U T E S

globalAttributeScope(scope) ::= <<
<if(scope.attributes)>
class <scope.name>_scope {
  <scope.attributes:{it | var <it.name>: <it.type> = _}; separator="\n">
}
val <scope.name>_stack = new collection.mutable.Stack[<scope.name>_scope]<\n>
<endif>
>>

ruleAttributeScope(scope) ::= <<
<if(scope.attributes)>
class <scope.name>_scope {
  <scope.attributes:{it | var <it.name>: <it.type> = _}; separator="\n">

```

```

}
val <scope.name>_stack = new collection.mutable.Stack[<scope.name>_scope]<\n>
<endif>
>>

returnStructName(r) ::= "<r.name>_return"

returnType() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<ruleDescriptor:returnStructName()>
<else>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnType>
<else>
Unit
<endif>
<endif>
>>

/** Generate the Java type associated with a single or multiple return
 * values.
 */
ruleLabelType(referencedRule) ::= <<
<if(referencedRule.hasMultipleReturnValues)>
<referencedRule.name>_return
<else>
<if(referencedRule.hasSingleReturnValue)>
<referencedRule.singleValueReturnType>
<else>
Unit
<endif>
<endif>
>>

delegateName(d) ::= <<
<if(d.label)><d.label><else>g<d.name><endif>
>>

/** Using a type to init value map, try to init a type; if not in table
 * must be an object, default value is "null".
 */
initValue(typeName) ::= <<
<scalaTypeInitMap.(typeName)>
>>

/** Define a rule label including default value */
ruleLabelDef(label) ::= <<
var <label.label.text>: <ruleLabelType(referencedRule=label.referencedRule)> =

```

```

<initValue(typeName=ruleLabelType(referencedRule=label.referencedRule))><\n>
>>

/** Define a return struct for a rule if the code needs to access its
 * start/stop tokens, tree stuff, attributes, ... Leave a hole for
 * subgroups to stick in members.
 * TODO(matthewlloyd): make this static
 */
returnScope(scope) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
final class <ruleDescriptor:returnStructName()> extends
<if(TREE_PARSER)>Tree<else>Parser<endif>RuleReturnScope {
  <scope.attributes:{it | var <it.name>: <it.type> = _}; separator="\n">
  <@ruleReturnMembers()>
}
<endif>
>>

parameterScope(scope) ::= <<
<scope.attributes:{it | <it.name>: <it.type>}; separator=", ">
>>

parameterAttributeRef(attr) ::= "<attr.name>"
parameterSetAttributeRef(attr,expr) ::= "<attr.name> =<expr>"

scopeAttributeRef(scope,attr,index,negIndex) ::= <%
<if(negIndex)>
<scope>_stack(<scope>_stack.size-<negIndex>-1).<attr.name>
<else>
<if(index)>
<scope>_stack(<index>).<attr.name>
<else>
<scope>_stack.top.<attr.name>
<endif>
<endif>
%>

scopeSetAttributeRef(scope,attr,expr,index,negIndex) ::= <%
<if(negIndex)>
<scope>_stack(<scope>_stack.size-<negIndex>-1).<attr.name> = <expr>
<else>
<if(index)>
<scope>_stack(<index>).<attr.name> = <expr>
<else>
<scope>_stack.top.<attr.name> = <expr>
<endif>
<endif>
%>

```

```

/** $x is either global scope or x is rule with dynamic scope; refers
 * to stack itself not top of stack. This is useful for predicates
 * like {$function.size()>0 && $function::name.equals("foo")}?
 */
isolatedDynamicScopeRef(scope) ::= "<scope>_stack"

/** reference an attribute of rule; might only have single return value */
ruleLabelRef(referencedRule,scope,attr) ::= <%
<if(referencedRule.hasMultipleReturnValues)>
(if (<scope>!=null) <scope>.<attr.name> else <initValue(attr.type)>)
<else>
<scope>
<endif>
%>

returnAttributeRef(ruleDescriptor,attr) ::= <%
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name>
<else>
<attr.name>
<endif>
%>

returnSetAttributeRef(ruleDescriptor,attr,expr) ::= <%
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name> =<expr>
<else>
<attr.name> =<expr>
<endif>
%>

/** How to translate $tokenLabel */
tokenLabelRef(label) ::= "<label>"

/** ids+=ID {$ids} or e+=expr {$e} */
listLabelRef(label) ::= "list_<label>"

// not sure the next are the right approach

tokenLabelPropertyRef_text(scope,attr) ::= "(if (<scope>!=null) <scope>.getText() else null)"
tokenLabelPropertyRef_type(scope,attr) ::= "(if (<scope>!=null) <scope>.getType() else 0)"
tokenLabelPropertyRef_line(scope,attr) ::= "(if (<scope>!=null) <scope>.getLine() else 0)"
tokenLabelPropertyRef_pos(scope,attr) ::= "(if (<scope>!=null) <scope>.getCharPositionInLine() else 0)"
tokenLabelPropertyRef_channel(scope,attr) ::= "(if (<scope>!=null) <scope>.getChannel() else 0)"
tokenLabelPropertyRef_index(scope,attr) ::= "(if (<scope>!=null) <scope>.getTokenIndex() else 0)"
tokenLabelPropertyRef_tree(scope,attr) ::= "<scope>_tree"

```

```

tokenLabelPropertyRef_int(scope,attr) ::= "(if (<scope>!=null) Integer.valueOf(<scope>.getText()) else 0)"

ruleLabelPropertyRef_start(scope,attr) ::= "(if (<scope>!=null) <scope>.start.asInstanceOf[<labelType>] else null)"
ruleLabelPropertyRef_stop(scope,attr) ::= "(if (<scope>!=null) <scope>.stop.asInstanceOf[<labelType>] else null)"
ruleLabelPropertyRef_tree(scope,attr) ::= "(if (<scope>!=null) <scope>.tree.asInstanceOf[<ASTLabelType>] else
null)"
ruleLabelPropertyRef_text(scope,attr) ::= <<
<if(TREE_PARSER)>
(if (<scope>!=null) (input.getTokenStream().toString(
input.getTreeAdaptor().getTokenStartIndex(<scope>.start),
input.getTreeAdaptor().getTokenStopIndex(<scope>.start))) else null)
<else>
(if (<scope>!=null) input.toString(<scope>.start,<scope>.stop) else null)
<endif>
>>

ruleLabelPropertyRef_st(scope,attr) ::= "(if (<scope>!=null) <scope>.st else null)"

/** Isolated $RULE ref ok in lexer as it's a Token */
lexerRuleLabel(label) ::= "<label>"

lexerRuleLabelPropertyRef_type(scope,attr) ::=
"(if (<scope>!=null) <scope>.getType() else 0)"
lexerRuleLabelPropertyRef_line(scope,attr) ::=
"(if (<scope>!=null) <scope>.getLine() else 0)"
lexerRuleLabelPropertyRef_pos(scope,attr) ::=
"(if (<scope>!=null) <scope>.getCharPositionInLine() else -1)"
lexerRuleLabelPropertyRef_channel(scope,attr) ::=
"(if (<scope>!=null) <scope>.getChannel() else 0)"
lexerRuleLabelPropertyRef_index(scope,attr) ::=
"(if (<scope>!=null) <scope>.getTokenIndex() else 0)"
lexerRuleLabelPropertyRef_text(scope,attr) ::=
"(if (<scope>!=null) <scope>.getText() else null)"
lexerRuleLabelPropertyRef_int(scope,attr) ::=
"(if (<scope>!=null) Integer.valueOf(<scope>.getText()) else 0)"

// Somebody may ref $template or $tree or $stop within a rule:
rulePropertyRef_start(scope,attr) ::= "(retval.start.asInstanceOf[<labelType>])"
rulePropertyRef_stop(scope,attr) ::= "(retval.stop.asInstanceOf[<labelType>])"
rulePropertyRef_tree(scope,attr) ::= "(retval.tree.asInstanceOf[<ASTLabelType>])"
rulePropertyRef_text(scope,attr) ::= <<
<if(TREE_PARSER)>
input.getTokenStream().toString(
input.getTreeAdaptor().getTokenStartIndex(retval.start),
input.getTreeAdaptor().getTokenStopIndex(retval.start))
<else>
input.toString(retval.start,input.LT(-1))
<endif>

```

```

>>
rulePropertyRef_st(scope,attr) ::= "retval.st"

lexerRulePropertyRef_text(scope,attr) ::= "getText()"
lexerRulePropertyRef_type(scope,attr) ::= "_type"
lexerRulePropertyRef_line(scope,attr) ::= "state.tokenStartLine"
lexerRulePropertyRef_pos(scope,attr) ::= "state.tokenStartCharPositionInLine"
lexerRulePropertyRef_index(scope,attr) ::= "-1" // undefined token index in lexer
lexerRulePropertyRef_channel(scope,attr) ::= "_channel"
lexerRulePropertyRef_start(scope,attr) ::= "state.tokenStartCharIndex"
lexerRulePropertyRef_stop(scope,attr) ::= "(getCharIndex()-1)"
lexerRulePropertyRef_int(scope,attr) ::= "Integer.valueOf(<scope>.getText())"

// setting $st and $tree is allowed in local rule. everything else
// is flagged as error
ruleSetPropertyRef_tree(scope,attr,expr) ::= "retval.tree =<expr>"
ruleSetPropertyRef_st(scope,attr,expr) ::= "retval.st =<expr>"

/** How to execute an action (only when not backtracking) */
execAction(action) ::= <<
<if(backtracking)>
if ( <actions.(actionScope).synpredgate> ) {
  <action>
}
<else>
<action>
<endif>
>>

/** How to always execute an action even when backtracking */
execForcedAction(action) ::= "<action>"

// M I S C (properties, etc...)

bitset(name, words64) ::= <<
val <name> = new BitSet(Array[Long](<words64:{it | <it>L};separator=",">))<\n>
>>

codeFileExtension() ::= ".scala"

true_value() ::= "true"
false_value() ::= "false"

Found in path(s):
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-
jar/org/antlr/codegen/templates/Scala/Scala.stg
No license file was found, but licenses were detected in source scan.

```

/*

[The "BSD license"]

Copyright (c) 2005-2006 Terence Parr

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

```
@outputFile.imports() ::= <<
<@super.imports()>
<if(!TREE_PARSER)><! tree parser would already have imported !>
import org.antlr.runtime.tree.*;<\n>
<endif>
>>
```

```
@genericParser.members() ::= <<
<@super.members()>
<parserMembers()>
>>
```

```
/** Add an adaptor property that knows how to build trees */
parserMembers() ::= <<
protected var adaptor:TreeAdaptor = new CommonTreeAdaptor();<\n>
override public function set treeAdaptor(adaptor:TreeAdaptor):void {
    this.adaptor = adaptor;
    <grammar.directDelegates:{g|<g.delegateName()>.treeAdaptor = this.adaptor;}>
}
override public function get treeAdaptor():TreeAdaptor {
```

```

    return adaptor;
}
>>

@returnScope.ruleReturnMembers() ::= <<
<ASTLabelType> tree;
public function get tree():Object { return tree; }
>>

/** Add a variable to track rule's return AST */
ruleDeclarations() ::= <<
<super.ruleDeclarations()>
var root_0:<ASTLabelType> = null;<\n>
>>

ruleLabelDefs() ::= <<
<super.ruleLabelDefs()>
<[ruleDescriptor.tokenLabels,ruleDescriptor.wildcardTreeLabels,
ruleDescriptor.wildcardTreeListLabels]:{it |var <it.label.text>_tree:<ASTLabelType>=null;}; separator="\n">
<ruleDescriptor.tokenListLabels:{it |var <it.label.text>_tree:<ASTLabelType>=null;}; separator="\n">
<ruleDescriptor.allTokenRefsInAltsWithRewrites
: {it |var stream_<it>:RewriteRule<rewriteElementType>Stream=new
RewriteRule<rewriteElementType>Stream(adaptor,"token <it>");}; separator="\n">
<ruleDescriptor.allRuleRefsInAltsWithRewrites
: {it |var stream_<it>:RewriteRuleSubtreeStream=new RewriteRuleSubtreeStream(adaptor,"rule <it>");};
separator="\n">
>>

/** When doing auto AST construction, we must define some variables;
 * These should be turned off if doing rewrites. This must be a "mode"
 * as a rule could have both rewrite and AST within the same alternative
 * block.
 */
@alt.declarations() ::= <<
<if(autoAST)>
<if(outerAlt)>
<if(!rewriteMode)>
root_0 = <ASTLabelType>(adaptor.nil());<\n>
<endif>
<endif>
<endif>
>>

// Tracking Rule Elements

/** ID and track it for use in a rewrite rule */
tokenRefTrack(token,label,elementIndex,terminalOptions) ::= <<
<tokenRefBang(...)> <! Track implies no auto AST construction!>

```

```

<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>stream_<token>.add(<label>);<n>
>>

/** ids+=ID and track it for use in a rewrite rule; adds to ids *and*
 * to the tracking list stream_ID for use in the rewrite.
 */
tokenRefTrackAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<tokenRefTrack(...)>
<listLabel(elem=label,...)>
>>

/** ^(ID ...) track for rewrite */
tokenRefRuleRootTrack(token,label,elementIndex,terminalOptions) ::= <<
<tokenRefBang(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>stream_<token>.add(<label>);<n>
>>

/** Match ^(label+=TOKEN ...) track for rewrite */
tokenRefRuleRootTrackAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<tokenRefRuleRootTrack(...)>
<listLabel(elem=label,...)>
>>

/** rule when output=AST and tracking for rewrite */
ruleRefTrack(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>stream_<rule.name>.add(<label>.tree);
>>

/** x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefTrack(...)>
<listLabel(label, {<label>.tree})>
>>

/** ^(rule ...) rewrite */
ruleRefRuleRootTrack(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>stream_<rule>.add(<label>.tree);
>>

/** ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRootTrack(...)>
<listLabel(label, {<label>.tree})>
>>

// R e w r i t e

```

```

rewriteCode(
  alts, description,
  referencedElementsDeep, // ALL referenced elements to right of ->
  referencedTokenLabels,
  referencedTokenListLabels,
  referencedRuleLabels,
  referencedRuleListLabels,
  referencedWildcardLabels,
  referencedWildcardListLabels,
  rewriteBlockLevel, enclosingTreeLevel, treeLevel) ::=
<<

// AST REWRITE
// elements: <referencedElementsDeep; separator=", ">
// token labels: <referencedTokenLabels; separator=", ">
// rule labels: <referencedRuleLabels; separator=", ">
// token list labels: <referencedTokenListLabels; separator=", ">
// rule list labels: <referencedRuleListLabels; separator=", ">
// wildcard labels: <[referencedWildcardLabels, referencedWildcardListLabels]; separator=", ">
<if(backtracking)>
if ( <actions.(actionScope).synpredgate> ) {<\n>
<endif>
<prevRuleRootRef(>.tree = root_0;
<rewriteCodeLabels(>
root_0 = <ASTLabelType>(adaptor.nil());
<alts:rewriteAlt(> separator="else ">
<! if tree parser and rewrite=true !>
<if(TREE_PARSER)>
<if(rewriteMode)>
<prevRuleRootRef(>.tree = <ASTLabelType>(adaptor.rulePostProcessing(root_0));
input.replaceChildren(adaptor.getParent(retval.start),
    adaptor.getChildIndex(retval.start),
    adaptor.getChildIndex(_last),
    retval.tree);
<endif>
<endif>
<! if parser or tree-parser && rewrite!=true, we need to set result !>
<if(!TREE_PARSER)>
<prevRuleRootRef(>.tree = root_0;
<else>
<if(!rewriteMode)>
<prevRuleRootRef(>.tree = root_0;
<endif>
<endif>
<if(backtracking)>
}
<endif>

```

```

>>

rewriteCodeLabels() ::= <<
<referencedTokenLabels
  :{it |var stream_<it>:RewriteRule<rewriteElementType>Stream=new
RewriteRule<rewriteElementType>Stream(adaptor,"token <it>",<it>)};
  separator="\n"
>
<referencedTokenListLabels
  :{it |var stream_<it>:RewriteRule<rewriteElementType>Stream=new
RewriteRule<rewriteElementType>Stream(adaptor,"token <it> ", list_<it>)};
  separator="\n"
>
<referencedWildcardLabels
  :{it |var stream_<it>:RewriteRuleSubtreeStream=new RewriteRuleSubtreeStream(adaptor,"wildcard
<it>",<it>)};
  separator="\n"
>
<referencedWildcardListLabels
  :{it |var stream_<it>:RewriteRuleSubtreeStream=new RewriteRuleSubtreeStream(adaptor,"wildcard
<it> ",list_<it>)};
  separator="\n"
>
<referencedRuleLabels
  :{it |var stream_<it>:RewriteRuleSubtreeStream=new RewriteRuleSubtreeStream(adaptor,"rule
<it>",<it>!=null?<it>.tree:null)};
  separator="\n"
>
<referencedRuleListLabels
  :{it |var stream_<it>:RewriteRuleSubtreeStream=new RewriteRuleSubtreeStream(adaptor,"token
<it> ",list_<it>)};
  separator="\n"
>
>>

```

```

/** Generate code for an optional rewrite block; note it uses the deep ref'd element

```

```

 * list rather shallow like other blocks.

```

```

 */

```

```

rewriteOptionalBlock(
  alt,rewriteBlockLevel,
  referencedElementsDeep, // all nested refs
  referencedElements, // elements in immediately block; no nested blocks
  description) ::=
<<
// <fileName>:<description>
if ( <referencedElementsDeep:{el | stream_<el>.hasNext}; separator="||"> ) {
  <alt>
}

```

```

<referencedElementsDeep:{el | stream_<el>.reset();<\n>}>
>>

rewriteClosureBlock(
  alt,rewriteBlockLevel,
  referencedElementsDeep, // all nested refs
  referencedElements, // elements in immediately block; no nested blocks
  description) ::=
<<
// <fileName>:<description>
while ( <referencedElements:{el | stream_<el>.hasNext}; separator="||"> ) {
  <alt>
}
<referencedElements:{el | stream_<el>.reset();<\n>}>
>>

rewritePositiveClosureBlock(
  alt,rewriteBlockLevel,
  referencedElementsDeep, // all nested refs
  referencedElements, // elements in immediately block; no nested blocks
  description) ::=
<<
if ( !(<referencedElements:{el | stream_<el>.hasNext}; separator="||">) ) {
  throw new RewriteEarlyExitException();
}
while ( <referencedElements:{el | stream_<el>.hasNext}; separator="||"> ) {
  <alt>
}
<referencedElements:{el | stream_<el>.reset();<\n>}>
>>

rewriteAlt(a) ::= <<
// <a.description>
<if(a.pred)>
if (<a.pred>) {
  <a.alt>
}<\n>
<else>
{
  <a.alt>
}<\n>
<endif>
>>

/** For empty rewrites: "r : ... -> ;" */
rewriteEmptyAlt() ::= "root_0 = null;"

rewriteTree(root,children,description,enclosingTreeLevel,treeLevel) ::= <<

```

```

// <fileName>:<description>
{
var root_<treeLevel>:<ASTLabelType> = <ASTLabelType>(adaptor.nil());
<root:rewriteElement()>
<children:rewriteElement()>
adaptor.addChild(root_<enclosingTreeLevel>, root_<treeLevel>);
}<\n>
>>

rewriteElementList(elements) ::= "<elements:rewriteElement()>"

rewriteElement(e) ::= <<
<@pregen()>
<e.el>
>>

/** Gen ID or ID[args] */
rewriteTokenRef(token,elementIndex,terminalOptions,args) ::= <<
adaptor.addChild(root_<treeLevel>, <createRewriteNodeFromElement(...)>);<\n>
>>

/** Gen $label ... where defined via label=ID */
rewriteTokenLabelRef(label,elementIndex) ::= <<
adaptor.addChild(root_<treeLevel>, stream_<label>.nextNode());<\n>
>>

/** Gen $label ... where defined via label+=ID */
rewriteTokenListLabelRef(label,elementIndex) ::= <<
adaptor.addChild(root_<treeLevel>, stream_<label>.nextNode());<\n>
>>

/** Gen ^($label ...) */
rewriteTokenLabelRefRoot(label,elementIndex) ::= <<
root_<treeLevel> = <ASTLabelType>(adaptor.becomeRoot(stream_<label>.nextNode(), root_<treeLevel>));<\n>
>>

/** Gen ^($label ...) where label+=... */
rewriteTokenListLabelRefRoot ::= rewriteTokenLabelRefRoot

/** Gen ^(ID ...) or ^(ID[args] ...) */
rewriteTokenRefRoot(token,elementIndex,terminalOptions,args) ::= <<
root_<treeLevel> = <ASTLabelType>(adaptor.becomeRoot(<createRewriteNodeFromElement(...)>,
root_<treeLevel>));<\n>
>>

rewriteImaginaryTokenRef(args,token,terminalOptions,elementIndex) ::= <<
adaptor.addChild(root_<treeLevel>, <createImaginaryNode(token,terminalOptions,args)>);<\n>
>>

```

```

rewriteImaginaryTokenRefRoot(args,token,terminalOptions,elementIndex) ::= <<
root_<treeLevel> = <ASTLabelType>(adaptor.becomeRoot(<createImaginaryNode(token,terminalOptions,args)>,
root_<treeLevel>));<\n>
>>

/** plain -> {foo} action */
rewriteAction(action) ::= <<
root_0 = <action>;<\n>
>>

/** What is the name of the previous value of this rule's root tree? This
* let's us refer to $rule to mean previous value. I am reusing the
* variable 'tree' sitting in retval struct to hold the value of root_0 right
* before I set it during rewrites. The assign will be to retval.tree.
*/
prevRuleRootRef() ::= "retval"

rewriteRuleRef(rule) ::= <<
adaptor.addChild(root_<treeLevel>, stream_<rule>.nextTree());<\n>
>>

rewriteRuleRefRoot(rule) ::= <<
root_<treeLevel> = <ASTLabelType>(adaptor.becomeRoot(stream_<rule>.nextNode(), root_<treeLevel>));<\n>
>>

rewriteNodeAction(action) ::= <<
adaptor.addChild(root_<treeLevel>, <action>);<\n>
>>

rewriteNodeActionRoot(action) ::= <<
root_<treeLevel> = <ASTLabelType>(adaptor.becomeRoot(<action>, root_<treeLevel>));<\n>
>>

/** Gen $ruleLabel ... where defined via ruleLabel=rule */
rewriteRuleLabelRef(label) ::= <<
adaptor.addChild(root_<treeLevel>, stream_<label>.nextTree());<\n>
>>

/** Gen $ruleLabel ... where defined via ruleLabel+=rule */
rewriteRuleListLabelRef(label) ::= <<
adaptor.addChild(root_<treeLevel>, stream_<label>.nextTree());<\n>
>>

/** Gen ^($ruleLabel ...) where ruleLabel=rule */
rewriteRuleLabelRefRoot(label) ::= <<
root_<treeLevel> = <ASTLabelType>(adaptor.becomeRoot(stream_<label>.nextNode(), root_<treeLevel>));<\n>
>>

```

```

/** Gen ^($ruleLabel ...) where ruleLabel+=rule */
rewriteRuleListLabelRefRoot(label) ::= <<
root_<treeLevel> = <ASTLabelType>(adaptor.becomeRoot(stream_<label>.nextNode(), root_<treeLevel>));<\n>
>>

```

```

rewriteWildcardLabelRef(label) ::= <<
adaptor.addChild(root_<treeLevel>, stream_<label>.nextTree());<\n>
>>

```

```

createImaginaryNode(tokenType,terminalOptions,args) ::= <<
<if(terminalOptions.node)>
<! new MethodNode(IDLabel, args) !>
new <terminalOptions.node>(<tokenType><if(args)>, <args; separator=", "><endif>)
<else>
<ASTLabelType>(adaptor.create(<tokenType>, <args; separator=", "><if(!args)>"<tokenType>"<endif>))
<endif>
>>

```

```

createRewriteNodeFromElement(token,terminalOptions,args) ::= <<
<if(terminalOptions.node)>
new <terminalOptions.node>(stream_<token>.nextToken(<if(args)>, <args; separator=", "><endif>))
<else>
<if(args)> <! must create new node from old !>
adaptor.create(<token>, <args; separator=", ">)
<else>
stream_<token>.nextNode()
<endif>
<endif>
>>

```

Found in path(s):

```

* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-
jar/org/antlr/codegen/templates/ActionScript/AST.stg

```

No license file was found, but licenses were detected in source scan.

```

/*

```

[The "BSD license"]

Copyright (c) 2010 Terence Parr

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright

- notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

/** How to generate rules derived from left-recursive rules.

- * These rely on recRuleDefArg(), recRuleAltPredicate(),
- * recRuleArg(), recRuleSetResultAction(), recRuleSetReturnAction()
- * templates in main language.stg

*/

group LeftRecursiveRules;

recRuleName(ruleName) ::= "<ruleName>_"

recPrimaryName(ruleName) ::= "<ruleName>_primary"

recRuleStart(ruleName, minPrec, userRetvals, userRetvalAssignments) ::= <<

<ruleName><if(userRetvals)> returns [<userRetvals>]<endif>

: <recRuleName(...)>[<minPrec>]

<if(userRetvals)>

{

<userRetvalAssignments; separator="\n">

}

<endif>

;

>>

recRule(ruleName, precArgDef, argName, alts, setResultAction, buildAST,
userRetvals, userRetvalAssignments) ::= <<

<recRuleName(...)>[<precArgDef>]<if(userRetvals)> returns [<userRetvals>]<endif>

: <recPrimaryName(...)>

<if(buildAST)>

{

<setResultAction>

}

<endif>

<if(userRetvals)>

```

    {
      <userRetValAssignments; separator="\n">
    }
    <endif>
    ( options {backtrack=false;}
    : ( options {backtrack=false;}
      : <alts; separator="\n | ">
    )
  )*)
;
>>

```

```

recPrimaryRule(ruleName, alts, userRetvals) ::= <<
<recPrimaryName(...)><if(userRetvals)> returns [<userRetvals>]<endif>
options {backtrack=true;}
: <alts; separator="\n | ">
;
>>

```

```

recRuleAlt(alt, pred) ::= "{<pred>}?=> <alt>"

```

```

recRuleRef(ruleName, arg) ::= "<recRuleName(...)>[<arg>]"

```

Found in path(s):

```

* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-
jar/org/antlr/codegen/templates/LeftRecursiveRules.stg

```

No license file was found, but licenses were detected in source scan.

```

/*
* [The "BSD license"]
* Copyright (c) 2005-2008 Terence Parr
* All rights reserved.
*
* Conversion to C#:
* Copyright (c) 2008-2009 Sam Harwell, Pixel Mine, Inc.
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products
* derived from this software without specific prior written permission.
*

```

```

* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```

```
*/
```

```
/** Template overrides to add debugging to normal Java output;
```

```
* If ASTs are built, then you'll also get ASTDbg.stg loaded.
```

```
*/
```

```
@outputFile.imports() ::= <<
```

```
<@super.imports(>
```

```
using Antlr.Runtime.Debug;
```

```
using IOException = System.IO.IOException;
```

```
>>
```

```
@genericParser.members() ::= <<
```

```
<if(grammar.grammarIsRoot)>
```

```
public static readonly string[] ruleNames =
```

```
new string[]
```

```
{
```

```
"invalidRule", <grammar.allImportedRules:{rST | "<rST.name>"}; wrap="\n ", separator=", ">
```

```
};<\n>
```

```
<endif>
```

```
<if(grammar.grammarIsRoot)><! grammar imports other grammar(s) !>
```

```
int ruleLevel = 0;
```

```
public virtual int RuleLevel { get { return ruleLevel; } }
```

```
public virtual void IncRuleLevel() { ruleLevel++; }
```

```
public virtual void DecRuleLevel() { ruleLevel--; }
```

```
<if(profile)>
```

```
<ctorForProfilingRootGrammar(>
```

```
<else>
```

```
<ctorForRootGrammar(>
```

```
<endif>
```

```
<ctorForPredefinedListener(>
```

```
<else><! imported grammar !>
```

```
public int RuleLevel { get { return <grammar.delegators:{g| <g.delegateName(>>}.RuleLevel; } }
```

```
public void IncRuleLevel() { <grammar.delegators:{g| <g.delegateName(>>}.IncRuleLevel(); }
```

```
public void DecRuleLevel() { <grammar.delegators:{g| <g.delegateName(>>}.DecRuleLevel(); }
```

```
<ctorForDelegateGrammar(>
```

```
<endif>
```

```
<if(profile)>
```

```
public override bool AlreadyParsedRule( IInputStream input, int ruleIndex )
```

```

{
int stopIndex = GetRuleMemoization(ruleIndex, input.Index);
((Profiler)dbg).ExamineRuleMemoization(input, ruleIndex, stopIndex,
<grammar.composite.rootGrammar.recognizerName>.ruleNames[ruleIndex]);
return base.AlreadyParsedRule(input, ruleIndex);
}<\n>
public override void Memoize( IIntStream input, int ruleIndex, int ruleStartIndex )
{
((Profiler)dbg).Memoize(input, ruleIndex, ruleStartIndex,
<grammar.composite.rootGrammar.recognizerName>.ruleNames[ruleIndex]);
base.Memoize(input, ruleIndex, ruleStartIndex);
}<\n>
<endif>
protected virtual bool EvalPredicate( bool result, string predicate )
{
dbg.SemanticPredicate( result, predicate );
return result;
}<\n>
>>

ctorForRootGrammar() ::= <<
<! bug: can't use <@super.members()> cut-n-paste instead !>
<! Same except we add port number and profile stuff if root grammar !>
<actions.(actionScope).ctorModifier; null="public"> <name>( <inputStreamType> input )
: this( input, DebugEventSocketProxy.DefaultDebuggerPort, new RecognizerSharedState() )
{
}
<actions.(actionScope).ctorModifier; null="public"> <name>( <inputStreamType> input, int port,
RecognizerSharedState state )
: base( input, state )
{
<createListenerAndHandshake()>
<grammar.directDelegates:{g|<g:delegateName()> = new <g.recognizerName>( input, dbg, this.state,
this<grammar.delegates:{g|, <g:delegateName()>}> );}; separator="\n">
<parserCtorBody()>
<@finally()>
}<\n>
>>

ctorForProfilingRootGrammar() ::= <<
<! bug: can't use <@super.members()> cut-n-paste instead !>
<actions.(actionScope).ctorModifier; null="public"> <name>( <inputStreamType> input )
: this( input, new Profiler(null), new RecognizerSharedState() )
{
}
<actions.(actionScope).ctorModifier; null="public"> <name>( <inputStreamType> input, IDebugEventListener dbg,
RecognizerSharedState state )
: base( input, dbg, state )

```

```

{
  Profiler p = (Profiler)dbg;
  p.setParser(this);
  <grammar.directDelegates:
    {g|<g:delegateName()> = new <g.recognizerName>( input, dbg, this.state, this<grammar.delegators:{g|,
  <g:delegateName()>> );}; separator="\n">
  <parserCtorBody()>
  <@finally()>
}
<\n>
>>

/** Basically we don't want to set any dbg listeners are root will have it. */
ctorForDelegateGrammar() ::= <<
<actions.(actionScope).ctorModifier; null="public"> <name>( <inputStreamType> input, IDebugEventListener dbg,
RecognizerSharedState state<grammar.delegators:{g|, <g.recognizerName> <g:delegateName()>> )
: base( input, dbg, state )
{
  <grammar.directDelegates:
    {g|<g:delegateName()> = new <g.recognizerName>( input, this, this.state<grammar.delegators:{g|,
  <g:delegateName()>> );}; separator="\n">
  <parserCtorBody()>
}<\n>
}>>

ctorForPredefinedListener() ::= <<
<actions.(actionScope).ctorModifier; null="public"> <name>( <inputStreamType> input, IDebugEventListener dbg
)
<@superClassRef>: base( input, dbg, new RecognizerSharedState() )<@end>
{
<if(profile)>
  Profiler p = (Profiler)dbg;
  p.setParser(this);
<endif>
  <grammar.directDelegates:{g|<g:delegateName()> = new <g.recognizerName>(input, dbg, this.state,
this<grammar.delegators:{g|, <g:delegateName()>>);}; separator="\n">
  <parserCtorBody()>
  <@finally()>
}<\n>
}>>

createListenerAndHandshake() ::= <<
<if(TREE_PARSER)>
DebugEventSocketProxy proxy = new DebugEventSocketProxy( this, port, input.TreeAdaptor );<\n>
<else>
DebugEventSocketProxy proxy = new DebugEventSocketProxy( this, port, null );<\n>
<endif>
DebugListener = proxy;

```

```

try
{
    proxy.Handshake();
}
catch ( IOException ioe )
{
    ReportError( ioe );
}
>>

@genericParser.superClassName() ::= "Debug<@super.superClassName()>"

/*
* Much of the following rules were merged into CSharp3.stg.
*/

@rule.preamble() ::= <<
if (RuleLevel == 0)
    DebugListener.Commence();
IncRuleLevel();
>>
//@rule.preamble() ::= <<
//try
//{
// dbg.EnterRule( GrammarFileName, "<ruleName>" );
// if ( RuleLevel == 0 )
// {
//   dbg.Commence();
// }
// IncRuleLevel();
// dbg.Location( <ruleDescriptor.tree.line>, <ruleDescriptor.tree.charPositionInLine> );<\n>
//>>

@rule.postamble() ::= <<
DecRuleLevel();
if (RuleLevel == 0)
    DebugListener.Terminate();
>>
//@rule.postamble() ::= <<
//dbg.Location(<ruleDescriptor.EORNode.line>, <ruleDescriptor.EORNode.charPositionInLine>);<\n>
//}
//finally
//{
// dbg.ExitRule( GrammarFileName, "<ruleName>" );
// DecRuleLevel();
// if ( RuleLevel == 0 )
// {
//   dbg.Terminate();

```

```

// }
//}<\n>
//>>

@@insertSynpreds.start() ::= "dbg.BeginBacktrack( state.backtracking );"
@@insertSynpreds.stop() ::= "dbg.EndBacktrack( state.backtracking, success );"

// Common debug event triggers used by region overrides below

//enterSubRule() ::= <<
//try
//{
// dbg.EnterSubRule( <decisionNumber> );<\n>
//>>

//exitSubRule() ::= <<
//}
//finally
//{
// dbg.ExitSubRule( <decisionNumber> );
//}<\n>
//>>

//enterDecision() ::= <<
//try
//{
// dbg.EnterDecision( <decisionNumber> );<\n>
//>>

//exitDecision() ::= <<
//}
//finally
//{
// dbg.ExitDecision( <decisionNumber> );
//}<\n>
//>>

//enterAlt(n) ::= "dbg.EnterAlt( <n> );<\n>"

// Region overrides that tell various constructs to add debugging triggers

@@block.predecision() ::= "<enterSubRule()><enterDecision()>"

@@block.postdecision() ::= "<exitDecision()>"

@@block.postbranch() ::= "<exitSubRule()>"

@@ruleBlock.predecision() ::= "<enterDecision()>"

```

```

@@ruleBlock.postdecision() ::= "<exitDecision(>"

@@ruleBlockSingleAlt.preal() ::= "<enterAlt(n=\"1\")>"

@@blockSingleAlt.preal() ::= "<enterAlt(n=\"1\")>"

@@positiveClosureBlock.preloop() ::= "<enterSubRule(>"

@@positiveClosureBlock.postloop() ::= "<exitSubRule(>"

@@positiveClosureBlock.predecision() ::= "<enterDecision(>"

@@positiveClosureBlock.postdecision() ::= "<exitDecision(>"

@@positiveClosureBlock.earlyExitException() ::=
// "dbg.RecognitionException( eee<decisionNumber> );<n>"

@@closureBlock.preloop() ::= "<enterSubRule(>"

@@closureBlock.postloop() ::= "<exitSubRule(>"

@@closureBlock.predecision() ::= "<enterDecision(>"

@@closureBlock.postdecision() ::= "<exitDecision(>"

@@altSwitchCase.preal() ::= "<enterAlt(n=i)>"

@@element.prematch() ::=
// "dbg.Location( <it.line>, <it.pos> );"

@@matchSet.mismatchedSetException() ::=
// "dbg.RecognitionException( mse );"

@@dfaState.noViableAltException() ::= "dbg.RecognitionException( nvae );"

@@dfaStateSwitch.noViableAltException() ::= "dbg.RecognitionException( nvae );"

dfaDecision(decisionNumber,description) ::= <<
//try
//{
// isCyclicDecision = true;
// <super.dfaDecision(...)>
//}
//catch ( NoViableAltException nvae )
//{
// dbg.RecognitionException( nvae );
// throw nvae;

```

```
//}
//>>

//@cyclicDFA.errorMethod() ::= <<
//public override void Error( NoViableAltException nvae )
//{{
// ((DebugParser)recognizer).dbg.RecognitionException( nvae );
//}}
//>>
```

```
/** Force predicate validation to trigger an event */
evalPredicate(pred,description) ::= <<
EvalPredicate(<pred>, "<description>")
>>
```

Found in path(s):

```
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-
jar/org/antlr/codegen/templates/CSharp3/Dbg.stg
```

No license file was found, but licenses were detected in source scan.

```
/*
[The "BSD license"]
Copyright (c) 2005-2006 Terence Parr
All rights reserved.
```

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```
*/
/** Template subgroup to add template rewrite output
```

* If debugging, then you'll also get STDbg.stg loaded.

*/

```
@outputFile.imports() ::= <<
<@super.imports(>
import org.stringtemplate.v4.*;
import java.util.HashMap;
>>
```

```
/** Add this to each rule's return value struct */
@returnScope.ruleReturnMembers() ::= <<
/* ST returnScope.ruleReturnMembers -- empty */
>>
```

```
/** Add this to each rule's return value struct */
@returnScope.ruleReturn.memvars() ::= <<
ST *st;
>>
```

```
/** Add this to each rule's return value struct */
@returnScope.ruleReturn.properties() ::= <<
@property (retain) ST *st;
>>
```

```
/** Add this to each rule's return value struct */
@returnScope.ruleReturn.methodsDecl() ::= <<
- (ST *)getTemplate;
- (NSString *)toString;
>>
```

```
/** Add this to each rule's return value struct */
@returnScope.ruleReturn.synthesize() ::= <<
@synthesize st;
>>
```

```
/** Add this to each rule's return value struct */
@returnScope.ruleReturn.methods() ::= <<
- (ST *)getTemplate { return st; }
- (NSString *)toString { return st==nil?nil:[st render]; }
>>
```

```
@genericParser.members() ::= <<
<@super.members(>
STGroup *templateLib = [STGroup newSTGroup];

- (void)setTemplateLib:(STGroup *)aTemplateLib
{
    self.templateLib = aTemplateLib;
}
```

```

}

- (STGroup *)getTemplateLib
{
    return templateLib;
}
>>

@genericParserHeaderFile.memVars() ::= <<
<@super.memVars()>
/* ST genericParserHeaderFile.memVars -- empty now */
STGroup *templateLib; /* ST -- really a part of STAttrMap */
>>

@genericParserHeaderFile.properties() ::= <<
<@super.properties()>
/* ST genericParser.properties */
@property (retain, getter=getTemplateLib, setter=setTemplateLib:) STGroup *templateLib;
>>

@genericParserHeaderFile.methodsDecl() ::= <<
<@super.methodsDecl()>
/* ST genericParser.methodsDecl */
- init;
- (STGroup *) getTemplateLib;
- (void) setTemplateLib:(STGroup *)aTemplateLib;
@end
>>

@genericParser.synthesize() ::= <<
<@super.synthesize()>
/* ST genericParserImplementation.synthesize */
@synthesize templateLib;
>>

@genericParser.methods() ::= <<
<@super.methods()>
/* ST genericParser.methods */

- (STGroup *)getTemplateLib
{
    return templateLib;
}

- (void) setTemplateLib:(STGroup *)aTemplateLib
{
    templateLib = aTemplateLib;
}

```

```

>>

@genericParser.members() ::= <<
<@super.members()>
STGroup *templateLib = [STGroup newSTGroup];

- (STGroup *) getTemplateLib
{
    return templateLib;
}

- (void) setTemplateLib:(STGroup *) templateLib
{
    this.templateLib = templateLib;
}

/** x+=rule when output=template */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(rule,label,elementIndex,args,scope)>
<listLabel(label, { [<label> getTemplate]; })>
>>

rewriteTemplate(alts) ::= <<

// TEMPLATE REWRITE
<if(backtracking)>
if ( <actions.(actionScope).synpredgate> ) {
    <alts:rewriteTemplateAlt(); separator="else ">
    <if(rewriteMode)><replaceTextInLine()><endif>
}
<else>
<alts:rewriteTemplateAlt(); separator="else ">
<if(rewriteMode)><replaceTextInLine()><endif>
<endif>
>>

replaceTextInLine() ::= <<
<if(TREE_PARSER)>
    [((TokenRewriteStream)input) getTokenStream]
    replaceFromIndex:[input getTreeAdaptor] getTokenStartIndex:retval.start]
    ToIndex:[input getTreeAdaptor] getTokenStopIndex:retval.start]
    Text:[retval.st render]];
<else>
    [((TokenRewriteStream)input)
    replaceFromIndex:[((Token)retval.start) getTokenIndex]
    ToIndex:[input LT:-1] getTokenIndex]
    Text:[retval.st render]];

```

```

<endif>
>>

rewriteTemplateAlt(alt) ::= <<
// <alt.description>
<if(alt.pred)>
if (<alt.pred>) {
    retval.st = <alt.alt>;
}<\n>
<else>
{
    retval.st = <alt.alt>;
}<\n>
<endif>
>>

rewriteEmptyTemplate(alts) ::= <<
nil;
>>

/** Invoke a template with a set of attribute name/value pairs.
 * Set the value of the rule's template *after* having set
 * the attributes because the rule's template might be used as
 * an attribute to build a bigger template; you get a self-embedded
 * template.
 */
rewriteExternalTemplate(name,args) ::= <%
<if(args)><args:{a | []}><endif>
[templateLib getInstanceOf:@"<name>"]
<if(args)><args:{a | add:@"<a.name>" value:<a.value>]}><endif>;
%>

/** expr is a string expression that says what template to load */
rewriteIndirectTemplate(expr,args) ::= <%
<if(args)><args:{a | []}><endif>
[templateLib getInstanceOf:<expr>];
<if(args)><args:{a | add:@"<a.name>" value:<a.value>]}><endif>;
%>

/** Invoke an inline template with a set of attribute name/value pairs */
rewriteInlineTemplate(args, template) ::= <%
<if(args)><args:{a | []}><endif>
[ST newST:templateLib template:"<template>"]
<if(args)><args:{a | add:@"<a.name>" <a.value>]}><endif>;
%>

/** plain -> {foo} action */
rewriteAction(action) ::= <<

```

<action>

>>

```
/** An action has %st.attrName=expr; or % {st}.attrName=expr; */
```

```
actionSetAttribute(st,attrName,expr) ::= <<
```

```
[(<st>) setAttribute:@"<attrName>" value:<expr>];
```

>>

```
/** Translate %{stringExpr} */
```

```
actionStringConstructor(stringExpr) ::= <<
```

```
[ST newST:templateLib template:<stringExpr>];
```

>>

Found in path(s):

* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-

jar/org/antlr/codegen/templates/ObjC/ST4ObjC.stg

No license file was found, but licenses were detected in source scan.

/*

* [The "BSD license"]

* Copyright (c) 2011 Terence Parr

* All rights reserved.

*

* Conversion to C#:

* Copyright (c) 2011 Sam Harwell, Tunnel Vision Laboratories, LLC.

* All rights reserved.

*

* Redistribution and use in source and binary forms, with or without

* modification, are permitted provided that the following conditions

* are met:

* 1. Redistributions of source code must retain the above copyright

* notice, this list of conditions and the following disclaimer.

* 2. Redistributions in binary form must reproduce the above copyright

* notice, this list of conditions and the following disclaimer in the

* documentation and/or other materials provided with the distribution.

* 3. The name of the author may not be used to endorse or promote products

* derived from this software without specific prior written permission.

*

* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR

* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES

* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.

* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,

* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT

* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,

* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY

* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT

* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF

* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

```
dfa(decisionRanks,states,edges,rankdir,startState,useBox) ::= <<
digraph NFA {
<if(rankdir)>rankdir=<rankdir>;<endif>
<decisionRanks; separator="\n">
<states; separator="\n">
<edges; separator="\n">
}
>>
```

```
nfa(decisionRanks,states,edges,rankdir,startState) ::= <<
digraph NFA {
rankdir=LR;
<decisionRanks; separator="\n">
<states; separator="\n">
<edges; separator="\n">
}
>>
```

```
decision-rank(states) ::= <<
{rank=same; rankdir=TB; <states; separator=" "; ">}
>>
```

```
edge(src,target,label,arrowhead) ::= <<
<src> -> <target> [fontsize=11, fontname="Courier", arrowsize=.7, label = "<label>"<if(arrowhead)>, arrowhead =
<arrowhead><endif>];
>>
```

```
action-edge(src,target,label,arrowhead) ::= <<
<src> -> <target> [fontsize=11, fontname="Courier", arrowsize=.7, label = "<label>"<if(arrowhead)>, arrowhead =
<arrowhead><endif>];
>>
```

```
epsilon-edge(src,target,label,arrowhead) ::= <<
<src> -> <target> [fontname="Times-Italic", label = "e"];
>>
```

```
state(name,useBox) ::= <<
node [fontsize=11, shape = <if(useBox)>box<else>circle, fixedsize=true, width=.4<endif>]; <name>
>>
```

```
stopstate(name,useBox) ::= <<
node [fontsize=11, shape = <if(useBox)>polygon,sides=4,peripheries=2<else>doublecircle, fixedsize=true,
width=.6<endif>]; <name>
>>
```

Found in path(s):

* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/tool/templates/dot/dot.stg

No license file was found, but licenses were detected in source scan.

/*

* [The "BSD license"]

* Copyright (c) 2012 Terence Parr

* Copyright (c) 2012 Sam Harwell

* All rights reserved.

*

* Redistribution and use in source and binary forms, with or without

* modification, are permitted provided that the following conditions

* are met:

* 1. Redistributions of source code must retain the above copyright

* notice, this list of conditions and the following disclaimer.

* 2. Redistributions in binary form must reproduce the above copyright

* notice, this list of conditions and the following disclaimer in the

* documentation and/or other materials provided with the distribution.

* 3. The name of the author may not be used to endorse or promote products

* derived from this software without specific prior written permission.

*

* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR

* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES

* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.

* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,

* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT

* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,

* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY

* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT

* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF

* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

Found in path(s):

* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/tool/ToolSTGroupFile.java

No license file was found, but licenses were detected in source scan.

/*

[The "BSD license"]

Copyright (c) 2005-2009 Jim Idle, Temporal Wave LLC

<http://www.temporal-wave.com>

<http://www.linkedin.com/in/jimidle>

All rights reserved.

Redistribution and use in source and binary forms, with or without

modification, are permitted provided that the following conditions

are met:

1. Redistributions of source code must retain the above copyright

- notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
 3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

/** Templates for building ASTs during tree parsing.

*

- * Deal with many combinations. Dimensions are:
- * Auto build or rewrite
- * no label, label, list label (label/no-label handled together)
- * child, root
- * token, set, rule, wildcard

*

- * Each combination has its own template except that label/no label
- * is combined into tokenRef, ruleRef, ...

*/

/** Add a variable to track last element matched */

```
ruleDeclarations() ::= <<
<super.ruleDeclarations()>
<ASTLabelType> _last;<\n>
<ASTLabelType> _first_0;<\n>
>>
```

/** Add a variable to track last element matched */

```
ruleInitializations() ::= <<
<super.ruleInitializations()>
_last = NULL;<\n>
_first_0 = NULL;<\n>
>>
```

/** What to emit when there is no rewrite rule. For auto build

- * mode, does nothing.

*/

```
noRewrite(rewriteBlockLevel, treeLevel) ::= <<
```

```

<if(backtracking)>if ( BACKTRACKING ==0 ) {<endif>
<if(rewriteMode)>
retval.tree = (<ASTLabelType>)_first_0;
if ( ADAPTOR->getParent(ADAPTOR, retval.tree) != NULL && ADAPTOR->isNilNode(ADAPTOR,
ADAPTOR->getParent(ADAPTOR, retval.tree) ) )
{
    retval.tree = (<ASTLabelType>)ADAPTOR->getParent(ADAPTOR, retval.tree);
}
<endif>
<if(backtracking)>}<endif>
>>

```

```

/** match ^(root children) in tree parser; override here to
 * add tree construction actions.
 */
tree(root, actionsAfterRoot, children, nullableChildList,
    enclosingTreeLevel, treeLevel) ::= <<
_last = (<ASTLabelType>)LT(1);
{
<ASTLabelType> _save_last_<treeLevel>;
<ASTLabelType> _first_<treeLevel>;
<if(!rewriteMode)>
<ASTLabelType> root_<treeLevel>;
<endif>
_save_last_<treeLevel> = _last;
_first_<treeLevel> = NULL;
<if(!rewriteMode)>
root_<treeLevel> = (<ASTLabelType>)(ADAPTOR->nilNode(ADAPTOR));
<endif>
<root:element()>
<if(rewriteMode)>
<if(backtracking)>if ( BACKTRACKING ==0 ) {<endif>
<if(root.el.rule)>
if ( _first_<enclosingTreeLevel> == NULL ) _first_<enclosingTreeLevel> = <root.el.label>.tree;
<else>
if ( _first_<enclosingTreeLevel> == NULL ) _first_<enclosingTreeLevel> = <root.el.label>;
<endif>
<if(backtracking)>}<endif>
<endif>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if ( LA(1)==ANTLR3_TOKEN_DOWN ) {
    MATCHT(ANTLR3_TOKEN_DOWN, NULL);
    <children:element()>
    MATCHT(ANTLR3_TOKEN_UP, NULL);
}
<else>
MATCHT(ANTLR3_TOKEN_DOWN, NULL);

```

```

<children:element()>
MATCHT(ANTLR3_TOKEN_UP, NULL);
<endif>
<if(!rewriteMode)>
ADAPTOR->addChild(ADAPTOR, root_<enclosingTreeLevel>, root_<treeLevel>);
<endif>
_last = _save_last_<treeLevel>;
}<\n>
>>

// TOKEN AST STUFF

/** ID! and output=AST (same as plain tokenRef) 'cept add
 * setting of _last
 */
tokenRefBang(token,label,elementIndex,terminalOptions) ::= <<
_last = (<ASTLabelType>)LT(1);
<super.tokenRef(...)>
>>

/** ID auto construct */
tokenRef(token,label,elementIndex,terminalOptions) ::= <<
_last = (<ASTLabelType>)LT(1);
<super.tokenRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( BACKTRACKING ==0 ) {<endif>
<if(terminalOptions.node)>
<label>_tree = <terminalOptions.node>New(<label>);
<else>
<label>_tree = (<ASTLabelType>)ADAPTOR->dupNode(ADAPTOR, <label>);
<endif>
ADAPTOR->addChild(ADAPTOR, root_<treeLevel>, <label>_tree);
<if(backtracking)>}<endif>
<else>
<if(backtracking)>if ( BACKTRACKING ==0 ) {<endif>
if ( _first_<treeLevel> == NULL ) _first_<treeLevel> = <label>;
<if(backtracking)>}<endif>
<endif>
>>

/** label+=TOKEN auto construct */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

/** ^(ID ...) auto construct */
tokenRefRuleRoot(token,label,elementIndex,terminalOptions) ::= <<

```

```

_last = (<ASTLabelType>)LT(1);
<super.tokenRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( BACKTRACKING == 0 ) {<endif>
<if(terminalOptions.node)>
<label>_tree = <terminalOptions.node>New(<label>);
<else>
<label>_tree = (<ASTLabelType>)ADAPTOR->dupNode(ADAPTOR, <label>);
<endif><\n>
root_<treeLevel> = (<ASTLabelType>)ADAPTOR->becomeRoot(ADAPTOR, <label>_tree, root_<treeLevel>);
<if(backtracking)>}<endif>
<endif>
>>

```

```

/** Match ^(label+=TOKEN ...) auto construct */
tokenRefRuleRootAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<tokenRefRuleRoot(...)>
<listLabel(elem=label,...)>
>>

```

```

/** Match . wildcard and auto dup the node/subtree */
wildcard(token,label,elementIndex,terminalOptions) ::= <<
_last = (<ASTLabelType>)LT(1);
<super.wildcard(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<label>_tree = (<ASTLabelType>)ADAPTOR->dupTree(ADAPTOR, <label>);
ADAPTOR->addChild(ADAPTOR, root_<treeLevel>, <label>_tree);
<if(backtracking)>}<endif>
<else> <! rewrite mode !>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> )<endif>
if ( _first_<treeLevel>==null ) _first_<treeLevel> = <label>;
<endif>
>>

```

```

// SET AST

```

```

matchSet(s,label,terminalOptions,elementIndex,postmatchCode) ::= <<
_last = (<ASTLabelType>)LT(1);
<super.matchSet(postmatchCode={
<if(!rewriteMode)>
<if(backtracking)>if ( BACKTRACKING == 0 ) {<endif>
<if(terminalOptions.node)>
<label>_tree = <terminalOptions.node>New(<label>);
<else>
<label>_tree = (<ASTLabelType>)ADAPTOR->dupNode(ADAPTOR, <label>);
<endif><\n>
ADAPTOR->addChild(ADAPTOR, root_<treeLevel>, <label>_tree);

```

```

<if(backtracking)>\}<endif>
<endif>
}
,...>
>>

matchRuleBlockSet(s,label,terminalOptions,elementIndex,postmatchCode,treeLevel="0") ::= <<
<matchSet(...)>
<noRewrite()> <! set return tree !>
>>

matchSetBang(s,label,terminalOptions,elementIndex,postmatchCode) ::= <<
_last = (<ASTLabelType>)LT(1);
<super.matchSet(...)>
>>

matchSetRuleRoot(s,label,terminalOptions,elementIndex,debug) ::= <<
<super.matchSet(postmatchCode={
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> )<endif>
<if(terminalOptions.node)>
<label>_tree = <terminalOptions.node>New(<label>);
<else>
<label>_tree = (<ASTLabelType>)ADAPTOR->dupNode(ADAPTOR, <label>);
<endif>
root_<treeLevel> = (<ASTLabelType>)ADAPTOR->becomeRoot(ADAPTOR, <label>_tree, root_<treeLevel>);
<if(backtracking)>\}<endif>
<endif>
}, ...
)>
>>

// RULE REF AST

/** rule auto construct */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
_last = (<ASTLabelType>)LT(1);
<super.ruleRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> )
{
<endif>
<if(!rewriteMode)>
ADAPTOR->addChild(ADAPTOR, root_<treeLevel>, <label>.tree);
<else>
if ( _first_<treeLevel> == NULL ) _first_<treeLevel> = <label>.tree;
<endif>
<if(backtracking)>}<endif>
>>

```

```

/** x+=rule auto construct */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<super.listLabelAST(elem=label,...)>
>>

/** ^(rule ...) auto construct */
ruleRefRuleRoot(rule,label,elementIndex,args,scope) ::= <<
_last = (<ASTLabelType>)LT(1);
<super.ruleRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( ( <actions.(actionScope).synpredgate> ) ) <endif>root_<treeLevel> =
(<ASTLabelType>)(ADAPTOR->becomeRoot(ADAPTOR, <label>.tree, root_<treeLevel>));
<endif>
>>

/** ^(x+=rule ...) auto construct */
ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<super.listLabelAST(elem=label,...)>
>>

/** rule when output=AST and tracking for rewrite */
ruleRefTrack(rule,label,elementIndex,args,scope) ::= <<
_last = (<ASTLabelType>)LT(1);
<super.ruleRefTrack(...)>
>>

/** x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
_last = (<ASTLabelType>)LT(1);
<super.ruleRefTrackAndListLabel(...)>
>>

/** ^(rule ...) rewrite */
ruleRefRuleRootTrack(rule,label,elementIndex,args,scope) ::= <<
_last = (<ASTLabelType>)LT(1);
<super.ruleRefRootTrack(...)>
>>

/** ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
_last = (<ASTLabelType>)LT(1);
<super.ruleRefRuleRootTrackAndListLabel(...)>
>>

/** Streams for token refs are tree nodes now; override to

```

```

* change nextToken to nextNode.
*/
createRewriteNodeFromElement(token,terminalOptions,scope) ::= <<
<if(terminalOptions.node)>
<terminalOptions.node>New(stream_<token>->nextNode(stream_<token>))
<else>
stream_<token>->nextNode(stream_<token>)
<endif>
>>

ruleCleanUp() ::= <<
<super.ruleCleanUp(...)>
<if(backtracking)>
if ( <actions.(actionScope).synpredgate> ) {<\n>
<endif>
<if(!ruleDescriptor.isSynPred)>
retval.stop = LT(-1);<\n>
<endif>
retval.tree = (<ASTLabelType>)ADAPTOR->rulePostProcessing(ADAPTOR, root_0);
<if(backtracking)>
}
<endif>
<ruleDescriptor.allTokenRefsInAltsWithRewrites
: {if (stream_<it> != NULL) stream_<it>->free(stream_<it>);}; separator="\n">
<ruleDescriptor.allRuleRefsInAltsWithRewrites
: {if (stream_<it> != NULL) stream_<it>->free(stream_<it>);}; separator="\n">
>>

```

Found in path(s):

```

* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-
jar/org/antlr/codegen/templates/C/ASTTreeParser.stg
No license file was found, but licenses were detected in source scan.

```

```

/*
[The "BSD license"]
Copyright (c) 2005-2009 Jim Idle, Temporal Wave LLC
http://www.temporal-wave.com
http://www.linkedin.com/in/jimidle

```

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the

- documentation and/or other materials provided with the distribution.
- The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

/** Templates for building ASTs during normal parsing.

*

* Deal with many combinations. Dimensions are:

* Auto build or rewrite

* no label, label, list label (label/no-label handled together)

* child, root

* token, set, rule, wildcard

*

* The situation is not too bad as rewrite (->) usage makes ^ and !

* invalid. There is no huge explosion of combinations.

*/

@rule.setErrorReturnValue() ::= <<

retval.tree = (<ASTLabelType>)(ADAPTOR->errorNode(ADAPTOR, INPUT, retval.start, LT(-1), EXCEPTION));

>>

// TOKEN AST STUFF

/** ID and output=AST */

tokenRef(token,label,elementIndex,terminalOptions) ::= <<

<super.tokenRef(...)>

<if(backtracking)>if (<actions.(actionScope).synpredgate>) {<endif>

<label>_tree = (<ASTLabelType>)(ADAPTOR->create(ADAPTOR, <label>));

ADAPTOR->addChild(ADAPTOR, root_0, <label>_tree);

<if(backtracking)>}<endif>

>>

/** ID! and output=AST (same as plain tokenRef) */

tokenRefBang(token,label,elementIndex,terminalOptions) ::= "<super.tokenRef(...)>"

/** ID^ and output=AST */

tokenRefRuleRoot(token,label,elementIndex,terminalOptions) ::= <<

```

<super.tokenRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<label>_tree = <createNodeFromToken(...)>;
root_0 = (<ASTLabelType>)(ADAPTOR->becomeRoot(ADAPTOR, <label>_tree, root_0));
<if(backtracking)>}<endif>
>>

/** ids+=ID! and output=AST */
tokenRefBangAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<tokenRefBang(...)>
<listLabel(elem=label,...)>
>>

/** label+=TOKEN when output=AST but not rewrite alt */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

/** Match label+=TOKEN^ when output=AST but not rewrite alt */
tokenRefRuleRootAndListLabel(token,label,terminalOptions,elementIndex) ::= <<
<tokenRefRuleRoot(...)>
<listLabel(elem=label,...)>
>>

// SET AST

// the match set stuff is interesting in that it uses an argument list
// to pass code to the default matchSet; another possible way to alter
// inherited code. I don't use the region stuff because I need to pass
// different chunks depending on the operator. I don't like making
// the template name have the operator as the number of templates gets
// large but this is the most flexible--this is as opposed to having
// the code generator call matchSet then add root code or ruleroot code
// plus list label plus ... The combinations might require complicated
// rather than just added on code. Investigate that refactoring when
// I have more time.

matchSet(s,label,terminalOptions,elementIndex,postmatchCode) ::= <<
<super.matchSet(postmatchCode={<if(backtracking)>if ( <actions.(actionScope).synpredgate> )
<endif>ADAPTOR->addChild(ADAPTOR, root_0, <createNodeFromToken(...)>);},...)>
>>

matchRuleBlockSet(s,label,terminalOptions,elementIndex,postmatchCode,treeLevel="0") ::= <<
<matchSet(...)>
>>

```

```

matchSetBang(s,label,elementIndex,terminalOptions,postmatchCode) ::= "<super.matchSet(...)>"

// note there is no matchSetTrack because -> rewrites force sets to be
// plain old blocks of alts: (A|B|...|C)

matchSetRuleRoot(s,label,terminalOptions,elementIndex,debug) ::= <<
<if(label)>
<label>=(<labelType>)LT(1);<\n>
<endif>
<super.matchSet(postmatchCode={<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>root_0 =
(<ASTLabelType>)(ADAPTOR->becomeRoot(ADAPTOR, <createNodeFromToken(...)>, root_0));},...)>
>>

// RULE REF AST

/** rule when output=AST */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>ADAPTOR->addChild(ADAPTOR, root_0,
<label>.tree);
>>

/** rule! is same as normal rule ref */
ruleRefBang(rule,label,elementIndex,args,scope) ::= "<super.ruleRef(...)>"

/** rule^ */
ruleRefRuleRoot(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>root_0 = (<ASTLabelType>)(ADAPTOR-
>becomeRoot(ADAPTOR, <label>.tree, root_0));
>>

/** x+=rule when output=AST */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabelAST(...)>
>>

/** x+=rule! when output=AST is a rule ref with list addition */
ruleRefBangAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefBang(...)>
<listLabelAST(...)>
>>

/** x+=rule^ */
ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>

```

```

<listLabelAST(...)>
>>

// WILDCARD AST

wildcard(token,label,elementIndex,terminalOptions) ::= <<
<super.wildcard(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<label>_tree = (<ASTLabelType>)(ADAPTOR->create(ADAPTOR, <label>));
ADAPTOR->addChild(ADAPTOR, root_0, <label>_tree);
<if(backtracking)>}<endif>
>>

wildcardBang(token,label,elementIndex,terminalOptions) ::= "<super.wildcard(...)>"

wildcardRuleRoot(label,elementIndex) ::= <<
<super.wildcard(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<label>_tree = (<ASTLabelType>)(ADAPTOR->create(ADAPTOR, <label>));
root_0 = (<ASTLabelType>)(ADAPTOR->becomeRoot(ADAPTOR, <label>_tree, root_0));
<if(backtracking)>}<endif>
>>

createNodeFromToken(label,terminalOptions) ::= <<
<if(terminalOptions.node)>
<terminalOptions.node>New(<label>) <! new MethodNode(IDLabel) !>
<else>
(<ASTLabelType>)(ADAPTOR->create(ADAPTOR, <label>))
<endif>
>>

ruleCleanUp() ::= <<
<super.ruleCleanUp()>
<if(backtracking)>
if ( <actions.(actionScope).synpredgate> )
{<\n>
<endif>
<if(!ruleDescriptor.isSynPred)>
retval.stop = LT(-1);<\n>
<endif>
retval.tree = (<ASTLabelType>)(ADAPTOR->rulePostProcessing(ADAPTOR, root_0));
ADAPTOR->setTokenBoundaries(ADAPTOR, retval.tree, retval.start, retval.stop);
<ruleDescriptor.allTokenRefsInAltsWithRewrites
: {it | if (stream_<it> != NULL) stream_<it>->free(stream_<it>);}; separator="\n">
<ruleDescriptor.allRuleRefsInAltsWithRewrites
: {it | if (stream_<it> != NULL) stream_<it>->free(stream_<it>);}; separator="\n">
<if(backtracking)>
}<\n>

```

<endif>

>>

Found in path(s):

* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/C/ASTParser.stg

No license file was found, but licenses were detected in source scan.

/*

[The "BSD license"]

Copyright (c) 2006 Kay Roepke

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

/*

New style messages. This file contains the actual layout of the messages emitted by ANTLR.

The text itself is coming out of the languages/*stg files, according to the chosen locale.

This file contains the default format ANTLR uses.

*/

group antlr;

location(file, line, column) ::= "<file>:<line>:<column>:"

message(id, text) ::= "<id> <text>"

report(location, message, type) ::= "<type><message.id>: <location> <message.text>"

```
wantsSingleLineMessage() ::= "false"
```

Found in path(s):

```
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-  
jar/org/antlr/tool/templates/messages/formats/antlr.stg
```

No license file was found, but licenses were detected in source scan.

```
/*
```

```
[The "BSD license"]
```

```
Copyright (c) 2006 Kay Roepke 2010 Alan Condit
```

```
All rights reserved.
```

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

```
THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR  
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES  
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.  
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,  
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT  
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,  
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY  
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT  
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF  
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

```
*/
```

```
/** Template overrides to add debugging to AST stuff. Dynamic inheritance
```

```
* hierarchy is set up as ASTDbg : AST : Dbg : Java by code generator.
```

```
*/
```

```
parserMembers() ::= <<
```

```
DebugTreeAdaptor *adaptor = [DebugTreeAdaptor newTreeAdaptor:(id)dbg Adaptor:[CommonTreeAdaptor  
newTreeAdaptor]];
```

```
// fix this
```

```
- (void) setTreeAdaptor:(id<TreeAdaptor>)anAdaptor
```

```
{
```

```
    adaptor = [DebugTreeAdaptor newTreeAdaptor:dbg Adaptor:anAdaptor];
```

```
<if(grammar.grammarIsRoot)>
```

```

    adaptor = [DebugTreeAdaptor newTreeAdaptor:adaptor withDBG:dbg];
<else>
    adaptor = (DebugTreeAdaptor *)adaptor; // delegator sends dbg adaptor
<endif><\n>
    <grammar.directDelegates:{g[<g.delegateName()> setTreeAdaptor:adaptor];}>
}

- (id<TreeAdaptor>)getTreeAdaptor
{
    return adaptor;
}<\n>
>>

parserCtorBody() ::= <<
<super.parserCtorBody()>
>>

createListenerAndHandshake() ::= <<
DebugEventSocketProxy proxy =
    [DebugEventSocketProxy newDebugEventSocketProxy:self, port, <if(TREE_PARSER)>[input
getTreeAdaptor]<else>adaptor<endif>];
[self setDebugListener:proxy];
[self set<inputStreamType>:[Debug<inputStreamType> newDebug<inputStreamType>:input with:proxy]];
try {
    [proxy handshake];
}
@catch (IOException *ioe) {
    [self reportError:ioe];
}
>>

@ctorForRootGrammar.finally() ::= <<
CommonTreeAdaptor *adap = [CommonTreeAdaptor newTreeAdaptor];
[self setTreeAdaptor:adap];
[proxy setTreeAdaptor:adap];
>>

@ctorForProfilingRootGrammar.finally() ::= <<
CommonTreeAdaptor *adap = [CommonTreeAdaptor newTreeAdaptor];
[self setTreeAdaptor:adap];
[proxy setTreeAdaptor:adap];
>>

@ctorForPredefinedListener.superClassRef() ::= @"super(input, dbg);"

@ctorForPredefinedListener.finally() ::= <<
<if(grammar.grammarIsRoot)> <! don't create new adaptor for delegates !>
CommonTreeAdaptor *adap = [CommonTreeAdaptor newTreeAdaptor];

```

```
[self setTreeAdaptor:adap];<\n>
```

```
<endif>
```

```
>>
```

```
@treeParserHeaderFile.superClassName ::= "DebugTreeParser"
```

```
@rewriteElement.pregen() ::= "[debugListener locationLine:<e.line> column:<e.pos>];"
```

Found in path(s):

```
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-  
jar/org/antlr/codegen/templates/ObjC/ASTDbg.stg
```

No license file was found, but licenses were detected in source scan.

```
/*
```

```
[The "BSD license"]
```

```
Copyright (c) 2006, 2007 Kay Roepke 2010 Alan Condit
```

```
All rights reserved.
```

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```
*/
```

```
@genericParserHeaderFile.memVars() ::= <<
```

```
/* AST parserHeaderFile.memVars */
```

```
NSInteger ruleLevel;
```

```
NSArray *ruleNames;
```

```
<@super.memVars()> /* AST super.memVars */
```

```
<parserMemVars()> /* AST parserMemVars */
```

```
>>
```

```

@genericParserHeaderFile.properties() ::= <<
/* AST parserHeaderFile.properties */
<@super.properties()> /* AST super.properties */
<parserProperties()> /* AST parserproperties */
>>

@genericParserHeaderFile.methodsDecl() ::= <<
/* AST parserHeaderFile.methodsDecl */
<@super.methodsDecl()> /* AST super.methodsDecl */
<parserMethodsDecl()> /* AST parsermethodsDecl */
>>

@genericParser.synthesize() ::= <<
/* AST genericParser.synthesize */
<@super.synthesize()>
<parserSynthesize()>
>>

@genericParser.methods() ::= <<
/* AST genericParser.methods */
<@super.methods()>
<parserMethods()>
>>

/* additional init code for tree support */
@genericParser.init() ::= <<
/* AST genericParser.init */
<@super.init()>
[self setTreeAdaptor:[[CommonTreeAdaptor newTreeAdaptor] retain]];
>>

@genericParser.dealloc() ::= <<
/* AST genericParser.dealloc */
[self setTreeAdaptor:nil];
<@super.dealloc()>
>>

/* Add an adaptor property that knows how to build trees */
parserMemVars() ::= <<
/* AST parserMemVars */
id<TreeAdaptor> treeAdaptor;
>>

/* Add an adaptor property that knows how to build trees */
parserProperties() ::= <<
/* AST parserProperties */
@property (retain, getter=getTreeAdaptor, setter=setTreeAdaptor:) id<TreeAdaptor> treeAdaptor;
>>

```

```

/** Declaration of additional tree support methods - go in interface of parserHeaderFile() */
parserMethodsDecl() ::= <<
/* AST parserMethodsDecl */
- (id\<TreeAdaptor>) getTreeAdaptor;
- (void) setTreeAdaptor:(id\<TreeAdaptor>)theTreeAdaptor;
>>

/* Add an adaptor property that knows how to build trees */
parserSynthesize() ::= <<
/* AST parserProperties */
@synthesize treeAdaptor;
>>

/** Definition of addition tree support methods - go in implementation of genericParser() */
parserMethods() ::= <<
/* AST parserMethods */
- (id\<TreeAdaptor>) getTreeAdaptor
{
return treeAdaptor;
}

- (void) setTreeAdaptor:(id\<TreeAdaptor>)aTreeAdaptor
{
if (aTreeAdaptor != treeAdaptor) {
treeAdaptor = aTreeAdaptor;
}
}
>>

/** addition memVars for returnscopes */
@returnScopeInterface.memVars() ::= <<
/* AST returnScopeInterface.memVars */
<recognizer.ASTLabelType; null="CommonTree"> *tree;
>>

/** the interface of returnScope properties */
@returnScopeInterface.properties() ::= <<
/* AST returnScopeInterface.properties */
@property (retain, getter=getTree, setter=setTree:) <recognizer.ASTLabelType; null="CommonTree"> *tree;
>>

/** the interface of returnScope methodsDecl */
@returnScopeInterface.methodsDecl() ::= <<
/* AST returnScopeInterface.methodsDecl */
- (<recognizer.ASTLabelType; null="CommonTree"> *)getTree;<\n>
- (void) setTree:(<recognizer.ASTLabelType; null="CommonTree"> *)aTree;<\n>
>>

```

```

/** the implementation of returnScope synthesize */
@returnScopeImplementation.synthesize() ::= <<
/* AST returnScope.synthesize */
@synthesize tree;
>>

```

```

/** the implementation of returnScope methods */
@returnScopeImplementation.methods() ::= <<
/* AST returnScope.methods */
- (<ASTLabelType> *)getTree
{
    return tree;
}

```

```

- (void) setTree:<ASTLabelType> *aTree
{
    if (tree != aTree) {
        if (tree != nil) [tree release];
        if (aTree != nil) [aTree retain];
        tree = aTree;
    }
}

```

```

- (void) dealloc
{
    self.tree = nil;
    [super dealloc];
}

```

```

>>

```

```

/** Add a variable to track rule's return AST */
ruleDeclarations() ::= <<
/* AST ruleDeclarations */
<super.ruleDeclarations()>
<ASTLabelType> *root_0 = nil;<\n>
>>

```

```

ruleLabelDefs() ::= <<
/* AST ruleLabelDefs */
<super.ruleLabelDefs()>
<[ruleDescriptor.tokenLabels,ruleDescriptor.wildcardTreeLabels,
    ruleDescriptor.wildcardTreeListLabels]:{it | <ASTLabelType> *<it.label.text>_tree=nil;}; separator="\n">
<ruleDescriptor.tokenListLabels:{it | <ASTLabelType> *<it.label.text>_tree = nil;}; separator="\n">
<ruleDescriptor.allTokenRefsInAltsWithRewrites:{it | RewriteRuleTokenStream *stream_<it> =
    [[RewriteRule<rewriteElementType>Stream newRewriteRule<rewriteElementType>Stream:treeAdaptor
        description:@"token <it>"] retain];}; separator="\n">

```

```

<ruleDescriptor.allRuleRefsInAltsWithRewrites: {it | RewriteRuleSubtreeStream *stream_<it> =
  [[RewriteRuleSubtreeStream newRewriteRuleSubtreeStream:treeAdaptor
    description:@"rule <it>"] retain];}; separator="\n">
>>

ruleCleanUp() ::= <<
/* AST ruleCleanUp */
<super.ruleCleanUp()>
<[ruleDescriptor.allTokenRefsInAltsWithRewrites,ruleDescriptor.allRuleRefsInAltsWithRewrites]: {it |
  [stream_<it> release];}; separator="\n">
<!
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(backtracking)>if ( state.backtracking == 0 ) {<\n>
<endif>
  [<prevRuleRootRef()> setTree:(<ASTLabelType> *)[treeAdaptor rulePostProcessing:root_0]];<\n>
  [treeAdaptor setTokenBoundaries:<prevRuleRootRef()> getTree]
    From:<prevRuleRootRef()> getStart]
    To:<prevRuleRootRef()> getStop]];<\n>
<if(backtracking)>}<\n>
<endif>
<endif>
[root_0 release];
!>
>>

rewriteCodeLabelsCleanUp() ::= <<
/* AST rewriteCodeLabelsCleanUp */
<referencedTokenLabels: {it | [stream_<it> release];}; separator="\n">
<referencedTokenListLabels: {it | [stream_<it> release];}; separator="\n">
<referencedRuleLabels: {it | [stream_<it> release];}; separator="\n">
<referencedRuleListLabels: {it | [stream_<it> release];}; separator="\n">
>>

/** When doing auto AST construction, we must define some variables;
 * These should be turned off if doing rewrites. This must be a "mode"
 * as a rule could have both rewrite and AST within the same alternative
 * block.
 */
@alt.declarations() ::= <<
<if(autoAST)>
<if(outerAlt)>
<if(!rewriteMode)>
root_0 = (<ASTLabelType> *)[[treeAdaptor class] newEmptyTree] retain];<\n>
<endif>
<endif>
<endif>
>>

```

```

// Tracking Rule Elements

/** ID and track it for use in a rewrite rule */
tokenRefTrack(token,label,elementIndex) ::= <<
<! <super.tokenRef(...)> !>
<tokenRefBang(...)> <! Track implies no auto AST construction!>
<if(backtracking)>
if ( <actions.(actionScope).synpredgate> ) <endif>
  [stream_<token> addElement:<label>];<\n>
>>

/** ids+=ID and track it for use in a rewrite rule; adds to ids *and*
 * to the tracking list stream_ID for use in the rewrite.
 */
tokenRefTrackAndListLabel(token,label,elementIndex) ::= <<
<tokenRefTrack(...)>
<listLabel(elem=label,...)>
>>

/** ^(ID ...) track for rewrite */
tokenRefRuleRootTrack(token,label,elementIndex) ::= <<
<! <super.tokenRef(...)> !>
<tokenRefBang(...)>
<if(backtracking)>
if ( !<actions.(actionScope).synpredgate> ) <endif>
  [stream_<token> addElement:<label>];<\n>
>>

/** Match ^(label+=TOKEN ...) track for rewrite */
tokenRefRuleRootTrackAndListLabel(token,label,elementIndex) ::= <<
<tokenRefRuleRootTrack(...)>
<listLabel(elem=label,...)>
>>

/** rule when output=AST and tracking for rewrite */
ruleRefTrack(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>
[stream_<rule.name> addElement:[<label> getTree]];
>>

/** x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefTrack(...)>
<listLabel(elem={[<label> getTree]},...)>
>>

/** ^(rule ...) rewrite */

```

```

ruleRefRuleRootTrack(rule,label,elementIndex,args,scope) ::= <<
<! <super.ruleRefRuleRoot(...)> !>
<ruleRefRuleRoot(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>
  [stream_<rule.name> addElement:[<label> getTree]];<\n>
>>

/** ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRootTrack(...)>
<listLabel(elem={ [<label> getTree] },...)>
>>

// R e w r i t e

rewriteCode(
alts, description,
referencedElementsDeep, // ALL referenced elements to right of ->
referencedTokenLabels,
referencedTokenListLabels,
referencedRuleLabels,
referencedRuleListLabels,
referencedWildcardLabels,
referencedWildcardListLabels,
rewriteBlockLevel, enclosingTreeLevel, treeLevel) ::=
<<

// AST REWRITE
// elements: <referencedElementsDeep; separator=", ">
// token labels: <referencedTokenLabels; separator=", ">
// rule labels: <referencedRuleLabels; separator=", ">
// token list labels: <referencedTokenListLabels; separator=", ">
// rule list labels: <referencedRuleListLabels; separator=", ">
// wildcard labels: <[referencedWildcardLabels,referencedWildcardListLabels]; separator=", ">
<if(backtracking)>
if ( <actions.(actionScope).synpredgate> ) {<\n>
<endif>
<prevRuleRootRef()>.tree = root_0;<\n>
<rewriteCodeLabels()>
root_0 = (<ASTLabelType> *)[[treeAdaptor class] newEmptyTree] retain];<\n>
<alts:rewriteAlt(); separator="else ">
<! if tree parser and rewrite=true !>
<if(TREE_PARSER)>
<if(rewriteMode)>
<prevRuleRootRef()>.tree = (<ASTLabelType>)[treeAdaptor rulePostProcessing:root_0];
[input replaceChildren:[treeAdaptor getParent:retval.start]
  From:[treeAdaptor getChildIndex:retval.start]
  To:[treeAdaptor getChildIndex:_last]

```

```

        With:retval.tree];
<endif>
<endif>
<! if parser or tree-parser && rewrite!=true, we need to set result !>
<if(!TREE_PARSER)>
<prevRuleRootRef(>.tree = root_0;<\n>
<else>
<if(!rewriteMode)>
<prevRuleRootRef(>.tree = root_0;<\n>
<endif>
<endif>
<endif>
<if(backtracking)>
}
<endif>
>>

rewriteCodeLabels() ::= <<
<referencedTokenLabels
: {it | RewriteRule<rewriteElementType>Stream *stream_<it> =
[[RewriteRule<rewriteElementType>Stream newRewriteRule<rewriteElementType>Stream:treeAdaptor
description:@ "token <it>" element:<it>] retain];};
separator="\n"
>
<referencedTokenListLabels: {it | RewriteRule<rewriteElementType>Stream *stream_<it> =
[[RewriteRule<rewriteElementType>Stream newRewriteRule<rewriteElementType>Stream:treeAdaptor
description:@ "token <it>" elements:list_<it>] retain];};
separator="\n"
>
<referencedWildcardLabels: {it | RewriteRuleSubtreeStream stream_<it> =
[[RewriteRuleSubtreeStream newRewriteRuleSubtreeStream:treeAdaptor
description:"wildcard <it>" element:<it>] retain];};
separator="\n"
>
<referencedWildcardListLabels: {it | RewriteRuleSubtreeStream stream_<it> =
[[RewriteRuleSubtreeStream newRewriteRuleSubtreeStream:treeAdaptor
descriptor:"wildcard <it>" elements:list_<it>] retain];};
separator="\n"
>
<referencedRuleLabels: {it | RewriteRuleSubtreeStream *stream_<it> =
[[RewriteRuleSubtreeStream newRewriteRuleSubtreeStream:treeAdaptor
description:@ "token <it>" element:<it>!=nil?[<it> getTree]:nil] retain];};
separator="\n"
>
<referencedRuleListLabels: {it | RewriteRuleSubtreeStream *stream_<it> =
[[RewriteRuleSubtreeStream newRewriteRuleSubtreeStream:treeAdaptor
description:@ "token <it>" elements:list_<it>] retain];};
separator="\n"
>

```

```

>>

/** Generate code for an optional rewrite block; note it uses the deep ref'd element
 * list rather shallow like other blocks.
 */
rewriteOptionalBlock(
  alt,rewriteBlockLevel,
  referencedElementsDeep, // all nested refs
  referencedElements, // elements in immediately block; no nested blocks
  description) ::=
<<
// <fileName>:<description>
if ( <referencedElementsDeep:{el | [stream_<el> hasNext]}; separator="||"> ) {
  <alt>
}
<referencedElementsDeep:{el | [stream_<el> reset];<\n>}>
>>

rewriteClosureBlock(
  alt,rewriteBlockLevel,
  referencedElementsDeep, // all nested refs
  referencedElements, // elements in immediately block; no nested blocks
  description) ::=
<<
// <fileName>:<description>
while ( <referencedElements:{el | [stream_<el> hasNext]}; separator="||"> ) {
  <alt>
}
<referencedElements:{el | [stream_<el> reset];<\n>}>
>>

rewritePositiveClosureBlock(
  alt,rewriteBlockLevel,
  referencedElementsDeep, // all nested refs
  referencedElements, // elements in immediately block; no nested blocks
  description) ::=
<<
// <fileName>:<description>
{
if ( !(<referencedElements:{el | [stream_<el> hasNext]}; separator=" || ">) ) {
  @throw [RewriteEarlyExitException newException];
}
while ( <referencedElements:{el | [stream_<el> hasNext]}; separator=" || "> ) {
  <alt>
}
<referencedElements:{el | [stream_<el> reset];<\n>}>
}
}
>>

```

```

rewriteAlt(a) ::= <<
// <a.description>
<if(a.pred)>
if (<a.pred>) {
    <a.alt>
}<\n>
<else>
{
    <a.alt>
}<\n>
<endif>
>>

/** For empty rewrites: "r : ... -> ;" */
rewriteEmptyAlt() ::= "root_0 = nil;"

rewriteTree(root,children,description,enclosingTreeLevel,treeLevel) ::= <<
// <fileName>:<description>
{
    <ASTLabelType> *root_<treeLevel> = (<ASTLabelType> *)[[treeAdaptor class] newEmptyTree] retain];
    <root:rewriteElement()>
    <children:rewriteElement()>
    [treeAdaptor addChild:root_<treeLevel> toTree:root_<enclosingTreeLevel>];
}<\n>
>>

rewriteElementList(elements) ::= "<elements:rewriteElement()>"

rewriteElement(e) ::= <<
<@pregen()>
<e.el>
>>

/** Gen ID or ID[args] */
rewriteTokenRef(token,elementIndex,terminalOptions,args) ::= <<
// TODO: args: <args; separator=", ">
[treeAdaptor addChild:<createRewriteNodeFromElement(...)> toTree:root_<treeLevel>];<\n>
>>

/** Gen $label ... where defined via label=ID */
rewriteTokenLabelRef(label,elementIndex) ::= <<
[treeAdaptor addChild:[stream_<label> nextNode] toTree:root_<treeLevel>];<\n>
>>

/** Gen $label ... where defined via label+=ID */
rewriteTokenListLabelRef(label,elementIndex) ::= <<
[treeAdaptor addChild:[stream_<label> nextNode] toTree:root_<treeLevel>];<\n>

```

>>

```
/** Gen ^($label ...) */
```

```
rewriteTokenLabelRefRoot(label,elementIndex) ::= <<  
root_<treeLevel> = (<ASTLabelType> *)[treeAdaptor becomeRoot:[stream_<label> nextNode]  
old:root_<treeLevel>];<\n>
```

>>

```
/** Gen ^($label ...) where label+=... */
```

```
rewriteTokenListLabelRefRoot ::= rewriteTokenLabelRefRoot
```

```
/** Gen ^($ID ...) or ^($ID[$args] ...) */
```

```
rewriteTokenRefRoot(token,elementIndex,terminalOptions,args) ::= <<  
root_<treeLevel> = (<ASTLabelType> *)[treeAdaptor becomeRoot:<createRewriteNodeFromElement(...)>  
old:root_<treeLevel>];<\n>
```

>>

```
rewriteImaginaryTokenRef(args,token,terminalOptions,elementIndex) ::= <<
```

```
[treeAdaptor addChild:<createImaginaryNode(tokenType=token, ...)> toTree:root_<treeLevel>];<\n>
```

>>

```
rewriteImaginaryTokenRefRoot(args,token,terminalOptions,elementIndex) ::= <<
```

```
root_<treeLevel> = (<ASTLabelType> *)[treeAdaptor becomeRoot:<createImaginaryNode(tokenType=token, ...)>  
old:root_<treeLevel>];<\n>
```

>>

```
/** plain -> {foo} action */
```

```
rewriteAction(action) ::= <<
```

```
root_0 = <action>;<\n>
```

>>

```
/** What is the name of the previous value of this rule's root tree? This
```

```
* let's us refer to $rule to mean previous value. I am reusing the
```

```
* variable 'tree' sitting in retval struct to hold the value of root_0 right
```

```
* before I set it during rewrites. The assign will be to retval.tree.
```

```
*/
```

```
prevRuleRootRef() ::= "retval"
```

```
rewriteRuleRef(rule) ::= <<
```

```
[treeAdaptor addChild:[stream_<rule> nextTree] toTree:root_<treeLevel>];<\n>
```

>>

```
rewriteRuleRefRoot(rule) ::= <<
```

```
root_<treeLevel> = (<ASTLabelType> *)[treeAdaptor becomeRoot:(id\<Tree>)[stream_<rule> nextNode]  
old:root_<treeLevel>];<\n>
```

>>

```
rewriteNodeAction(action) ::= <<
```

```

[treeAdaptor addChild:<action> toTree:root_<treeLevel>];<\n>
>>

rewriteNodeActionRoot(action) ::= <<
root_<treeLevel> = (<ASTLabelType> *)[treeAdaptor becomeRoot:<action> old:root_<treeLevel>];<\n>
>>

/** Gen $ruleLabel ... where defined via ruleLabel=rule */
rewriteRuleLabelRef(label) ::= <<
[treeAdaptor addChild:[stream_<label> nextTree] toTree:root_<treeLevel>];<\n>
>>

/** Gen $ruleLabel ... where defined via ruleLabel+=rule */
rewriteRuleListLabelRef(label) ::= <<
[treeAdaptor addChild:[stream_<label> nextTree] toTree:root_<treeLevel>];<\n>
>>

/** Gen ^($ruleLabel ...) where ruleLabel=rule */
rewriteRuleLabelRefRoot(label) ::= <<
root_<treeLevel> = (<ASTLabelType> *)[treeAdaptor becomeRoot:[stream_<label> nextNode]
old:root_<treeLevel>];<\n>
>>

/** Gen ^($ruleLabel ...) where ruleLabel+=rule */
rewriteRuleListLabelRefRoot(label) ::= <<
root_<treeLevel> = (<ASTLabelType> *)[treeAdaptor becomeRoot:[stream_<label> nextNode]
old:root_<treeLevel>];<\n>
>>

rewriteWildcardLabelRef(label) ::= <<
[treeAdaptor addChild:[stream_<label> nextTree] toTree:root_<treeLevel>];<\n>
>>

createImaginaryNode(tokenType,terminalOptions,args) ::= <<
<if(terminalOptions.node)>
  [<terminalOptions.node> new<terminalOptions.node>:<tokenType> <if(args)>, <args; separator=", "><endif>]
<else>
  <if(args)>
    [[treeAdaptor createTree:<tokenType> <if(first(args))>FromToken:<first(args)><endif>
<if(first(rest(args)))>Text:<first(rest(args))><else>Text:@ "<tokenType>"<endif>] retain]
  <else>
    [[treeAdaptor createTree:<tokenType> Text:@ "<tokenType>"] retain]
  <endif>
<endif>
>>

createRewriteNodeFromElement(token,terminalOptions,args) ::= <<
<if(terminalOptions.node)>

```

```

[<terminalOptions.node> new<terminalOptions.node>:[stream_<token> nextToken]<if(args)>, <args;
separator=", "><endif>]
<else>
  <if(args)> <! must create new node from old !>
    [[treeAdaptor createTree:<token> Text:<first(rest(args))> <args; separator=", ">] retain]
  <else>
    [stream_<token> nextNode]
  <endif>
<endif>
>>

```

Found in path(s):

```

* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/ObjC/AST.stg
No license file was found, but licenses were detected in source scan.

```

```

/*
 * [The "BSD license"]
 * Copyright (c) 2011 Terence Parr
 * All rights reserved.
 *
 * Conversion to C#:
 * Copyright (c) 2011 Sam Harwell, Pixel Mine, Inc.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * 3. The name of the author may not be used to endorse or promote products
 * derived from this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
 * IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
 * IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
 * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
 * DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
 * THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
 * THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */

```

```

@outputFile.imports() ::= <<

```

```

<@super.imports(>

<if(!TREE_PARSER)>
<! tree parser would already have imported !>
using Antlr.Runtime.Tree;
using RewriteRuleITokenStream = Antlr.Runtime.Tree.RewriteRuleTokenStream;
<endif>
>>

@genericParser.members() ::= <<
<@super.members(>
<parserMembers(>
>>

parserCtorBody() ::= <<
<super.parserCtorBody(>
<treeAdaptorType(> treeAdaptor = default(<treeAdaptorType(>>);
CreateTreeAdaptor(ref treeAdaptor);
TreeAdaptor = treeAdaptor<if(!actions.(actionScope).treeAdaptorType)> ?? new CommonTreeAdaptor(<endif>;
>>

/** Add an adaptor property that knows how to build trees */
parserMembers() ::= <<
// Implement this function in your helper file to use a custom tree adaptor
partial void CreateTreeAdaptor(ref <treeAdaptorType(> adaptor);

private <treeAdaptorType(> adaptor;

public <treeAdaptorType(> TreeAdaptor
{
    get
    {
        return adaptor;
    }

    set
    {
        this.adaptor = value;
        <grammar.directDelegates: {g|<g:delegateName(>.TreeAdaptor = this.adaptor;}>
    }
}
>>

treeAdaptorType() ::= <<
<actions.(actionScope).treeAdaptorType; null="ITreeAdaptor">
>>

ruleReturnBaseType() ::= <%

```

```

Ast<if(TREE_PARSER)>Tree<else>Parser<endif>RuleReturnScope\<<ASTLabelType>, <labelType>>
%>

/** Add a variable to track rule's return AST */
ruleDeclarations() ::= <<
<super.ruleDeclarations()>
<ASTLabelType> root_0 = default(<ASTLabelType>);<\n>
>>

ruleLabelDefs(ruleDescriptor, labelType, ASTLabelType, rewriteElementType) ::= <%
<super.ruleLabelDefs(...)>
<if(!ruleDescriptor.isSynPred)>
<[ruleDescriptor.tokenLabels,ruleDescriptor.wildcardTreeLabels,ruleDescriptor.wildcardTreeListLabels]
: {it|<\n><ASTLabelType> <it.label.text>_tree = default(<ASTLabelType>);}>
<ruleDescriptor.tokenListLabels: {it|<\n><ASTLabelType> <it.label.text>_tree = default(<ASTLabelType>);}>
<ruleDescriptor.allTokenRefsInAltsWithRewrites
: {it|<\n>RewriteRule<rewriteElementType>Stream stream_<it>=new
RewriteRule<rewriteElementType>Stream(adaptor,"token <it>");}>
<ruleDescriptor.allRuleRefsInAltsWithRewrites
: {it|<\n>RewriteRuleSubtreeStream stream_<it>=new RewriteRuleSubtreeStream(adaptor,"rule <it>");}>
<endif>
%>

/** When doing auto AST construction, we must define some variables;
 * These should be turned off if doing rewrites. This must be a "mode"
 * as a rule could have both rewrite and AST within the same alternative
 * block.
 */
@alt.declarations() ::= <<
<if(autoAST && outerAlt && !rewriteMode && !ruleDescriptor.isSynPred)>
root_0 = (<ASTLabelType>)adaptor.Nil();
<endif>
>>

// Tracking Rule Elements

/** ID and track it for use in a rewrite rule */
tokenRefTrack(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRefBang(...)> <! Track implies no auto AST construction!>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) <endif>stream_<token>.Add(<label>);<\n>
>>

/** ids+=ID and track it for use in a rewrite rule; adds to ids *and*
 * to the tracking list stream_ID for use in the rewrite.
 */
tokenRefTrackAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRefTrack(...)>
<listLabelElem(elem=label,elemType=labelType,...)>

```

```

>>

/** ^(ID ...) track for rewrite */
tokenRefRuleRootTrack(token,label,elementIndex,terminalOptions={ }) ::= <<
<tokenRefBang(...)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) <endif>stream_<token>.Add(<label>);
>>

/** Match ^(label+=TOKEN ...) track for rewrite */
tokenRefRuleRootTrackAndListLabel(token,label,elementIndex,terminalOptions={ }) ::= <<
<tokenRefRuleRootTrack(...)>
<listLabelElem(elem=label,elemType=labelType,...)>
>>

/** rule when output=AST and tracking for rewrite */
ruleRefTrack(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) <endif>stream_<rule.name>.Add(<label>.Tree);
>>

/** x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefTrack(...)>
<listLabelElem(elem={<label>.Tree},elemType=ASTLabelType,...)>
>>

/** ^(rule ...) rewrite */
ruleRefRuleRootTrack(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) <endif>stream_<rule>.Add(<label>.Tree);
>>

/** ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRootTrack(...)>
<listLabelElem(elem={<label>.Tree},elemType=ASTLabelType,...)>
>>

// R e w r i t e

rewriteCode(
alts, description,
referencedElementsDeep, // ALL referenced elements to right of ->
referencedTokenLabels,
referencedTokenListLabels,
referencedRuleLabels,
referencedRuleListLabels,
referencedWildcardLabels,

```

```

referencedWildcardListLabels,
rewriteBlockLevel, enclosingTreeLevel, treeLevel) ::= <<
<\n>{
// AST REWRITE
// elements: <referencedElementsDeep; separator=", ">
// token labels: <referencedTokenLabels; separator=", ">
// rule labels: <referencedRuleLabels; separator=", ">
// token list labels: <referencedTokenListLabels; separator=", ">
// rule list labels: <referencedRuleListLabels; separator=", ">
// wildcard labels: <[referencedWildcardLabels,referencedWildcardListLabels]; separator=", ">
<if(backtracking)>
if (<actions.(actionScope).synpredgate>) {
<endif>
<prevRuleRootRef().Tree = root_0;
<rewriteCodeLabels()>
root_0 = (<ASTLabelType>)adaptor.Nil();
<alts:rewriteAlt(); separator="else ">
<! if tree parser and rewrite=true !>
<if(TREE_PARSER&&rewriteMode)>
<prevRuleRootRef().Tree = (<ASTLabelType>)adaptor.RulePostProcessing(root_0);
if (<prevRuleRootRef().Tree != null)
input.ReplaceChildren(adaptor.GetParent(retval.Start), adaptor.GetChildIndex(retval.Start),
adaptor.GetChildIndex(_last), retval.Tree);
<endif>
<! if parser or tree-parser && rewrite!=true, we need to set result !>
<if(!TREE_PARSER||!rewriteMode)>
<prevRuleRootRef().Tree = root_0;
<endif>
<endif>
}
<endif>
}
>>

rewriteCodeLabels() ::= <<
<referencedTokenLabels
:{it|RewriteRule<rewriteElementType>Stream stream_<it>=new
RewriteRule<rewriteElementType>Stream(adaptor,"token <it>",<it>)};
separator="\n"
>
<referencedTokenListLabels
:{it|RewriteRule<rewriteElementType>Stream stream_<it>=new
RewriteRule<rewriteElementType>Stream(adaptor,"token <it> ", list_<it>)};
separator="\n"
>
<referencedWildcardLabels
:{it|RewriteRuleSubtreeStream stream_<it>=new RewriteRuleSubtreeStream(adaptor,"wildcard <it>",<it>)};

```

```

separator="\n"
>
<referencedWildcardListLabels
: {it|RewriteRuleSubtreeStream stream_<it>=new RewriteRuleSubtreeStream(adaptor,"wildcard <it>","list_<it>");}
separator="\n"
>
<referencedRuleLabels
: {it|RewriteRuleSubtreeStream stream_<it>=new RewriteRuleSubtreeStream(adaptor,"rule
<it>",<it>!=null?<it>.Tree:null);}
separator="\n"
>
<referencedRuleListLabels
: {it|RewriteRuleSubtreeStream stream_<it>=new RewriteRuleSubtreeStream(adaptor,"token <it>","list_<it>");}
separator="\n"
>
>>

/** Generate code for an optional rewrite block; note it uses the deep ref'd element
 * list rather shallow like other blocks.
 */
rewriteOptionalBlock(
alt,rewriteBlockLevel,
referencedElementsDeep, // all nested refs
referencedElements, // elements in immediately block; no nested blocks
description) ::=
<<
// <fileName>:<description>
if (<referencedElementsDeep:{el | stream_<el>.HasNext}; separator="||">)
{
<alt>
}
<referencedElementsDeep:{el | stream_<el>.Reset();<n>}>
>>

rewriteClosureBlock(
alt,rewriteBlockLevel,
referencedElementsDeep, // all nested refs
referencedElements, // elements in immediately block; no nested blocks
description) ::=
<<
// <fileName>:<description>
while ( <referencedElements:{el | stream_<el>.HasNext}; separator="||"> )
{
<alt>
}
<referencedElements:{el | stream_<el>.Reset();<n>}>
>>

```

```

rewritePositiveClosureBlock(
  alt,rewriteBlockLevel,
  referencedElementsDeep, // all nested refs
  referencedElements, // elements in immediately block; no nested blocks
  description) ::=
<<
if (!(<referencedElements:{el | stream_<el>.HasNext}; separator="||">))
{
  throw new RewriteEarlyExitException();
}
while ( <referencedElements:{el | stream_<el>.HasNext}; separator="||"> )
{
  <alt>
}
<referencedElements:{el | stream_<el>.Reset();<\n>}>
>>

```

```

rewriteAlt(a) ::= <<
// <a.description>
<if(a.pred)>
if (<a.pred>)
{
  <a.alt>
}
<else>
{
  <a.alt>
}
<endif>
>>

```

```

/** For empty rewrites: "r : ... -> ;" */
rewriteEmptyAlt() ::= "root_0 = null;"

```

```

rewriteTree(root,children,description,enclosingTreeLevel,treeLevel) ::= <<
// <fileName>:<description>
{
  <ASTLabelType> root_<treeLevel> = (<ASTLabelType>)adaptor.Nil();
  <root:rewriteElement()>
  <children:rewriteElement()>
  adaptor.AddChild(root_<enclosingTreeLevel>, root_<treeLevel>);
}<\n>
>>

```

```

rewriteElementList(elements) ::= "<elements:rewriteElement()>"

```

```

rewriteElement(e) ::= <%
<@pregen()>

```

```

DebugLocation(<e.line>, <e.pos>);<\n>
<e.el>
%>

/** Gen ID or ID[args] */
rewriteTokenRef(token,elementIndex,args,terminalOptions={}) ::= <<
adaptor.AddChild(root_<treeLevel>, <createRewriteNodeFromElement(...)>);<\n>
>>

/** Gen $label ... where defined via label=ID */
rewriteTokenLabelRef(label,elementIndex) ::= <<
adaptor.AddChild(root_<treeLevel>, stream_<label>.NextNode());<\n>
>>

/** Gen $label ... where defined via label+=ID */
rewriteTokenListLabelRef(label,elementIndex) ::= <<
adaptor.AddChild(root_<treeLevel>, stream_<label>.NextNode());<\n>
>>

/** Gen ^($label ...) */
rewriteTokenLabelRefRoot(label,elementIndex) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(stream_<label>.NextNode(), root_<treeLevel>);<\n>
>>

/** Gen ^($label ...) where label+=... */
rewriteTokenListLabelRefRoot ::= rewriteTokenLabelRefRoot

/** Gen ^(ID ...) or ^(ID[args] ...) */
rewriteTokenRefRoot(token,elementIndex,args,terminalOptions={}) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(<createRewriteNodeFromElement(...)>,
root_<treeLevel>);<\n>
>>

rewriteImaginaryTokenRef(args,token,elementIndex,terminalOptions={}) ::= <<
adaptor.AddChild(root_<treeLevel>, <createImaginaryNode(tokenType=token, ...)>);<\n>
>>

rewriteImaginaryTokenRefRoot(args,token,elementIndex,terminalOptions={}) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(<createImaginaryNode(tokenType=token, ...)>,
root_<treeLevel>);<\n>
>>

/** plain -> {foo} action */
rewriteAction(action) ::= <<
root_0 = <action>;<\n>
>>

/** What is the name of the previous value of this rule's root tree? This

```

```

* let's us refer to $rule to mean previous value. I am reusing the
* variable 'tree' sitting in retval struct to hold the value of root_0 right
* before I set it during rewrites. The assign will be to retval.tree.
*/
prevRuleRootRef() ::= "retval"

rewriteRuleRef(rule) ::= <<
adaptor.AddChild(root_<treeLevel>, stream_<rule>.NextTree());<\n>
>>

rewriteRuleRefRoot(rule) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(stream_<rule>.NextNode(), root_<treeLevel>);<\n>
>>

rewriteNodeAction(action) ::= <<
adaptor.AddChild(root_<treeLevel>, <action>);<\n>
>>

rewriteNodeActionRoot(action) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(<action>, root_<treeLevel>);<\n>
>>

/** Gen $ruleLabel ... where defined via ruleLabel=rule */
rewriteRuleLabelRef(label) ::= <<
adaptor.AddChild(root_<treeLevel>, stream_<label>.NextTree());<\n>
>>

/** Gen $ruleLabel ... where defined via ruleLabel+=rule */
rewriteRuleListLabelRef(label) ::= <<
adaptor.AddChild(root_<treeLevel>, stream_<label>.NextTree());<\n>
>>

/** Gen ^($ruleLabel ...) where ruleLabel=rule */
rewriteRuleLabelRefRoot(label) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(stream_<label>.NextNode(), root_<treeLevel>);<\n>
>>

/** Gen ^($ruleLabel ...) where ruleLabel+=rule */
rewriteRuleListLabelRefRoot(label) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(stream_<label>.NextNode(), root_<treeLevel>);<\n>
>>

rewriteWildcardLabelRef(label) ::= <<
adaptor.AddChild(root_<treeLevel>, stream_<label>.NextTree());<\n>
>>

createImaginaryNode(tokenType,args,terminalOptions={ }) ::= <%
<if(terminalOptions.node)>

```

```

<! new MethodNode(IDLabel, args) !>
new <terminalOptions.node><(tokenType)<if(args)>, <args; separator=", "><endif>
<else>
(<ASTLabelType>)adaptor.Create(<tokenType>, <args; separator=", "><if(!args)>"<tokenType>"<endif>
<endif>
%>

```

```

createRewriteNodeFromElement(token,args,terminalOptions={}) ::= <%
<if(terminalOptions.node)>
new <terminalOptions.node>(stream_<token>.NextToken())<if(args)>, <args; separator=", "><endif>
<else>
<if(args)> <! must create new node from old !>
adaptor.Create(<token>, <args; separator=", ">)
<else>
stream_<token>.NextNode()
<endif>
<endif>
%>

```

Found in path(s):

```

* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-
jar/org/antlr/codegen/templates/CSharp3/AST.stg

```

No license file was found, but licenses were detected in source scan.

```

/*

```

```

[The "BSD license"]

```

```

Copyright (c) 2005-2006 Terence Parr

```

```

All rights reserved.

```

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT

(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```
*/
/** Template subgroup to add template rewrite output
 * If debugging, then you'll also get STDbg.stg loaded.
 */

@outputFile.imports() ::= <<
<@super.imports(>
import stringtemplate3
>>

/** Add this to each rule's return value struct */
@returnScope.ruleReturnInit() ::= <<
self.st = None
>>

@returnScope.ruleReturnMembers() ::= <<
def getTemplate(self):
    return self.st

def toString(self):
    if self.st is not None:
        return self.st.toString()
    return None
__str__ = toString

>>

@genericParser.init() ::= <<
<@super.init(>
self.templateLib = stringtemplate3.StringTemplateGroup(
    '<name>Templates', lexer='angle-bracket'
)
)

>>

@genericParser.members() ::= <<
<@super.members(>
def setTemplateLib(self, templateLib):
    self.templateLib = templateLib

def getTemplateLib(self):
    return self.templateLib

>>

/** x+=rule when output=template */
```

```

ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(label, {<label>.st})>
>>

```

```

rewriteTemplate(alts) ::= <<
# TEMPLATE REWRITE
<if(backtracking)>
if <actions.(actionScope).synpredgate>:
  <first(alts):rewriteTemplateAltFirst(>
  <rest(alts):{it | el<rewriteTemplateAlt(it)>}>
  <if(rewriteMode)><replaceTextInLine(><endif>

<else>
<first(alts):rewriteTemplateAltFirst(>
<rest(alts):{it | el<rewriteTemplateAlt(it)>}>
<if(rewriteMode)><replaceTextInLine(><endif>
<endif>
>>

```

```

replaceTextInLine() ::= <<
<if(TREE_PARSER)>
self.input.getTokenStream().replace(
  self.input.getTreeAdaptor().getTokenStartIndex(retval.start),
  self.input.getTreeAdaptor().getTokenStopIndex(retval.start),
  retval.st
)
<else>
self.input.replace(
  retval.start.getTokenIndex(),
  self.input.LT(-1).getTokenIndex(),
  retval.st
)
<endif>
>>

```

```

rewriteTemplateAltFirst(alt) ::= <<
<if(alt.pred)>
if <alt.pred>:
  # <alt.description>
  retval.st = <alt.alt>
<\n>
<else>
# <alt.description>
retval.st = <alt.alt>
<\n>
<endif>
>>

```

```

rewriteTemplateAlt(alt) ::= <<
<if(alt.pred)>if <alt.pred>:
  # <alt.description>
  retval.st = <alt.alt>
<\n>
<else>se:
  # <alt.description>
  retval.st = <alt.alt>
<\n>
<endif>
>>

```

```

rewriteEmptyTemplate(alts) ::= <<
None
>>

```

/** Invoke a template with a set of attribute name/value pairs.

* Set the value of the rule's template *after* having set
* the attributes because the rule's template might be used as
* an attribute to build a bigger template; you get a self-embedded
* template.
*/

```

rewriteExternalTemplate(name,args) ::= <%
self.templateLib.getInstanceOf("<name><if(args)>, attributes={<args:{a | "<a.name>": <a.value>}; separator=",
">}<endif>
%>

```

/** expr is a string expression that says what template to load */

```

rewriteIndirectTemplate(expr,args) ::= <%
self.templateLib.getInstanceOf(<expr><if(args)>, attributes={<args:{a | "<a.name>": <a.value>}; separator=",
">}<endif>
%>

```

/** Invoke an inline template with a set of attribute name/value pairs */

```

rewriteInlineTemplate(args, template) ::= <%
stringtemplate3.StringTemplate("<template>", group=self.templateLib<if(args)>, attributes={<args:{a |
"<a.name>": <a.value>}; separator=", ">}<endif>
%>

```

/** plain -> {foo} action */

```

rewriteAction(action) ::= <<
<action>
>>

```

/** An action has %st.attrName=expr; or % {st}.attrName=expr; */

```

actionSetAttribute(st,attrName,expr) ::= <<
(<st>)["<attrName>"] = <expr>

```

>>

```
/** Translate %{stringExpr} */  
actionStringConstructor(stringExpr) ::= <<  
stringtemplate3.StringTemplate(<stringExpr>, group=self.templateLib)  
>>
```

Found in path(s):

* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/Python/ST.stg
No license file was found, but licenses were detected in source scan.

```
/*  
* [The "BSD license"]  
* Copyright (c) 2007-2008 Johannes Luber  
* Copyright (c) 2005-2007 Kunle Odutola  
* Copyright (c) 2011 Sam Harwell  
* Copyright (c) 2011 Terence Parr  
* All rights reserved.  
*  
* Redistribution and use in source and binary forms, with or without  
* modification, are permitted provided that the following conditions  
* are met:  
* 1. Redistributions of source code must retain the above copyright  
* notice, this list of conditions and the following disclaimer.  
* 2. Redistributions in binary form must reproduce the above copyright  
* notice, this list of conditions and the following disclaimer in the  
* documentation and/or other materials provided with the distribution.  
* 3. The name of the author may not be used to endorse or promote products  
* derived from this software without specific prior written permission.  
*  
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR  
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES  
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.  
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,  
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT  
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,  
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY  
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT  
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF  
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.  
*/  
/** Template overrides to add debugging to normal Java output;  
* If ASTs are built, then you'll also get ASTDbg.stg loaded.  
*/  
  
@outputFile.debugPreprocessor() ::= "#define ANTLR_DEBUG"  
  
@outputFile.imports() ::= <<
```

```

<@super.imports()>
using Antlr.Runtime.Debug;
using IOException = System.IO.IOException;
>>

@genericParser.members() ::= <<
<if(grammar.grammarIsRoot)>
public static readonly string[] ruleNames =
new string[]
{
    "invalidRule", <grammar.allImportedRules:{rST | "<rST.name>"}; wrap="\n ", separator=", ">
};<\n>
<endif>
<if(grammar.grammarIsRoot)><! grammar imports other grammar(s) !>
int ruleLevel = 0;
public virtual int RuleLevel { get { return ruleLevel; } }
public virtual void IncRuleLevel() { ruleLevel++; }
public virtual void DecRuleLevel() { ruleLevel--; }
<if(profile)>
<ctorForProfilingRootGrammar()>
<else>
<ctorForRootGrammar()>
<endif>
<ctorForPredefinedListener()>
<else><! imported grammar !>
public int RuleLevel { get { return <grammar.delegators:{g| <g.delegateName()>>.RuleLevel; } } }
public void IncRuleLevel() { <grammar.delegators:{g| <g.delegateName()>>.IncRuleLevel(); } }
public void DecRuleLevel() { <grammar.delegators:{g| <g.delegateName()>>.DecRuleLevel(); } }
<ctorForDelegateGrammar()>
<endif>
<if(profile)>
public override bool AlreadyParsedRule( IIntStream input, int ruleIndex )
{
    int stopIndex = GetRuleMemoization(ruleIndex, input.Index);
    ((Profiler)dbg).ExamineRuleMemoization(input, ruleIndex, stopIndex,
    <grammar.composite.rootGrammar.recognizerName>.ruleNames[ruleIndex]);
    return base.AlreadyParsedRule(input, ruleIndex);
}<\n>
public override void Memoize( IIntStream input, int ruleIndex, int ruleStartIndex )
{
    ((Profiler)dbg).Memoize(input, ruleIndex, ruleStartIndex,
    <grammar.composite.rootGrammar.recognizerName>.ruleNames[ruleIndex]);
    base.Memoize(input, ruleIndex, ruleStartIndex);
}<\n>
<endif>
protected virtual bool EvalPredicate( bool result, string predicate )
{
    dbg.SemanticPredicate( result, predicate );
}

```

```

return result;
}<\n>
>>

```

```

ctorForRootGrammar() ::= <<
<! bug: can't use <@super.members()> cut-n-paste instead !>
<! Same except we add port number and profile stuff if root grammar !>
<actions.(actionScope).ctorModifier; null="public"> <name>( <inputStreamType> input )
: this( input, DebugEventSocketProxy.DefaultDebuggerPort, new RecognizerSharedState() )
{
}
<actions.(actionScope).ctorModifier; null="public"> <name>( <inputStreamType> input, int port,
RecognizerSharedState state )
: base( input, state )
{
<createListenerAndHandshake()>
<grammar.directDelegates:{g|<g:delegateName()> = new <g.recognizerName>( input, dbg, this.state,
this<grammar.delegates:{g|, <g:delegateName()>}> );}; separator="\n">
<parserCtorBody()>
<@finally()>
}<\n>
>>

```

```

ctorForProfilingRootGrammar() ::= <<
<! bug: can't use <@super.members()> cut-n-paste instead !>
<actions.(actionScope).ctorModifier; null="public"> <name>( <inputStreamType> input )
: this( input, new Profiler(null), new RecognizerSharedState() )
{
}
<actions.(actionScope).ctorModifier; null="public"> <name>( <inputStreamType> input, IDebugEventListener dbg,
RecognizerSharedState state )
: base( input, dbg, state )
{
Profiler p = (Profiler)dbg;
p.setParser(this);
<grammar.directDelegates:
{g|<g:delegateName()> = new <g.recognizerName>( input, dbg, this.state, this<grammar.delegates:{g|,
<g:delegateName()>}> );}; separator="\n">
<parserCtorBody()>
<@finally()>
}
}<\n>
>>

```

/** Basically we don't want to set any dbg listeners are root will have it. */

```

ctorForDelegateGrammar() ::= <<
<actions.(actionScope).ctorModifier; null="public"> <name>( <inputStreamType> input, IDebugEventListener dbg,
RecognizerSharedState state<grammar.delegates:{g|, <g.recognizerName> <g:delegateName()>}> )

```

```

: base( input, dbg, state )
{
<grammar.directDelegates:
  {g|<g:delegateName()> = new <g.recognizerName>( input, this, this.state<grammar.delegates:{ g|,
<g:delegateName()>> );}; separator="\n">
  <parserCtorBody()>
}<\n>
>>

ctorForPredefinedListener() ::= <<
<actions.(actionScope).ctorModifier; null="public"> <name>( <inputStreamType> input, IDebugEventListener dbg
)
<@superClassRef>: base( input, dbg, new RecognizerSharedState() )<@end>
{
<if(profile)>
  Profiler p = (Profiler)dbg;
  p.setParser(this);
<endif>
<grammar.directDelegates:{ g|<g:delegateName()> = new <g.recognizerName>(input, dbg, this.state,
this<grammar.delegates:{ g|, <g:delegateName()>>);}; separator="\n">
  <parserCtorBody()>
<@finally()>
}<\n>
>>

createListenerAndHandshake() ::= <<
<if(TREE_PARSER)>
DebugEventSocketProxy proxy = new DebugEventSocketProxy( this, port, input.TreeAdaptor );<\n>
<else>
DebugEventSocketProxy proxy = new DebugEventSocketProxy( this, port, null );<\n>
<endif>
DebugListener = proxy;
try
{
  proxy.Handshake();
}
catch ( IOException ioe )
{
  ReportError( ioe );
}
>>

@genericParser.superClassName() ::= "Debug<@super.superClassName()>"

/*
* Many of the following rules were merged into CSharp2.stg.
*/

```

```

@rule.preamble() ::= <<
if (RuleLevel == 0)
    DebugListener.Commence();
IncRuleLevel();
>>

//@rule.preamble() ::= <<
//try
//{{
// dbg.EnterRule( GrammarFileName, "<ruleName>" );
// if ( RuleLevel == 0 )
// {
//   dbg.Commence();
// }
// IncRuleLevel();
// dbg.Location( <ruleDescriptor.tree.line>, <ruleDescriptor.tree.charPositionInLine> );<\n>
//>>

@rule.postamble() ::= <<
DecRuleLevel();
if (RuleLevel == 0)
    DebugListener.Terminate();
>>

//@rule.postamble() ::= <<
//dbg.Location(<ruleDescriptor.EORNode.line>, <ruleDescriptor.EORNode.charPositionInLine>);<\n>
//}
//finally
//{{
// dbg.ExitRule( GrammarFileName, "<ruleName>" );
// DecRuleLevel();
// if ( RuleLevel == 0 )
// {
//   dbg.Terminate();
// }
//}<\n>
//>>

//@insertSynpreds.start() ::= "dbg.BeginBacktrack( state.backtracking );"
//@insertSynpreds.stop() ::= "dbg.EndBacktrack( state.backtracking, success );"

// Common debug event triggers used by region overrides below

//enterSubRule() ::= <<
//try
//{{
// dbg.EnterSubRule( <decisionNumber> );<\n>
//>>

//exitSubRule() ::= <<

```

```

//}
//finally
//{{
// dbg.ExitSubRule( <decisionNumber> );
//}<\n>
//>>

//enterDecision() ::= <<
//try
//{{
// dbg.EnterDecision( <decisionNumber> );<\n>
//>>

//exitDecision() ::= <<
//}
//finally
//{{
// dbg.ExitDecision( <decisionNumber> );
//}<\n>
//>>

//enterAlt(n) ::= "dbg.EnterAlt( <n> );<\n>"

// Region overrides that tell various constructs to add debugging triggers

//@block.predecision() ::= "<enterSubRule()><enterDecision()>"

//@block.postdecision() ::= "<exitDecision()>"

//@block.postbranch() ::= "<exitSubRule()>"

//@ruleBlock.predecision() ::= "<enterDecision()>"

//@ruleBlock.postdecision() ::= "<exitDecision()>"

//@ruleBlockSingleAlt.preal() ::= "<enterAlt(n=\"1\")>"

//@blockSingleAlt.preal() ::= "<enterAlt(n=\"1\")>"

//@positiveClosureBlock.preloop() ::= "<enterSubRule()>"

//@positiveClosureBlock.postloop() ::= "<exitSubRule()>"

//@positiveClosureBlock.predecision() ::= "<enterDecision()>"

//@positiveClosureBlock.postdecision() ::= "<exitDecision()>"

//@positiveClosureBlock.earlyExitException() ::=

```

```

// "dbg.RecognitionException( eee<decisionNumber> );<n>"

//@closureBlock.preloop() ::= "<enterSubRule>"

//@closureBlock.postloop() ::= "<exitSubRule>"

//@closureBlock.predecision() ::= "<enterDecision>"

//@closureBlock.postdecision() ::= "<exitDecision>"

//@altSwitchCase.preal() ::= "<enterAlt(n=i)>"

//@element.prematch() ::=
// "dbg.Location( <it.line>, <it.pos> );"

//@matchSet.mismatchedSetException() ::=
// "dbg.RecognitionException( mse );"

//@dfaState.noViableAltException() ::= "dbg.RecognitionException( nvae );"

//@dfaStateSwitch.noViableAltException() ::= "dbg.RecognitionException( nvae );"

dfaDecision(decisionNumber,description) ::= <<
//try
//{
// isCyclicDecision = true;
// <super.dfaDecision(...)>
//}
//catch ( NoViableAltException nvae )
//{
// dbg.RecognitionException( nvae );
// throw nvae;
//}
//>>

//@cyclicDFA.errorMethod() ::= <<
//public override void Error( NoViableAltException nvae )
//{
// ((DebugParser)recognizer).dbg.RecognitionException( nvae );
//}
//>>

/** Force predicate validation to trigger an event */
evalPredicate(pred,description) ::= <<
EvalPredicate(<pred>, "<description>")
>>

```

Found in path(s):

```
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-  
jar/org/antlr/codegen/templates/CSharp2/Dbg.stg
```

No license file was found, but licenses were detected in source scan.

```
/*
```

```
[The "BSD license"]
```

```
Copyright (c) 2007 Kay Roepke 2010 Alan Condit
```

```
All rights reserved.
```

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

```
THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR  
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES  
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.  
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,  
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT  
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,  
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY  
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT  
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF  
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

```
*/
```

```
/** Templates for building ASTs during normal parsing.
```

```
*
```

```
* Deal with many combinations. Dimensions are:
```

```
* Auto build or rewrite
```

```
* no label, label, list label (label/no-label handled together)
```

```
* child, root
```

```
* token, set, rule, wildcard
```

```
*
```

```
* The situation is not too bad as rewrite (->) usage makes ^ and !
```

```
* invalid. There is no huge explosion of combinations.
```

```
*/
```

```
@rule.setErrorReturnValue() ::= <<
```

```
/* ASTParser rule.setErrorReturnValue */
```

```
retval.tree = (<ASTLabelType> *)[treeAdaptor errorNode:input From:retval.start To:[input LT:-1] Exception:re];
```

```
<! System.out.println("<ruleName> returns "+((CommonTree)retval.tree).toStringTree()); !>
```

```
>>
```

```

// TOKEN AST STUFF

/** ID and output=AST */
tokenRef(token, label, elementIndex, terminalOptions) ::= <<
/* ASTParser tokenRef */
<super.tokenRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<label>_tree = <createNodeFromToken(...)>;
[treeAdaptor addChild:<label>_tree toTree:root_0];
<if(backtracking)>}<endif>
>>

/* ID! and output=AST (same as plain tokenRef) */
/* ASTParser tokenRefBang */
tokenRefBang(token,label,elementIndex,terminalOptions) ::= "<super.tokenRef(...)>"

/** ID^ and output=AST */
tokenRefRuleRoot(token,label,elementIndex,terminalOptions) ::= <<
<super.tokenRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<label>_tree = <createNodeFromToken(...)>;
root_0 = (<ASTLabelType> *)[treeAdaptor becomeRoot:<label>_tree old:root_0];
<if(backtracking)>}<endif>
>>

/** ids+=ID! and output=AST */
tokenRefBangAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
/* ASTParser tokenRefBangAndListLabel */
<tokenRefBang(...)>
<listLabel(elem=label,...)>
>>

/** label+=TOKEN when output=AST but not rewrite alt */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
/* ASTParser tokenRefAndListLabel */
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

/** Match label+=TOKEN^ when output=AST but not rewrite alt */
tokenRefRuleRootAndListLabel(token,label,terminalOptions,elementIndex) ::= <<
/* ASTParser tokenRefRuleRootAndListLabel */
<tokenRefRuleRoot(...)>
<listLabel(elem=label,...)>
>>

// SET AST

```

```

// the match set stuff is interesting in that it uses an argument list
// to pass code to the default matchSet; another possible way to alter
// inherited code. I don't use the region stuff because I need to pass
// different chunks depending on the operator. I don't like making
// the template name have the operator as the number of templates gets
// large but this is the most flexible--this is as opposed to having
// the code generator call matchSet then add root code or ruleroot code
// plus list label plus ... The combinations might require complicated
// rather than just added on code. Investigate that refactoring when
// I have more time.

matchSet(s,label,terminalOptions,elementIndex,postmatchCode) ::= <%
/* ASTParser matchSet */
<super.matchSet(postmatchCode={<if(backtracking)>if ( <actions.(actionScope).synpredgate> )<endif>
 [treeAdaptor addChild:<createNodeFromToken(...)> toTree:root_0 ];}, ...)>
%>

matchRuleBlockSet(s,label,terminalOptions,elementIndex,postmatchCode,treeLevel="0") ::= <<
/* ASTParser matchRuleBlockSet */
<matchSet(...)>
>>

matchSetBang(s,label,elementIndex,terminalOptions, postmatchCode) ::= "<super.matchSet(...)>"

// note there is no matchSetTrack because -> rewrites force sets to be
// plain old blocks of alts: (A|B|...|C)

matchSetRuleRoot(s,label,terminalOptions,elementIndex,debug) ::= <<
/* ASTParser matchSetRuleRoot */
<if(label)>
<label>=(<labelType> *)[input LT:1]; /* matchSetRuleRoot */<\n>
<endif>
<super.matchSet(postmatchCode={<if(backtracking)>if ( <actions.(actionScope).synpredgate> )<endif>
root_0 = (<ASTLabelType> *)[treeAdaptor becomeRoot:<createNodeFromToken(...)> old:root_0];}, ...)>
>>

// RULE REF AST

/** rule when output=AST */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
/* ASTParser ruleRef */
<super.ruleRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> )<endif>
[treeAdaptor addChild:[<label> getTree] toTree:root_0];
>>

/** rule! is same as normal rule ref */

```

```

ruleRefBang(rule,label,elementIndex,args,scope) ::= "<super.ruleRef(...)>"

/** rule^ */
ruleRefRuleRoot(rule,label,elementIndex,args,scope) ::= <<
/* ASTParser ruleRefRuleRoot */
<super.ruleRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> )<endif>
root_0 = (<ASTLabelType> *)[treeAdaptor becomeRoot:[<label> getTree] old:root_0];
>>

/** x+=rule when output=AST */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
/* ASTParser ruleRefAndListLabel */
<ruleRef(...)>
<listLabel(elem = {[<label> getTree]},...)>
>>

/** x+=rule! when output=AST is a rule ref with list addition */
ruleRefBangAndListLabel(rule,label,elementIndex,args,scope) ::= <<
/* ASTParser ruleRefBangAndListLabel */
<ruleRefBang(...)>
<listLabel(elem = {[<label> getTree]},...)>
>>

/** x+=rule^ */
ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope) ::= <<
/* ASTParser ruleRefRuleRootAndListLabel */
<ruleRefRuleRoot(...)>
<listLabel(elem = {[<label> getTree]},...)>
>>

// WILDCARD AST

wildcard(token,label,elementIndex,terminalOptions) ::= <<
/* ASTParser wildcard */
<super.wildcard(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
  [treeAdaptor addChild:[[treeAdaptor create:<label>] retain] toTree:root_0];
<if(backtracking)>}<endif>
>>

wildcardBang(token,label,elementIndex,terminalOptions) ::= "<super.wildcard(...)>"

wildcardRuleRoot(token,label,elementIndex,terminalOptions) ::= <<
/* ASTParser wildcardRuleRoot */
<super.wildcard(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
  <label>_tree = [[treeAdaptor create:<label>] retain]

```

```

    root_0 = (<ASTLabelType> *)[treeAdaptor becomeRoot:<label>_tree old:root_0];
<if(backtracking)></endif>
>>

createNodeFromToken(label,terminalOptions) ::= <<
/* ASTParser createNodeFromToken */
<if(terminalOptions.node)>
[<terminalOptions.node> new<terminalOptions.node>:<label>] <! new MethodNode(IDLabel) !>
<else>
(<ASTLabelType> *)[treeAdaptor create:<label>] retain]
<endif>
>>

// straight from java cleanup ///
ruleCleanUp() ::= <<
/* ASTParser ruleCleanUp */
<super.ruleCleanUp()>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<\n><endif>
    retval.tree = (<ASTLabelType> *)[treeAdaptor rulePostProcessing:root_0];
    [treeAdaptor setTokenBoundaries:retval.tree From:retval.start To:retval.stopToken];
<if(backtracking)></endif>
>>

```

Found in path(s):

```
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-
jar/org/antlr/codegen/templates/ObjC/ASTParser.stg
```

No license file was found, but licenses were detected in source scan.

```
/*
```

```
[The "BSD license"]
```

```
Copyright (c) 2005-2006 Terence Parr
```

```
All rights reserved.
```

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,

INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

```
/** Template subgroup to add template rewrite output
 * If debugging, then you'll also get STDbg.stg loaded.
 */
```

```
@returnScopeInterface.memVars() ::= <<
<@super.memVars()>
/* ST returnInterface.memVars */
ST *st;
>>
```

```
@returnScopeInterface.properties() ::= <<
<@super.properties()>
/* ST returnScope.properties */
@property (retain, getter=getST, setter=setST:) ST *st;
>>
```

```
@returnScopeInterface.methodsDecl() ::= <<
<@super.methodsDecl()>
/* ST AST returnScopeInterface.methodsDecl */
- (id) getTemplate;
- (NSString *) toString;
- (NSString *) description;
>>
```

```
@returnScopeInterface() ::= <<
/* ST returnScopeInterface */
@interface <returnScopeInterface.name> : ReturnScope {
<returnScopeInterface.memVars()>
}
<returnScopeInterface.properties()>

<returnScopeInterface.methods()>
@end
>>
```

```
@returnScopeImplementation.synthesize() ::= <<
<@super.synthesize()>
/* ST returnScope.synthesize */
@synthesize st;
>>
```

```

@returnScopeImplementation.methods() ::= <<
<@super.methods()>
/* ST returnScope.methods */
- (id) getTemplate { return st; }
- (NSString *) toString { return st==nil?nil:[st toString]; }
- (NSString *) description { [self toString]; }
>>

```

```

@returnScopeImplementation() ::= <<
@implementation <returnScopeImplementation.name>
<returnScopeImplementation.synthesize()>

```

```

<returnScopeImplementation.methods()>
@end
>>

```

```

/** Add this to each rule's return value struct */
@returnScope.ruleReturnMembers() ::= <<
<@super.ruleReturnMembers()>
/* ST returnScope.ruleReturnMembers -- empty */
>>

```

```

@genericParserHeaderFile.memVars() ::= <<
<@super.memVars()>
/* ST genericParserHeaderFile.memVars -- empty now */
STGroup *templateLib; /* ST -- really a part of STAttrMap */
>>

```

```

@genericParserHeaderFile.properties() ::= <<
<@super.properties()>
/* ST genericParser.properties */
@property (retain, getter=getTemplateLib, setter=setTemplateLib:) STGroup *templateLib;
>>

```

```

@genericParserHeaderFile.methodsDecl() ::= <<
<@super.methodsDecl()>
/* ST genericParser.methodsDecl */
- init;
- (STGroup *) getTemplateLib;
- (void) setTemplateLib:(STGroup *)aTemplateLib;
@end
>>

```

```

@genericParser.synthesize() ::= <<
<@super.synthesize()>
/* ST genericParserImplementation.synthesize */
@synthesize templateLib;
>>

```

```

@genericParser.methods() ::= <<
<@super.methods()>
/* ST genericParser.methods */

- (STGroup *)getTemplateLib
{
    return templateLib;
}

- (void) setTemplateLib:(STGroup *)aTemplateLib
{
    templateLib = aTemplateLib;
}

>>

@genericParser.members() ::= <<
<@super.members()>
STGroup *templateLib = [STGroup newSTGroup:@"<name>Templates"];

- (STGroup *) getTemplateLib
{
    return templateLib;
}

- (void) setTemplateLib:(STGroup *) templateLib
{
    this.templateLib = templateLib;
}

/** allows convenient multi-value initialization:
 * "new STAttrMap().put(...).put(...)"
 */
/* REPLACE THIS STATIC CLASS
static class STAttrMap extends HashMap {
- (STAttrMap *) setObject:(id)aValue forKey:(NS*)String attrName
{
    [super setObject:value forKey:attrName];
    return self;
}
- (STAttrMap *) setObjectWithInt:(NSInteger)value forKey:(NSString *)attrName
{
    [super setObject:[NSNumber numberWithInt:value] forKey:attrName];
    return self;
}
}
*/

```

```

>>

@STAttrMap() ::= <<
/* ----- ST start STAttrMap ----- */
<@STAttrMap.interface()>
<@STAttrMap.implementation()>
/* ----- ST end STAttrMap ----- */
>>

@STAttrMap.interface() ::= <<
/* ----- ST start STAttrMap.interface ----- */
@interface STAttrMap : HashMap {
/* <@STAttrMap.memVars()> */
    STGroup *templateLib;
}

/* <@STAttrMap.properties()> */
@property (retain, getter=getTemplateLib, setter=setTemplateLib:) STGroup *templateLib;
/* <@STAttrMap.methodsDecl()> */
- (id) init;
- (STAttrMap *) setObject:(id)value forKey:(NSString *)attrName;
- (STAttrMap *) setObjectWithInt:(NSInteger)value forKey:(NSString *)attrName;
- (void) setTemplateLib:(STGroup *)aTemplateLib;
- (STGroup *) getTemplateLib;
@end
/* ----- ST end STAttrMap.interface ----- */
>>

@STAttrMap.implementation() ::= <<
/* ----- ST start STAttrMap.implementation ----- */
/** allows convenient multi-value initialization:
 * "new STAttrMap().put(...).put(...)"
 */
@implementation STAttrMap
@synthesize templateLib;

<@STAttrMap.methods()>
@end
/* ----- ST end STAttrMap.implementation ----- */
>>

@STAttrMap.memVars() ::= <<
/* ----- ST start STAttrMap.memVars ----- */
    STGroup *templateLib;
/* ----- ST end STAttrMap.memVars ----- */
>>

@STAttrMap.properties() ::= <<

```

```

/* ----- ST start STAttrMap.properties ----- */
@property (retain, getter=getTemplateLib, setter=setTemplateLib:) STGroup *templateLib;
/* ----- ST end STAttrMap.properties ----- */
>>

@STAttrMap.methodsDecl() ::= <<
/* ----- ST start STAttrMap.methodsDecl ----- */
- (id) init;
- (STAttrMap *) setObject:(id)value forKey:(NSString *)attrName;
- (STAttrMap *) setObjectWithInt:(NSInteger)value forKey:(NSString *)attrName;
- (void) setTemplateLib:(STGroup *)aTemplateLib;
- (STGroup *) getTemplateLib;
/* ----- ST end STAttrMap.methodsDecl ----- */
>>

@STAttrMap.methods() ::= <<
/* ----- ST start STAttrMap.methods ----- */
- (id) init
{
    self = [super initWithLen:16];
    if ( self != nil ) {
        templateLib = [STGroup newSTGroup:"<name>Templates"];
    }
    return self;
}

- (STAttrMap *) setObject:(id)aValue forKey:(NSString *)aAttrName
{
    [super setObject:aValue forKey:aAttrName];
    return self;
}

- (STAttrMap *) setObjectWithInt:(NSInteger)aValue forKey:(NSString *)aAttrName
{
    [super setObject:[NSNumber numberWithInt:aValue] forKey:aAttrName];
    return self;
}

- (void) setTemplateLib:(STGroup *)aTemplateLib
{
    templateLib = aTemplateLib;
}

- (STGroup *)getTemplateLib
{
    return templateLib;
}
/* ----- ST end STAttrMap.methods ----- */
>>

```

```

/** x+=rule when output=template */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
/* ST ruleRefAndListLable */
<ruleRef(...)>
<listLabel(elem=[label getTemplate,...]>
>>

rewriteTemplate(alts) ::= <<
/* ----- ST start rewriteTemplate ----- */
// TEMPLATE REWRITE
<if(backtracking)>
if ( <actions.(actionScope).synpredgate> ) {
  <alts:rewriteTemplateAlt(); separator="else ">
  <if(rewriteMode)><replaceTextInLine()><endif>
}
<else>
<alts:rewriteTemplateAlt(); separator="else ">
<if(rewriteMode)><replaceTextInLine()><endif>
<endif>
/* ----- ST end rewriteTemplate ----- */
>>

replaceTextInLine() ::= <<
/* ----- ST start replaceTextInLine ----- */
<if(TREE_PARSER)>
[[ (TokenRewriteStream *)input getTokenStream]
  replaceFromIndex: [[input getTreeAdaptor] getTokenStartIndex:retval.start]
    ToIndex: [[input getTreeAdaptor] getTokenStopIndex:retval.start]
    Text:retval.st];
<else>
[[ (TokenRewriteStream *)input]
  replaceFromIndex: [[(CommonToken *)retval.start] getTokenIndex]
    ToIndex: [[input LT:-1] getTokenIndex]
    Text:retval.st];
<endif>
/* ----- ST end replaceTextInLine ----- */
>>

rewriteTemplateAlt() ::= <<
/* ----- ST start rewriteTemplateAlt ----- */
/* ST <it.description> */
<if(it.pred)>
if (<it.pred> ) {
  retval.st = <it.alt>;
}<\n>
<else>
{

```

```

    retval.st = <it.alt>;
}<\n>
<endif>
/* ----- ST end rewriteTemplateAlt ----- */
>>

rewriteEmptyTemplate(alts) ::= <<
nil;
>>

/** Invoke a template with a set of attribute name/value pairs.
 * Set the value of the rule's template after having set
 * the attributes because the rule's template might be used as
 * an attribute to build a bigger template; you get a self-embedded
 * template.
 */
rewriteExternalTemplate(name,args) ::= <<
/* ----- ST start rewriteExternalTemplate ----- */
[templateLib getInstanceOf:@"<name>"
<if(args)>[[STAttrMap newSTAttrMap] <args:{a | setObject:<a.value> forKey:@"<a.name>"}><endif>]
/* ----- ST end rewriteExternalTemplate ----- */
>>

/** expr is a string expression that says what template to load */
rewriteIndirectTemplate(expr,args) ::= <<
/* ----- ST start rewriteIndirectTemplate ----- */
[templateLib getInstanceOf:<expr>
<if(args)> [[STAttrMap newSTAttrMap]<args:{a | setObject:<a.value> forKey:@"<a.name>"}>]
<else>]<endif>
/* ----- ST end rewriteIndirectTemplate ----- */
>>

/** Invoke an inline template with a set of attribute name/value pairs */
rewriteInlineTemplate(args, template) ::= <<
/* ----- ST start rewriteInlineTemplate ----- */
STGroup *templateLib;
templateLib.templates = [STAttrMap newSTAttrMap];
<if(args)> [templateLib.templates <args:{a | setObject:<a.value> forKey:@"<a.name>"}>]<endif>
[ST newST:templateLib template:@"<template>"];
/* ----- ST end rewriteInlineTemplate ----- */
>>

/** plain -> {foo} action */
rewriteAction(action) ::= <<
/* ----- ST start rewriteAction ----- */
<action>
/* ----- ST end rewriteAction ----- */
>>

```

```

/** An action has %st.attrName=expr; or % {st}.attrName=expr; */
actionSetAttribute(st,attrName,expr) ::= <<
/* ----- ST start actionSetAttribute ----- */
[[ST attributes] setObject:<expr> forKey:@"<attrName>"];
<![<st> setAttribute:<expr> name:@"<attrName>"];!>
/* ----- ST end actionSetAttribute ----- */
>>

```

```

/** Translate %{stringExpr} */
actionStringConstructor(stringExpr) ::= <<
/* ----- ST start actionStringConstructor ----- */
[ST newSTWithTemplate:<stringExpr>]
/* ----- ST end actionStringConstructor ----- */
>>

```

Found in path(s):

* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/ObjC/ST.stg

No license file was found, but licenses were detected in source scan.

```

/*
[The "BSD license"]
Copyright (c) 2010 Terence Parr
All rights reserved.

```

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```
*/
```

```

javaTypeInitMap ::= [
    "int":"0",
    "long":"0",
    "float":"0.0f",
    "double":"0.0",
    "boolean":"false",
    "byte":"0",
    "short":"0",
    "char":"0",
    default:"null" // anything other than an atomic type
]

// System.Boolean.ToString() returns "True" and "False", but the proper C# literals are "true" and "false"
// The Java version of Boolean returns "true" and "false", so they map to themselves here.
booleanLiteral ::= [
    "True":"true",
    "False":"false",
    "true":"true",
    "false":"false",
    default:"false"
]

/** The overall file structure of a recognizer; stores methods for rules
 * and cyclic DFAs plus support code.
 */
outputFile(LEXER,PARSER,TREE_PARSER, actionScope, actions,
    docComment, recognizer,
    name, tokens, tokenNames, rules, cyclicDFAs,
    bitsets, buildTemplate, buildAST, rewriteMode, profile,
    backtracking, synpreds, memoize, numRules,
    fileName, ANTLRVersion, generatedTimestamp, trace,
    scopes, superClass, literals) ::=
<<
// $ANTLR <ANTLRVersion> <fileName> <generatedTimestamp>
<actions.(actionScope).header>

<@imports>
import org.antlr.runtime.*;
<if(TREE_PARSER)>
import org.antlr.runtime.tree.*;
<endif>
import java.util.Stack;
import java.util.List;
import java.util.ArrayList;
<if(backtracking)>
import java.util.Map;
import java.util.HashMap;
<endif>

```

```

<@end>

<docComment>
@SuppressWarnings("all")
<recognizer>

>>

lexer(grammar, name, tokens, scopes, rules, numRules, filterMode, labelType="CommonToken",
    superClass="Lexer") ::= <<
public class <grammar.recognizerName> extends <@superClassName><superClass><@end> {
    <tokens:{it | public static final int <it.name>=<it.type>;}; separator="\n">
    <scopes:{it |<if(it.isDynamicGlobalScope)><globalAttributeScope(it)><endif>}>
    <actions.lexer.members>

    // delegates
    <grammar.delegates:
    {g|public <g.recognizerName> <g.delegateName()>;}; separator="\n">
    // delegators
    <grammar.delegators:
    {g|public <g.recognizerName> <g.delegateName()>;}; separator="\n">
    <last(grammar.delegators):{g|public <g.recognizerName> gParent;}>
    public <superClass>[] getDelegates() {
        return new <superClass>[] {<grammar.delegates: {g|<g.delegateName()>;}; separator = ", ">};
    }

    public <grammar.recognizerName>() {} <! needed by subclasses !>
    public <grammar.recognizerName>(CharStream input<grammar.delegators:{g|, <g.recognizerName>
    <g.delegateName()>}>) {
        this(input, new RecognizerSharedState()<grammar.delegators:{g|, <g.delegateName()>}>);
    }
    public <grammar.recognizerName>(CharStream input, RecognizerSharedState state<grammar.delegators:{g|,
    <g.recognizerName> <g.delegateName()>}>) {
        super(input,state);
    }
    <if(memoize)>
    <if(grammar.grammarIsRoot)>
        state.ruleMemo = new HashMap[<numRules>+1];<\n><! index from 1..n !>
    <endif>
    <endif>

    <grammar.directDelegates:
    {g|<g.delegateName()> = new <g.recognizerName>(input, state<trunc(g.delegators):{p|, <p.delegateName()>}>,
this);}; separator="\n">
    <grammar.delegators:
    {g|this.<g.delegateName()> = <g.delegateName()>;}; separator="\n">
    <last(grammar.delegators):{g|gParent = <g.delegateName()>;}>
    }
    @Override public String getGrammarFileName() { return "<fileName>"; }

```

```

<if(filterMode)>
  <filteringNextToken()>
<endif>
<rules; separator="\n\n">

<synpreds:{p | <lexerSynpred(p)>}>

<cyclicDFAs:{dfa | protected DFA<dfa.decisionNumber> dfa<dfa.decisionNumber> = new
DFA<dfa.decisionNumber>(this);}; separator="\n">
<cyclicDFAs:cyclicDFA(); separator="\n\n"><! dump tables for all DFA !>

}
>>

```

```

/** A override of Lexer.nextToken() that backtracks over mTokens() looking
 * for matches. No error can be generated upon error; just rewind, consume
 * a token and then try again. backtracking needs to be set as well.
 * Make rule memoization happen only at levels above 1 as we start mTokens
 * at backtracking==1.
 */

```

```

filteringNextToken() ::= <<
@Override
public Token nextToken() {
  while (true) {
    if ( input.LA(1)==CharStream.EOF ) {
      Token eof = new CommonToken(input,Token.EOF,
        Token.DEFAULT_CHANNEL,
        input.index(),input.index());
      eof.setLine(getLine());
      eof.setCharPositionInLine(getCharPositionInLine());
      return eof;
    }
    state.token = null;
    state.channel = Token.DEFAULT_CHANNEL;
    state.tokenStartCharIndex = input.index();
    state.tokenStartCharPositionInLine = input.getCharPositionInLine();
    state.tokenStartLine = input.getLine();
    state.text = null;
    try {
      int m = input.mark();
      state.backtracking=1; <! means we won't throw slow exception !>
      state.failed=false;
      mTokens();
      state.backtracking=0;
      <! mTokens backtracks with synpred at backtracking==2
      and we set the synpredgate to allow actions at level 1. !>
      if ( state.failed ) {
        input.rewind(m);

```

```

    input.consume(); <! advance one char and try again !>
    }
    else {
        emit();
        return state.token;
    }
}
}
catch (RecognitionException re) {
    // shouldn't happen in backtracking mode, but...
    reportError(re);
    recover(re);
}
}
}

@Override
public void memoize(IntStream input,
    int ruleIndex,
    int ruleStartIndex)
{
    if ( state.backtracking>1 ) super.memoize(input, ruleIndex, ruleStartIndex);
}

@Override
public boolean alreadyParsedRule(IntStream input, int ruleIndex) {
    if ( state.backtracking>1 ) return super.alreadyParsedRule(input, ruleIndex);
    return false;
}
}
>>

actionGate() ::= "state.backtracking==0"

filteringActionGate() ::= "state.backtracking==1"

/** How to generate a parser */
genericParser(grammar, name, scopes, tokens, tokenNames, rules, numRules,
    bitsets, inputStreamType, superClass,
    labelType, members, rewriteElementType,
    filterMode, ASTLabelType="Object") ::= <<
public class <grammar.recognizerName> extends <@superClassName><superClass><@end> {
<if(grammar.grammarIsRoot)>
    public static final String[] tokenNames = new String[] {
        "<invalid>", "<EOR>", "<DOWN>", "<UP>", <tokenNames; separator=", ", wrap="\n\t">
    };
<endif>
    <tokens:{it |public static final int <it.name>=<it.type>;}; separator="\n">

    // delegates

```

```

<grammar.delegates: {g|public <g.recognizerName> <g.delegateName()>; separator="\n">
public <superClass>[] getDelegates() {
return new <superClass>[] {<grammar.delegates: {g|<g.delegateName()>; separator = ", ">}
}

// delegators
<grammar.delegators:
{g|public <g.recognizerName> <g.delegateName()>; separator="\n">
<last(grammar.delegators):{g|public <g.recognizerName> gParent;}>

<scopes:{it |<if(it.isDynamicGlobalScope)><globalAttributeScope(it)><endif>}>

<@members>
<! WARNING. bug in ST: this is cut-n-paste into Dbg.stg !>
public <grammar.recognizerName>(<inputStreamType> input<grammar.delegators:{g|, <g.recognizerName>
<g.delegateName()>}>) {
this(input, new RecognizerSharedState()<grammar.delegators:{g|, <g.delegateName()>}>);
}
public <grammar.recognizerName>(<inputStreamType> input, RecognizerSharedState
state<grammar.delegators:{g|, <g.recognizerName> <g.delegateName()>}>) {
super(input, state);
<parserCtorBody()>
<grammar.directDelegates:
{g|<g.delegateName()> = new <g.recognizerName>(input, state<trunc(g.delegators):{p|, <p.delegateName()>}>,
this);}; separator="\n">
<grammar.indirectDelegates:{g | <g.delegateName()> = <g.delegator.delegateName()>.<g.delegateName()>;
separator="\n">
<last(grammar.delegators):{g|gParent = <g.delegateName()>;}>
}
<@end>

@Override public String[] getTokenNames() { return
<grammar.composite.rootGrammar.recognizerName>.tokenNames; }
@Override public String getGrammarFileName() { return "<fileName>"; }

<members>

<rules; separator="\n\n">

<! generate rule/method definitions for imported rules so they
appear to be defined in this recognizer. !>
// Delegated rules
<grammar.delegatedRules:{ruleDescriptor|
public <returnType(ruleDescriptor)> <ruleDescriptor.name>(<ruleDescriptor.parameterScope:parameterScope()>)
throws <ruleDescriptor.throwsSpec; separator=", "> { <if(ruleDescriptor.hasReturnValue)>return
<endif><ruleDescriptor.grammar:delegateName()>.<ruleDescriptor.name>(<if(ruleDescriptor.parameterScope)><ru
leDescriptor.parameterScope.attributes:{a|<a.name>}>; separator=", "><endif>); \}}; separator="\n">

```

```

<synpreds:{p | <synpred(p)>}>

<cyclicDFAs:{ dfa | protected DFA<dfa.decisionNumber> dfa<dfa.decisionNumber> = new
DFA<dfa.decisionNumber>(this);}; separator="\n">
<cyclicDFAs:cyclicDFA(); separator="\n\n"><! dump tables for all DFA !>

<bitsets:{it | <bitset(name={FOLLOW_<it.name>_in_<it.inName><it.tokenIndex>},
words64=it.bits)>}; separator="\n">
}
>>

parserCtorBody() ::= <<
<if(memoize)>
<if(grammar.grammarIsRoot)>
this.state.ruleMemo = new HashMap[<length(grammar.allImportedRules)>+1];<\n><! index from 1..n !>
<endif>
<endif>
<grammar.delegators:
{g|this.<g:delegateName()> = <g:delegateName()>;}; separator="\n">
>>

parser(grammar, name, scopes, tokens, tokenNames, rules, numRules, bitsets,
ASTLabelType="Object", superClass="Parser", labelType="Token",
members={<actions.parser.members>}) ::= <<
<genericParser(grammar, name, scopes, tokens, tokenNames, rules, numRules,
bitsets, "TokenStream", superClass,
labelType, members, "Token",
false, ASTLabelType)>
>>

/** How to generate a tree parser; same as parser except the input
* stream is a different type.
*/
treeParser(grammar, name, scopes, tokens, tokenNames, globalAction, rules,
numRules, bitsets, filterMode, labelType={<ASTLabelType>}, ASTLabelType="Object",
superClass={<if(filterMode)><if(buildAST)>TreeRewriter<else>TreeFilter<endif><else>TreeParser<endif>},
members={<actions.treeparser.members>}
) ::= <<
<genericParser(grammar, name, scopes, tokens, tokenNames, rules, numRules,
bitsets, "TreeNodeStream", superClass,
labelType, members, "Node",
filterMode, ASTLabelType)>
>>

/** A simpler version of a rule template that is specific to the imaginary
* rules created for syntactic predicates. As they never have return values
* nor parameters etc..., just give simplest possible method. Don't do
* any of the normal memoization stuff in here either; it's a waste.

```

```

* As predicates cannot be inlined into the invoking rule, they need to
* be in a rule by themselves.
*/
synpredRule(ruleName, ruleDescriptor, block, description, nakedBlock) ::=
<<
// $ANTLR start <ruleName>
public final void <ruleName>_fragment(<ruleDescriptor.parameterScope:parameterScope(>>) throws
<ruleDescriptor	throwsSpec:{x|<x>}; separator=", "> {
    <ruleLabelDefs(>
    <if(trace)>
    traceIn("<ruleName>_fragment", <ruleDescriptor.index>);
    try {
        <block>
    }
    finally {
        traceOut("<ruleName>_fragment", <ruleDescriptor.index>);
    }
    <else>
    <block>
    <endif>
}
// $ANTLR end <ruleName>
>>

synpred(name) ::= <<
public final boolean <name>() {
    state.backtracking++;
    <@start(>
    int start = input.mark();
    try {
        <name>_fragment(); // can never throw exception
    } catch (RecognitionException re) {
        System.err.println("impossible: "+re);
    }
    boolean success = !state.failed;
    input.rewind(start);
    <@stop(>
    state.backtracking--;
    state.failed=false;
    return success;
}<\n>
>>

lexerSynpred(name) ::= <<
<synpred(name)>
>>

ruleMemoization(name) ::= <<

```

```

<if(memoize)>
if ( state.backtracking>0 && alreadyParsedRule(input, <ruleDescriptor.index>) ) {
<returnStatement((( <ruleReturnValue(>)))> }
<endif>
>>

/** How to test for failure and return from rule */
checkRuleBacktrackFailure() ::= <<
<if(backtracking)>if (state.failed) <returnStatement((( <ruleReturnValue(>)))><endif>
>>

/** This rule has failed, exit indicating failure during backtrack */
ruleBacktrackFailure() ::= <<
<if(backtracking)>if (state.backtracking>0) {state.failed=true;
<returnStatement((( <ruleReturnValue(>)))>}<endif>
>>

/** How to generate code for a rule. This includes any return type
 * data aggregates required for multiple return values.
 */
rule(ruleName,ruleDescriptor,block,emptyRule,description,exceptions,finally,memoize) ::= <<
<ruleAttributeScope(scope=ruleDescriptor.ruleScope)>
<returnScope(scope=ruleDescriptor.returnScope)>

// $ANTLR start "<ruleName>"
// <fileName>:<description>
<if(isPredefinedRewriteRule.(ruleName) && filterMode && buildAST)>
@Override
<endif>
public final <returnType(ruleDescriptor)> <ruleName>(<ruleDescriptor.parameterScope:parameterScope(>)>) throws
<ruleDescriptor.throwsSpec:{ x|<x>}; separator=", "> {
<if(trace)>traceIn("<ruleName>", <ruleDescriptor.index>);<endif>
<ruleScopeSetUp(>
<ruleDeclarations(>
<ruleLabelDefs(>
<ruleDescriptor.actions.init>
<@preamble(>
try {
<ruleMemoization(name=ruleName)>
<block>
<ruleCleanUp(>
<(ruleDescriptor.actions.after):execAction(>
}
<if(exceptions)>
<exceptions:{ e|<catch(decl=e.decl,action=e.action)><\n>}>
<else>
<if(!emptyRule)>
<if(actions.(actionScope).rulecatch)>

```

```

    <actions.(actionScope).rulecatch>
<else>
    catch (RecognitionException re) {
        reportError(re);
        recover(input,re);
        <@setErrorReturnValue()>
    }
<endif>
<endif>
<endif>
    finally {
        // do for sure before leaving
        <if(trace)>traceOut("<ruleName>", <ruleDescriptor.index>);<endif>
        <memoize()>
        <ruleScopeCleanUp()>
        <finally>
    }
    <@postamble()>
    <returnStatement((({ <ruleReturnValue()>}), false)>
}
// $ANTLR end "<ruleName>"
>>

returnStatement(returnValue, force=true) ::= <%
<if(!isEmptyString.(returnValue))>
    return <returnValue>;
<elseif(force)>
    return;
<endif>
%>

catch(decl,action) ::= <<
catch (<e.decl>) {
    <e.action>
}
>>

ruleDeclarations() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<returnType(ruleDescriptor)> retval = new <returnType(ruleDescriptor)>();
retval.start = input.LT(1);
<elseif(ruleDescriptor.returnScope)>
<ruleDescriptor.returnScope.attributes:{ a |
<a.type> <a.name> = <if(a.initValue)><a.initValue><else><initValue(a.type)><endif>;
}>
<endif>
<if(memoize)>
int <ruleDescriptor.name>_startIndex = input.index();

```

```

<endif>
>>

ruleScopeSetUp() ::= <<
<ruleDescriptor.useScopes:{it |<it>_stack.push(new <it>_scope());}; separator="\n">
<ruleDescriptor.ruleScope:{it |<it.name>_stack.push(new <it.name>_scope());}; separator="\n">
>>

ruleScopeCleanUp() ::= <<
<ruleDescriptor.useScopes:{it |<it>_stack.pop();}; separator="\n">
<ruleDescriptor.ruleScope:{it |<it.name>_stack.pop();}; separator="\n">
>>

ruleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,ruleDescriptor.tokenListLabels,
ruleDescriptor.wildcardTreeLabels,ruleDescriptor.wildcardTreeListLabels]
:{it |<labelType> <it.label.text>=null;}; separator="\n"
>
<[ruleDescriptor.tokenListLabels,ruleDescriptor.ruleListLabels,ruleDescriptor.wildcardTreeListLabels]
:{it |List<Object> list_<it.label.text>=null;}; separator="\n"
>
<ruleDescriptor.ruleLabels:ruleLabelDef(); separator="\n">
<ruleDescriptor.ruleListLabels:{ll|RuleReturnScope <ll.label.text> = null;}; separator="\n">
>>

lexerRuleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,
ruleDescriptor.tokenListLabels,
ruleDescriptor.ruleLabels]
:{it |<labelType> <it.label.text>=null;}; separator="\n"
>
<ruleDescriptor.charLabels:{it |int <it.label.text>;}; separator="\n">
<[ruleDescriptor.tokenListLabels,
ruleDescriptor.ruleListLabels]
:{it |List<Object> list_<it.label.text>=null;}; separator="\n"
>
>>

ruleReturnValue() ::= <%
<if(!ruleDescriptor.isSynPred)>
<if(ruleDescriptor.hasReturnValue)>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnName>
<else>
retval
<endif>
<endif>
<endif>

```

```

<endif>
%>

ruleCleanUp() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(!TREE_PARSER)>
retval.stop = input.LT(-1);
<endif>
<endif>
>>

memoize() ::= <<
<if(memoize)>
<if(backtracking)>
if ( state.backtracking>0 ) { memoize(input, <ruleDescriptor.index>, <ruleDescriptor.name>_StartIndex); }
<endif>
<endif>
>>

/** How to generate a rule in the lexer; naked blocks are used for
 * fragment rules.
 */
lexerRule(ruleName,nakedBlock,ruleDescriptor,block,memoize) ::= <<
// $ANTLR start "<ruleName>"
public final void m<ruleName>(<ruleDescriptor.parameterScope:parameterScope()>) throws RecognitionException
{
<if(trace)>traceIn("<ruleName>", <ruleDescriptor.index>);<endif>
<ruleScopeSetUp()>
<ruleDeclarations()>
try {
<if(nakedBlock)>
<ruleMemoization(name=ruleName)>
<lexerRuleLabelDefs()>
<ruleDescriptor.actions.init>
<block>
<else>
int _type = <ruleName>;
int _channel = DEFAULT_TOKEN_CHANNEL;
<ruleMemoization(name=ruleName)>
<lexerRuleLabelDefs()>
<ruleDescriptor.actions.init>
<block>
<ruleCleanUp()>
state.type = _type;
state.channel = _channel;
<(ruleDescriptor.actions.after):execAction()>
<endif>
}

```

```

finally {
    // do for sure before leaving
    <if(trace)>traceOut("<ruleName>", <ruleDescriptor.index>);<endif>
    <ruleScopeCleanUp()>
    <memoize()>
}
}
// $ANTLR end "<ruleName>"
>>

/** How to generate code for the implicitly-defined lexer grammar rule
 * that chooses between lexer rules.
 */
tokensRule(ruleName,nakedBlock,args,block,ruleDescriptor) ::= <<
@Override
public void mTokens() throws RecognitionException {
    <block>
}
>>

// S U B R U L E S

/** A (...) subrule with multiple alternatives */
block(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
// <fileName>:<description>
int alt<decisionNumber>=<maxAlt>;
<decls>
<@predecision()>
<decision>
<@postdecision()>
<@prebranch()>
switch (alt<decisionNumber>) {
    <alts:{a | <altSwitchCase(i,a)>}>
}
<@postbranch()>
>>

/** A rule block with multiple alternatives */
ruleBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
// <fileName>:<description>
int alt<decisionNumber>=<maxAlt>;
<decls>
<@predecision()>
<decision>
<@postdecision()>
switch (alt<decisionNumber>) {
    <alts:{a | <altSwitchCase(i,a)>}>
}

```

>>

```
ruleBlockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
// <fileName>:<description>
<decls>
<@prealt()>
<alts>
<@postalt()>
>>
```

*/** A special case of a (...) subrule with a single alternative */*

```
blockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
// <fileName>:<description>
<decls>
<@prealt()>
<alts>
<@postalt()>
>>
```

*/** A (..)+ block with 1 or more alternatives */*

```
positiveClosureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
// <fileName>:<description>
int cnt<decisionNumber>=0;
<decls>
<@preloop()>
loop<decisionNumber>:
while (true) {
int alt<decisionNumber>=<maxAlt>;
<@predecision()>
<decision>
<@postdecision()>
switch (alt<decisionNumber>) {
<alts:{ a | <altSwitchCase(i,a)> }>
default :
if ( cnt<decisionNumber> >= 1 ) break loop<decisionNumber>;
<ruleBacktrackFailure()>
EarlyExitException eee = new EarlyExitException(<decisionNumber>, input);
<@earlyExitException()>
throw eee;
}
cnt<decisionNumber>++;
}
<@postloop()>
>>
```

```
positiveClosureBlockSingleAlt ::= positiveClosureBlock
```

```

/** A (.)* block with 1 or more alternatives */
closureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::=
<<
// <fileName>:<description>
<decls>
<@preloop()>
loop<decisionNumber>:
while (true) {
int alt<decisionNumber>=<maxAlt>;
<@predecision()>
<decision>
<@postdecision()>
switch (alt<decisionNumber>) {
<alts:{ a | <altSwitchCase(i,a)>}>
default :
break loop<decisionNumber>;
}
}
<@postloop()>
>>

```

```
closureBlockSingleAlt ::= closureBlock
```

```

/** Optional blocks (x)? are translated to (x|) by before code generation
* so we can just use the normal block template
*/

```

```
optionalBlock ::= block
```

```
optionalBlockSingleAlt ::= block
```

```

/** A case in a switch that jumps to an alternative given the alternative
* number. A DFA predicts the alternative and then a simple switch
* does the jump to the code that actually matches that alternative.
*/

```

```

altSwitchCase(altNum,alt) ::= <<
case <altNum> :
<@prealt()>
<alt>
break;<\n>
>>

```

```

/** An alternative is just a list of elements; at outermost level */
alt(elements,altNum,description,autoAST,outerAlt,treeLevel,rew) ::= <<
// <fileName>:<description>
{
<@declarations()>
<elements:element()>
<rew>

```

```

<@cleanup(>
}
>>

/** What to emit when there is no rewrite. For auto build
 * mode, does nothing.
 */
noRewrite(rewriteBlockLevel, treeLevel) ::= ""

// E L E M E N T S

/** Dump the elements one per line */
element(e) ::= <<
<@prematch(>
<e.el>
>>

/** match a token optionally with a label in front */
tokenRef(token,label,elementIndex,terminalOptions={}) ::= <<
<if(label)><label>=<labelType><endif>match(input,<token>,FOLLOW_<token>_in_<ruleName><elementIndex
>); <checkRuleBacktrackFailure(>
>>

/** ids+=ID */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRef(token,label,elementIndex,terminalOptions)>
<listLabel(label, label)>
>>

listLabel(label,elem) ::= <<
if (list_<label>==null) list_<label>=new ArrayList<Object>();
list_<label>.add(<elem>);
>>

/** match a character */
charRef(char,label) ::= <<
<if(label)>
<label> = input.LA(1);
<endif>
match(<char>); <checkRuleBacktrackFailure(>
>>

/** match a character range */
charRangeRef(a,b,label) ::= <<
<if(label)>
<label> = input.LA(1);
<endif>
matchRange(<a>,<b>); <checkRuleBacktrackFailure(>

```

```

>>

/** For now, sets are interval tests and must be tested inline */
matchSet(s,label,elementIndex,postmatchCode="",terminalOptions={}) ::= <<
<if(label)>
<if(LEXER)>
<label>= input.LA(1);
<else>
<label>=<castToLabelType("input.LT(1)")>;
<endif>
<endif>
if ( <s> ) {
input.consume();
<postmatchCode>
<if(!LEXER)>
state.errorRecovery=false;
<endif>
<if(backtracking)>state.failed=false;<endif>
}
else {
<ruleBacktrackFailure()>
MismatchedSetException mse = new MismatchedSetException(null,input);
<@mismatchedSetException()>
<if(LEXER)>
recover(mse);
throw mse;
<else>
throw mse;
<! use following code to make it recover inline; remove throw mse;
recoverFromMismatchedSet(input,mse,FOLLOW_set_in_<ruleName><elementIndex>);
!>
<endif>
}
>>

```

```

matchRuleBlockSet ::= matchSet

```

```

matchSetAndListLabel(s,label,elementIndex,postmatchCode) ::= <<
<matchSet(...)>
<listLabel(label, label)>
>>

```

```

/** Match a string literal */
lexerStringRef(string,label,elementIndex="0") ::= <<
<if(label)>
int <label>Start = getCharIndex();
match(<string>); <checkRuleBacktrackFailure()>
int <label>StartLine<elementIndex> = getLine();

```

```

int <label>StartCharPos<elementIndex> = getCharPositionInLine();
<label> = new <labelType>(input, Token.INVALID_TOKEN_TYPE, Token.DEFAULT_CHANNEL, <label>Start,
getCharIndex()-1);
<label>.setLine(<label>StartLine<elementIndex>);
<label>.setCharPositionInLine(<label>StartCharPos<elementIndex>);
<else>
match(<string>); <checkRuleBacktrackFailure()>
<endif>
>>

wildcard(token,label,elementIndex,terminalOptions={ }) ::= <<
<if(label)>
<label>=<castToLabelType("input.LT(1)")>;
<endif>
matchAny(input); <checkRuleBacktrackFailure()>
>>

wildcardAndListLabel(token,label,elementIndex,terminalOptions={ }) ::= <<
<wildcard(...)>
<listLabel(label, label)>
>>

/** Match . wildcard in lexer */
wildcardChar(label, elementIndex) ::= <<
<if(label)>
<label> = input.LA(1);
<endif>
matchAny(); <checkRuleBacktrackFailure()>
>>

wildcardCharListLabel(label, elementIndex) ::= <<
<wildcardChar(label, elementIndex)>
<listLabel(label, label)>
>>

/** Match a rule reference by invoking it possibly with arguments
 * and a return value or values. The 'rule' argument was the
 * target rule name, but now is type Rule, whose toString is
 * same: the rule name. Now though you can access full rule
 * descriptor stuff.
 */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
pushFollow(FOLLOW_<rule.name>_in_<ruleName><elementIndex>);
<if(label)><label>=<endif><if(scope)><scope.delegateName()>.<endif><rule.name>(<args; separator=", ">);
state._fsp--;
<checkRuleBacktrackFailure()>
>>

```

```

/** ids+=r */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(rule,label,elementIndex,args,scope)>
<listLabel(label, label)>
>>

/** A lexer rule reference.
*
* The 'rule' argument was the target rule name, but now
* is type Rule, whose toString is same: the rule name.
* Now though you can access full rule descriptor stuff.
*/
lexerRuleRef(rule,label,args,elementIndex,scope) ::= <<
<if(label)>
int <label>Start<elementIndex> = getCharIndex();
int <label>StartLine<elementIndex> = getLine();
int <label>StartCharPos<elementIndex> = getCharPositionInLine();
<if(scope)><scope:delegateName().<endif>m<rule.name>(<args; separator=" ">);
<checkRuleBacktrackFailure()>
<label> = new <labelType>(input, Token.INVALID_TOKEN_TYPE, Token.DEFAULT_CHANNEL,
<label>Start<elementIndex>, getCharIndex()-1);
<label>.setLine(<label>StartLine<elementIndex>);
<label>.setCharPositionInLine(<label>StartCharPos<elementIndex>);
<else>
<if(scope)><scope:delegateName().<endif>m<rule.name>(<args; separator=" ">);
<checkRuleBacktrackFailure()>
<endif>
>>

/** i+=INT in lexer */
lexerRuleRefAndListLabel(rule,label,args,elementIndex,scope) ::= <<
<lexerRuleRef(rule,label,args,elementIndex,scope)>
<listLabel(label, label)>
>>

/** EOF in the lexer */
lexerMatchEOF(label,elementIndex) ::= <<
<if(label)>
int <label>Start<elementIndex> = getCharIndex();
int <label>StartLine<elementIndex> = getLine();
int <label>StartCharPos<elementIndex> = getCharPositionInLine();
match(EOF); <checkRuleBacktrackFailure()>
<labelType> <label> = new <labelType>(input, EOF, Token.DEFAULT_CHANNEL, <label>Start<elementIndex>,
getCharIndex()-1);
<label>.setLine(<label>StartLine<elementIndex>);
<label>.setCharPositionInLine(<label>StartCharPos<elementIndex>);
<else>
match(EOF); <checkRuleBacktrackFailure()>

```

```

<endif>
>>

// used for left-recursive rules
recRuleDefArg()          ::= "int <recRuleArg()>"
recRuleArg()            ::= "_p"
recRuleAltPredicate(ruleName,opPrec) ::= "<recRuleArg()> |<= <opPrec>"
recRuleSetResultAction() ::= "root_0=$<ruleName>_primary.tree;"
recRuleSetReturnAction(src,name)    ::= "$<name>=$<src>.<name>;"

/** match ^(root children) in tree parser */
tree(root, actionsAfterRoot, children, nullableChildList,
    enclosingTreeLevel, treeLevel) ::= <<
<root:element()>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if ( input.LA(1)==Token.DOWN ) {
    match(input, Token.DOWN, null); <checkRuleBacktrackFailure()>
    <children:element()>
    match(input, Token.UP, null); <checkRuleBacktrackFailure()>
}
<else>
match(input, Token.DOWN, null); <checkRuleBacktrackFailure()>
<children:element()>
match(input, Token.UP, null); <checkRuleBacktrackFailure()>
<endif>
>>

/** Every predicate is used as a validating predicate (even when it is
 * also hoisted into a prediction expression).
 */
validateSemanticPredicate(pred,description) ::= <<
if ( !(<evalPredicate(pred,description)>) ) {
    <ruleBacktrackFailure()>
    throw new FailedPredicateException(input, "<ruleName>", "<description>");
}
>>

// F i x e d D F A (if-then-else)

dfaState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
int LA<decisionNumber>_<stateNumber> = input.LA(<k>);
<edges; separator="\nelse ">
<if(!isTrue.(last(edges).labelExpr) && (!last(edges).predicates))>
else {
<if(eotPredictsAlt)>
alt<decisionNumber>=<eotPredictsAlt>;
<else>

```

```

<ruleBacktrackFailure()>
<(nvaExceptionWrapperMap.(k))({NoViableAltException nvae =
new NoViableAltException("<description>", <decisionNumber>, <stateNumber>, input);
<@noViableAltException()>
throw nvae;})>
<endif>
}
<endif>
>>

nvaExceptionWrapperMap ::= [
"1":"wrapNvaExceptionForK1",
"2":"wrapNvaExceptionForK2",
default:"wrapNvaExceptionForKN"
]

wrapNvaExceptionForK1(exceptionCode) ::= <<
<exceptionCode>
>>

wrapNvaExceptionForK2(exceptionCode) ::= <<
int nvaeMark = input.mark();
try {
input.consume();
<exceptionCode>
} finally {
input.rewind(nvaeMark);
}
>>

wrapNvaExceptionForKN(exceptionCode) ::= <<
int nvaeMark = input.mark();
try {
for (int nvaeConsume = 0; nvaeConsume << <k> - 1; nvaeConsume++) {
input.consume();
}
<exceptionCode>
} finally {
input.rewind(nvaeMark);
}
>>

/** Same as a normal DFA state except that we don't examine lookahead
* for the bypass alternative. It delays error detection but this
* is faster, smaller, and more what people expect. For (X)? people
* expect "if ( LA(1)==X ) match(X);" and that's it.
*/
dfaOptionalBlockState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<

```

```

int LA<decisionNumber>_<stateNumber> = input.LA(<k>);
<edges; separator="\nelse ">
>>

/** A DFA state that is actually the loopback decision of a closure
 * loop. If end-of-token (EOT) predicts any of the targets then it
 * should act like a default clause (i.e., no error can be generated).
 * This is used only in the lexer so that for ('a')* on the end of a rule
 * anything other than 'a' predicts exiting.
 */
dfaLoopbackState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
int LA<decisionNumber>_<stateNumber> = input.LA(<k>);
<edges; separator="\nelse ">
<if(eotPredictsAlt)>
<if(!edges)>
alt<decisionNumber>=<eotPredictsAlt>; <! if no edges, don't gen ELSE !>
<else>
else {
alt<decisionNumber>=<eotPredictsAlt>;
}
<endif>
<endif>
>>

/** An accept state indicates a unique alternative has been predicted */
dfaAcceptState(alt) ::= "alt<decisionNumber>=<alt>;"

/** A simple edge with an expression. If the expression is satisfied,
 * enter to the target state. To handle gated productions, we may
 * have to evaluate some predicates for this edge.
 */
dfaEdge(labelExpr, targetState, predicates) ::= <<
if ( (<labelExpr> <if(predicates)>&& (<predicates>)<endif> ) {
<targetState>
}
>>

// F i x e d D F A (switch case)

/** A DFA state where a SWITCH may be generated. The code generator
 * decides if this is possible: CodeGenerator.canGenerateSwitch().
 */
dfaStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ( input.LA(<k> ) ) {
<edges; separator="\n">
default:
<if(eotPredictsAlt)>
alt<decisionNumber>=<eotPredictsAlt>;

```

```

<else>
  <ruleBacktrackFailure()>
  <(nvaExceptionWrapperMap.(k))({NoViableAltException nvae =
    new NoViableAltException("<description>", <decisionNumber>, <stateNumber>, input);
  <@noViableAltException()>
  throw nvae;})>
<endif>
}
>>

dfaOptionalBlockStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ( input.LA(<k>)) {
  <edges; separator="\n">
}
>>

dfaLoopbackStateSwitch(k, edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ( input.LA(<k>)) {
  <edges; separator="\n">
  <if(eotPredictsAlt)>
  default:
    alt<decisionNumber>=<eotPredictsAlt>;
    break;
  <endif>
}
>>

dfaEdgeSwitch(labels, targetState) ::= <<
<labels:{it |case <it>:}; separator="\n">
{
  <targetState>
}
break;
>>

// C y c l i c D F A

/** The code to initiate execution of a cyclic DFA; this is used
 * in the rule to predict an alt just like the fixed DFA case.
 * The <name> attribute is inherited via the parser, lexer, ...
 */
dfaDecision(decisionNumber,description) ::= <<
alt<decisionNumber> = dfa<decisionNumber>.predict(input);
>>

/* Dump DFA tables as run-length-encoded Strings of octal values.
 * Can't use hex as compiler translates them before compilation.
 * These strings are split into multiple, concatenated strings.

```

- * Java puts them back together at compile time thankfully.
- * Java cannot handle large static arrays, so we're stuck with this
- * encode/decode approach. See analysis and runtime DFA for
- * the encoding methods.

*/

```
cyclicDFA(dfa) ::= <<
static final String DFA<dfa.decisionNumber>_eotS =
"<dfa.javaCompressedEOT; wrap="\n\t">";
static final String DFA<dfa.decisionNumber>_eofS =
"<dfa.javaCompressedEOF; wrap="\n\t">";
static final String DFA<dfa.decisionNumber>_minS =
"<dfa.javaCompressedMin; wrap="\n\t">";
static final String DFA<dfa.decisionNumber>_maxS =
"<dfa.javaCompressedMax; wrap="\n\t">";
static final String DFA<dfa.decisionNumber>_acceptS =
"<dfa.javaCompressedAccept; wrap="\n\t">";
static final String DFA<dfa.decisionNumber>_specialS =
"<dfa.javaCompressedSpecial; wrap="\n\t">";
static final String[] DFA<dfa.decisionNumber>_transitionS = {
<dfa.javaCompressedTransition:{s|<s; wrap="\n\t">"; separator=",\n">
};

static final short[] DFA<dfa.decisionNumber>_eot =
DFA.unpackEncodedString(DFA<dfa.decisionNumber>_eotS);
static final short[] DFA<dfa.decisionNumber>_eof =
DFA.unpackEncodedString(DFA<dfa.decisionNumber>_eofS);
static final char[] DFA<dfa.decisionNumber>_min =
DFA.unpackEncodedStringToUnsignedChars(DFA<dfa.decisionNumber>_minS);
static final char[] DFA<dfa.decisionNumber>_max =
DFA.unpackEncodedStringToUnsignedChars(DFA<dfa.decisionNumber>_maxS);
static final short[] DFA<dfa.decisionNumber>_accept =
DFA.unpackEncodedString(DFA<dfa.decisionNumber>_acceptS);
static final short[] DFA<dfa.decisionNumber>_special =
DFA.unpackEncodedString(DFA<dfa.decisionNumber>_specialS);
static final short[][] DFA<dfa.decisionNumber>_transition;

static {
int numStates = DFA<dfa.decisionNumber>_transitionS.length;
DFA<dfa.decisionNumber>_transition = new short[numStates][];
for (int i=0; i<numStates; i++) {
DFA<dfa.decisionNumber>_transition[i] = DFA.unpackEncodedString(DFA<dfa.decisionNumber>_transitionS[i]);
}
}

protected class DFA<dfa.decisionNumber> extends DFA {

public DFA<dfa.decisionNumber>(BaseRecognizer recognizer) {
this.recognizer = recognizer;
```

```

this.decisionNumber = <dfa.decisionNumber>;
this.eot = DFA<dfa.decisionNumber>_eot;
this.eof = DFA<dfa.decisionNumber>_eof;
this.min = DFA<dfa.decisionNumber>_min;
this.max = DFA<dfa.decisionNumber>_max;
this.accept = DFA<dfa.decisionNumber>_accept;
this.special = DFA<dfa.decisionNumber>_special;
this.transition = DFA<dfa.decisionNumber>_transition;
}
@Override
public String getDescription() {
    return "<dfa.description>";
}
<@errorMethod()>
<if(dfa.specialStateSTs)>
@Override
public int specialStateTransition(int s, IntStream _input) throws NoViableAltException {
    <if(LEXER)>
    IntStream input = _input;
    <endif>
    <if(PARSER)>
    TokenStream input = (TokenStream)_input;
    <endif>
    <if(TREE_PARSER)>
    TreeNodeStream input = (TreeNodeStream)_input;
    <endif>
    int _s = s;
    switch ( s ) {
    <dfa.specialStateSTs:{ state |
    case <i0> : <! compressed special state numbers 0..n-1 !>
    <state>}; separator="\n">
    }
    <if(backtracking)>
    if (state.backtracking>0) {state.failed=true; return -1;}
    <endif>
    NoViableAltException nvae =
    new NoViableAltException(getDescription(), <dfa.decisionNumber>, _s, input);
    error(nvae);
    throw nvae;
    }
    <endif>
}
>>

/** A state in a cyclic DFA; it's a special state and part of a big switch on
 * state.
 */
cyclicDFAState(decisionNumber,stateNumber,edges,needErrorClause,semPredState) ::= <<

```

```

int LA<decisionNumber>_<stateNumber> = input.LA(1);
<if(semPredState)> <! get next lookahead symbol to test edges, then rewind !>
int index<decisionNumber>_<stateNumber> = input.index();
input.rewind();
<endif>
s = -1;
<edges; separator="\nelse ">
<if(semPredState)> <! return input cursor to state before we rewound !>
input.seek(index<decisionNumber>_<stateNumber>);
<endif>
if ( s>=0 ) return s;
break;
>>

/** Just like a fixed DFA edge, test the lookahead and indicate what
 * state to jump to next if successful.
 */
cyclicDFAEdge(labelExpr, targetStateNumber, edgeNumber, predicates) ::= <<
if ( (<labelExpr> <if(predicates)>&& (<predicates>)<endif> ) { s = <targetStateNumber>;}
>>

/** An edge pointing at end-of-token; essentially matches any char;
 * always jump to the target.
 */
eotDFAEdge(targetStateNumber,edgeNumber, predicates) ::= <<
s = <targetStateNumber>;
>>

// D F A E X P R E S S I O N S

andPredicates(left,right) ::= "<left>&&<right>"

orPredicates(operands) ::= "<operands; separator=\"||\">"

notPredicate(pred) ::= "!(<evalPredicate(pred,{}>)"

evalPredicate(pred,description) ::= "<pred>"

evalSynPredicate(pred,description) ::= "<pred>()"

lookaheadTest(atom,k,atomAsInt) ::= "LA<decisionNumber>_<stateNumber>===<atom>"

/** Sometimes a lookahead test cannot assume that LA(k) is in a temp variable
 * somewhere. Must ask for the lookahead directly.
 */
isolatedLookaheadTest(atom,k,atomAsInt) ::= "input.LA(<k>)==<atom>"

```

```

lookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= <%
(LA<decisionNumber>_<stateNumber> >= <lower> && LA<decisionNumber>_<stateNumber> \<= <upper>)
%>

isolatedLookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= "(input.LA(<k>) >= <lower>
&& input.LA(<k>) \<= <upper>)"

setTest(ranges) ::= <<
<ranges; separator="|">
>>

// A T T R I B U T E S

globalAttributeScope(scope) ::= <<
<if(scope.attributes)>
protected static class <scope.name>_scope {
  <scope.attributes:{it |<it.decl>;}; separator="\n">
}
protected Stack\<<scope.name>_scope> <scope.name>_stack = new Stack\<<scope.name>_scope>();
<endif>
>>

ruleAttributeScope(scope) ::= <<
<if(scope)>
<if(scope.attributes)>
protected static class <scope.name>_scope {
  <scope.attributes:{it |<it.decl>;}; separator="\n">
}
protected Stack\<<scope.name>_scope> <scope.name>_stack = new Stack\<<scope.name>_scope>();
<endif>
<endif>
>>

returnStructName(r) ::= "<r.name>_return"

returnType(ruleDescriptor) ::= <%
<if(ruleDescriptor.hasMultipleReturnValues)>
<ruleDescriptor.grammar.recognizerName>.<ruleDescriptor:returnStructName()>
<elseif(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnType>
<else>
void
<endif>
%>

/** Generate the Java type associated with a single or multiple return
 * values.
 */

```

```

ruleLabelType(referencedRule) ::= <%
<if(referencedRule.hasMultipleReturnValues)>
<returnScopeBaseType()>
<elseif(referencedRule.hasSingleReturnValue)>
<referencedRule.singleValueReturnType>
<else>
void
<endif>
%>

delegateName(d) ::= <<
<if(d.label)><d.label><else>g<d.name><endif>
>>

/** Using a type to init value map, try to init a type; if not in table
 * must be an object, default value is "null".
 */
initValue(typeName) ::= <<
<javaTypeInitMap.(typeName)>
>>

/** Define a rule label including default value */
ruleLabelDef(label) ::= <%
<ruleLabelType(referencedRule=label.referencedRule)> <label.label.text> =
<initValue(typeName=ruleLabelType(referencedRule=label.referencedRule))>;
%>

/** Define a return struct for a rule if the code needs to access its
 * start/stop tokens, tree stuff, attributes, ... Leave a hole for
 * subgroups to stick in members.
 */
returnScope(scope) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
public static class <ruleDescriptor:returnStructName()> extends <returnScopeBaseType()> {
<if(scope)><scope.attributes:{it |public <it.decl>;}; separator="\n"><endif>
<@ruleReturnMembers()>
};
<endif>
>>

returnScopeBaseType() ::= <%
<if(TREE_PARSER)>Tree<else>Parser<endif>RuleReturnScope
%>

parameterScope(scope) ::= <<
<scope.attributes:{it |<it.decl>;}; separator=", ">
>>

```

```

parameterAttributeRef(attr) ::= "<attr.name>"
parameterSetAttributeRef(attr,expr) ::= "<attr.name> =<expr>";

scopeAttributeRef(scope,attr,index,negIndex) ::= <%
<if(negIndex)>
<scope>_stack.elementAt(<scope>_stack.size()-<negIndex>-1).<attr.name>
<else>
<if(index)>
<scope>_stack.elementAt(<index>).<attr.name>
<else>
<scope>_stack.peek().<attr.name>
<endif>
<endif>
%>

scopeSetAttributeRef(scope,attr,expr,index,negIndex) ::= <%
<if(negIndex)>
<scope>_stack.elementAt(<scope>_stack.size()-<negIndex>-1).<attr.name> =<expr>;
<else>
<if(index)>
<scope>_stack.elementAt(<index>).<attr.name> =<expr>;
<else>
<scope>_stack.peek().<attr.name> =<expr>;
<endif>
<endif>
%>

/** $x is either global scope or x is rule with dynamic scope; refers
 * to stack itself not top of stack. This is useful for predicates
 * like { $function.size()>0 && $function::name.equals("foo") }?
 */
isolatedDynamicScopeRef(scope) ::= "<scope>_stack"

/** reference an attribute of rule; might only have single return value */
ruleLabelRef(referencedRule,scope,attr) ::= <%
<if(referencedRule.hasMultipleReturnValues)>
(<scope>!=null?(<return Type(referencedRule)><scope>).<attr.name>:<init Value(attr.type)>)
<else>
<scope>
<endif>
%>

returnAttributeRef(ruleDescriptor,attr) ::= <%
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name>
<else>
<attr.name>
<endif>

```

```

%>

returnSetAttributeRef(ruleDescriptor,attr,expr) ::= <%
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name> =<expr>;
<else>
<attr.name> =<expr>;
<endif>
%>

/** How to translate $tokenLabel */
tokenLabelRef(label) ::= "<label>"

/** ids+=ID {$ids} or e+=expr {$e} */
listLabelRef(label) ::= "list_<label>"

// not sure the next are the right approach

tokenLabelPropertyRef_text(scope,attr) ::= "<scope>!=null?<scope>.getText():null)"
tokenLabelPropertyRef_type(scope,attr) ::= "<scope>!=null?<scope>.getType():0)"
tokenLabelPropertyRef_line(scope,attr) ::= "<scope>!=null?<scope>.getLine():0)"
tokenLabelPropertyRef_pos(scope,attr) ::= "<scope>!=null?<scope>.getCharPositionInLine():0)"
tokenLabelPropertyRef_channel(scope,attr) ::= "<scope>!=null?<scope>.getChannel():0)"
tokenLabelPropertyRef_index(scope,attr) ::= "<scope>!=null?<scope>.getTokenIndex():0)"
tokenLabelPropertyRef_tree(scope,attr) ::= "<scope>_tree"
tokenLabelPropertyRef_int(scope,attr) ::= "<scope>!=null?Integer.valueOf(<scope>.getText()):0)"

ruleLabelPropertyRef_start(scope,attr) ::= "<scope>!=null?(<castToLabelType({<scope>.start})>):null)"
ruleLabelPropertyRef_stop(scope,attr) ::= "<scope>!=null?(<castToLabelType({<scope>.stop})>):null)"
ruleLabelPropertyRef_tree(scope,attr) ::= "<scope>!=null?(<ASTLabelType><scope>.getTree()):null)"
ruleLabelPropertyRef_text(scope,attr) ::= <%
<if(TREE_PARSER)>
<scope>!=null?(input.getTokenStream().toString(
input.getTreeAdaptor().getTokenStartIndex(<scope>.start),
input.getTreeAdaptor().getTokenStopIndex(<scope>.start))):null)
<else>
<scope>!=null?input.toString(<scope>.start,<scope>.stop):null)
<endif>
%>

ruleLabelPropertyRef_st(scope,attr) ::= "<scope>!=null?((StringTemplate)<scope>.getTemplate()):null)"

/** Isolated $RULE ref ok in lexer as it's a Token */
lexerRuleLabel(label) ::= "<label>"

lexerRuleLabelPropertyRef_type(scope,attr) ::=
"<scope>!=null?<scope>.getType():0)"

```

```

lexerRuleLabelPropertyRef_line(scope,attr) ::=
    "<scope>!=null?<scope>.getLine():0)"
lexerRuleLabelPropertyRef_pos(scope,attr) ::=
    "<scope>!=null?<scope>.getCharPositionInLine():-1)"
lexerRuleLabelPropertyRef_channel(scope,attr) ::=
    "<scope>!=null?<scope>.getChannel():0)"
lexerRuleLabelPropertyRef_index(scope,attr) ::=
    "<scope>!=null?<scope>.getTokenIndex():0)"
lexerRuleLabelPropertyRef_text(scope,attr) ::=
    "<scope>!=null?<scope>.getText():null)"
lexerRuleLabelPropertyRef_int(scope,attr) ::=
    "<scope>!=null?Integer.valueOf(<scope>.getText()):0)"

// Somebody may ref $template or $tree or $stop within a rule:
rulePropertyRef_start(scope,attr) ::= "<castToLabelType(\"retval.start\")>"
rulePropertyRef_stop(scope,attr) ::= "<castToLabelType(\"retval.stop\")>"
rulePropertyRef_tree(scope,attr) ::= "retval.tree"
rulePropertyRef_text(scope,attr) ::= <%
<if(TREE_PARSER)>
input.getTokenStream().toString(
    input.getTreeAdaptor().getTokenStartIndex(retval.start),
    input.getTreeAdaptor().getTokenStopIndex(retval.start))
<else>
input.toString(retval.start,input.LT(-1))
<endif>
%>
rulePropertyRef_st(scope,attr) ::= "retval.st"

lexerRulePropertyRef_text(scope,attr) ::= "getText()"
lexerRulePropertyRef_type(scope,attr) ::= "_type"
lexerRulePropertyRef_line(scope,attr) ::= "state.tokenStartLine"
lexerRulePropertyRef_pos(scope,attr) ::= "state.tokenStartCharPositionInLine"
lexerRulePropertyRef_index(scope,attr) ::= "-1" // undefined token index in lexer
lexerRulePropertyRef_channel(scope,attr) ::= "_channel"
lexerRulePropertyRef_start(scope,attr) ::= "state.tokenStartCharIndex"
lexerRulePropertyRef_stop(scope,attr) ::= "(getCharIndex()-1)"
lexerRulePropertyRef_int(scope,attr) ::= "Integer.valueOf(<scope>.getText())"

// setting $st and $tree is allowed in local rule. everything else
// is flagged as error
ruleSetPropertyRef_tree(scope,attr,expr) ::= "retval.tree =<expr>";
ruleSetPropertyRef_st(scope,attr,expr) ::= "retval.st =<expr>";

/** How to execute an action (only when not backtracking) */
execAction(action) ::= <%
<if(backtracking)>
if ( <actions.(actionScope).synpredgate> ) {
    <action>

```

```

}
<else>
<action>
<endif>
%>

/** How to always execute an action even when backtracking */
execForcedAction(action) ::= "<action>"

// M I S C (properties, etc...)

bitset(name, words64) ::= <<
public static final BitSet <name> = new BitSet(new long[] { <words64: { it |<it>L }; separator=", "> });
>>

codeFileExtension() ::= ".java"

true_value() ::= "true"
false_value() ::= "false"

isEmptyString ::= [
    "" : true,
    default : false
]

isTrue ::= [
    "true" : true,
    default : false
]

isDefaultLabelType ::= [
    "Token" : true,
    default : false
]

isPredefinedRewriteRule ::= [
    "topdown" : true,
    "bottomup" : true,
    default : false
]

castToLabelType(value) ::= <%
<if(!isDefaultLabelType.(labelType))>
(<labelType>)
<endif>
<value>
%>

```

Found in path(s):

* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/Java/Java.stg

No license file was found, but licenses were detected in source scan.

/*

* [The "BSD license"]

* Copyright (c) 2011 Terence Parr

* All rights reserved.

*

* Conversion to C#:

* Copyright (c) 2011 Sam Harwell, Pixel Mine, Inc.

* All rights reserved.

*

* Redistribution and use in source and binary forms, with or without

* modification, are permitted provided that the following conditions

* are met:

* 1. Redistributions of source code must retain the above copyright

* notice, this list of conditions and the following disclaimer.

* 2. Redistributions in binary form must reproduce the above copyright

* notice, this list of conditions and the following disclaimer in the

* documentation and/or other materials provided with the distribution.

* 3. The name of the author may not be used to endorse or promote products

* derived from this software without specific prior written permission.

*

* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR

* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES

* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.

* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,

* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT

* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,

* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY

* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT

* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF

* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

```
csharpVisibilityMap ::= [
```

```
  "private": "private",
```

```
  "protected": "protected",
```

```
  "public": "public",
```

```
  "fragment": "private",
```

```
  default: "private"
```

```
]
```

```
/** The overall file structure of a recognizer; stores methods for rules
```

```
* and cyclic DFAs plus support code.
```

```
*/
```

```
outputFile( LEXER,PARSER,TREE_PARSER, actionScope, actions,
```

```

docComment, recognizer,
name, tokens, tokenNames, rules, cyclicDFAs,
bitsets, buildTemplate, buildAST, rewriteMode, profile,
backtracking, synpreds, memoize, numRules,
fileName, ANTLRVersion, generatedTimestamp, trace,
scopes, superClass, literals) ::=
<<
//-----
//\<auto-generated>
// This code was generated by a tool.
// ANTLR Version: <ANTLRVersion>
//
// Changes to this file may cause incorrect behavior and will be lost if
// the code is regenerated.
//\</auto-generated>
//-----

// $ANTLR <ANTLRVersion> <fileName> <generatedTimestamp>

// The variable 'variable' is assigned but its value is never used.
#pragma warning disable 219
// Unreachable code detected.
#pragma warning disable 162
// Missing XML comment for publicly visible type or member 'Type_or_Member'
#pragma warning disable 1591
// CLS compliance checking will not be performed on 'type' because it is not visible from outside this assembly.
#pragma warning disable 3019

<actions.(actionScope).header>

<@imports>
using System.Collections.Generic;
using Antlr.Runtime;
using Antlr.Runtime.Misc;
<if(TREE_PARSER)>
using Antlr.Runtime.Tree;
using RewriteRuleITokenStream = Antlr.Runtime.Tree.RewriteRuleTokenStream;
<endif>
<@end>
<if(actions.(actionScope).namespace)>
namespace <actions.(actionScope).namespace>
{
<endif>
<docComment>
<recognizer>
<if(actions.(actionScope).namespace)>

} // namespace <actions.(actionScope).namespace>

```

```

<endif>
>>

lexerInputStreamType() ::= <<
<actions.(actionScope).inputStreamType; null="ICharStream">
>>

lexer(grammar, name, tokens, scopes, rules, numRules, filterMode, labelType="CommonToken",
superClass={ <if(actions.(actionScope).superClass)><actions.(actionScope).superClass><else>Antlr.Runtime.Lexer
<endif>},
rewriteElementType={}, ASTLabelType={}) ::= <<
[System.CodeDom.Compiler.GeneratedCode("ANTLR", "<ANTLRVersion>")]
[System.CLSCompliant(false)]
<parserModifier(grammar=grammar, actions=actions)> partial class <grammar.recognizerName> :
<@superClassName><superClass><@end>
{
<tokens:{ it|public const int <it.name; format="id">=<it.type>;}; separator="\n">
<scopes:{ it|<if(it.isDynamicGlobalScope)><globalAttributeScope(scope=it)><endif>}>
<actions.lexer.members>

// delegates
<grammar.delegates:
{ g|private <g.recognizerName> <g.delegateName(>);}; separator="\n">
// delegators
<grammar.delegators:
{ g|private <g.recognizerName> <g.delegateName(>);}; separator="\n">
<last(grammar.delegators):{ g|private <g.recognizerName> gParent;}>

<actions.(actionScope).ctorModifier; null="public"> <grammar.recognizerName>(<#! needed by subclasses !>
{
OnCreated();
}

<actions.(actionScope).ctorModifier; null="public"> <grammar.recognizerName>(<lexerInputStreamType(>
input<grammar.delegators:{ g|, <g.recognizerName> <g.delegateName(>}&> )
: this(input, new RecognizerSharedState(<grammar.delegators:{ g|, <g.delegateName(>}&>))
{
}

<actions.(actionScope).ctorModifier; null="public"> <grammar.recognizerName>(<lexerInputStreamType(> input,
RecognizerSharedState state<grammar.delegators:{ g|, <g.recognizerName> <g.delegateName(>}&>))
: base(input, state)
{
<if(memoize)>
<if(grammar.grammarIsRoot)>
state.ruleMemo = new System.Collections.Generic.Dictionary<int, int>[<numRules>+1];<\n><#! index from 1..n !>
<endif>
<endif>
}
}

```

```

    <grammar.directDelegates:
    { g|<g:delegateName()> = new <g.recognizerName>(input, this.state<trunc(g.delegates):{p|,
    <p:delegateName()>}>, this);}; separator="\n">
    <grammar.delegates:
    { g|this.<g:delegateName()> = <g:delegateName()>;}; separator="\n">
    <last(grammar.delegates):{ g|gParent = <g:delegateName()>;}>

    OnCreated();
    }
    public override string GrammarFileName { get { return "<fileName>"; } }

    <if(grammar.hasDelegates)>
    public override <lexerInputStreamType()> CharStream
    {
    get
    {
    return base.CharStream;
    }
    set
    {
    base.CharStream = value;
    <grammar.directDelegates:
    { g|<g:delegateName()> = new <g.recognizerName>(input, state<trunc(g.delegates):{p|, <p:delegateName()>}>,
    this);}; separator="\n">
    <grammar.delegates:
    { g|this.<g:delegateName()> = <g:delegateName()>;}; separator="\n">
    <last(grammar.delegates):{ g|gParent = <g:delegateName()>;}>
    }
    }

    <if(grammar.delegates)>
    public override void SetState(RecognizerSharedState state)
    {
    base.SetState(state);
    <grammar.delegates:{ g|<g:delegateName()>.SetState(state);}; separator="\n">
    }
    <endif>

    <endif>
    <if(filterMode)>
    <filteringNextToken()>
    <endif>

    partial void OnCreated();
    partial void EnterRule(string ruleName, int ruleIndex);
    partial void LeaveRule(string ruleName, int ruleIndex);

```

```

<rules; separator="\n">

<insertLexerSynpreds(synpreds)>

#region DFA
<cyclicDFAs:{ dfa | DFA<dfa.decisionNumber> dfa<dfa.decisionNumber>;}; separator="\n">

protected override void InitDFAs()
{
    base.InitDFAs();
    <cyclicDFAs:{ dfa | dfa<dfa.decisionNumber> = new DFA<dfa.decisionNumber>(this<if(dfa.specialStateSTs)>,
SpecialStateTransition<dfa.decisionNumber><endif>;); separator="\n">
}

<cyclicDFAs:cyclicDFA()><! dump tables for all DFA !>
#endregion

}
>>

/** A override of Lexer.nextToken() that backtracks over mTokens() looking
 * for matches. No error can be generated upon error; just rewind, consume
 * a token and then try again. backtracking needs to be set as well.
 * Make rule memoization happen only at levels above 1 as we start mTokens
 * at backtracking==1.
 */
filteringNextToken() ::= <<
public override IToken NextToken()
{
    while (true)
    {
        if (input.LA(1) == CharStreamConstants.EndOfFile)
        {
            IToken eof = new CommonToken((ICharStream)input, CharStreamConstants.EndOfFile, TokenChannels.Default,
input.Index, input.Index);
            eof.Line = Line;
            eof.CharPositionInLine = CharPositionInLine;
            return eof;
        }
        state.token = null;
        state.channel = TokenChannels.Default;
        state.tokenStartCharIndex = input.Index;
        state.tokenStartCharPositionInLine = input.CharPositionInLine;
        state.tokenStartLine = input.Line;
        state.text = null;
        try
        {
            int m = input.Mark();

```

```

state.backtracking=1;<! means we won't throw slow exception !>
state.failed=false;
mTokens();
state.backtracking=0;
<! mTokens backtracks with synpred at backtracking==2
and we set the synpredgate to allow actions at level 1. !>
if (state.failed)
{
input.Rewind(m);
input.Consume();<! advance one char and try again !>
}
else
{
Emit();
return state.token;
}
}
catch (RecognitionException re)
{
// shouldn't happen in backtracking mode, but...
ReportError(re);
Recover(re);
}
}
}

public override void Memoize(IIntStream input, int ruleIndex, int ruleStartIndex)
{
if (state.backtracking > 1)
base.Memoize(input, ruleIndex, ruleStartIndex);
}

public override bool AlreadyParsedRule(IIntStream input, int ruleIndex)
{
if (state.backtracking > 1)
return base.AlreadyParsedRule(input, ruleIndex);

return false;
}
>>

actionGate() ::= "state.backtracking == 0"

filteringActionGate() ::= "state.backtracking == 1"

/** How to generate a parser */
genericParser(grammar, name, scopes, tokens, tokenNames, rules, numRules,
bitsets, inputStreamType, superClass,

```

```

        labelType, members, rewriteElementType,
        filterMode, ASTLabelType="object") ::= <<
[System.CodeDom.Compiler.GeneratedCode("ANTLR", "<ANTLRVersion>")]
[System.CLSCompliant(false)]
<parserModifier(grammar=grammar, actions=actions)> partial class <grammar.recognizerName> :
<@superClassName><superClass><@end>
{
<if(grammar.grammarIsRoot)>
internal static readonly string[] tokenNames = new string[] {
    "\<invalid>", "\<EOR>", "\<DOWN>", "\<UP>", <tokenNames; separator=", ">
};
<endif>
<tokens:{it|public const int <it.name; format="id">=<it.type>;}; separator="\n">

<if(grammar.delegates)>
// delegates
<grammar.delegates:
    {g|private <g.recognizerName> <g.delegateName(>);}; separator="\n">
<endif>
<if(grammar.delegators)>
// delegators
<grammar.delegators:
    {g|private <g.recognizerName> <g.delegateName(>);}; separator="\n">
<last(grammar.delegators):{g|private <g.recognizerName> gParent;}>
<endif>

<if(grammar.delegates)>
public override void SetState(RecognizerSharedState state)
{
    base.SetState(state);
    <grammar.delegates:{g|<g.delegateName(>.SetState(state);}; separator="\n">
}

<if(TREE_PARSER)>
public override void SetTreeNodeStream(ITreeNodeStream input)
{
    base.SetTreeNodeStream(input);
    <grammar.delegates:{g|<g.delegateName(>.SetTreeNodeStream(input);}; separator="\n">
}
<endif>
<endif>

<scopes:{it|<if(it.isDynamicGlobalScope)><globalAttributeScope(scope=it)><endif>}>
<@members(>

public override string[] TokenNames { get { return
<grammar.composite.rootGrammar.recognizerName>.tokenNames; } }
public override string GrammarFileName { get { return "<fileName>"; } }

```

```

<members>

partial void OnCreated();
partial void EnterRule(string ruleName, int ruleIndex);
partial void LeaveRule(string ruleName, int ruleIndex);

#region Rules
<rules; separator="\n">
#endregion Rules

<if(grammar.delegatedRules)>
<! generate rule/method definitions for imported rules so they
appear to be defined in this recognizer. !>
#region Delegated rules
<grammar.delegatedRules:{ ruleDescriptor|
<ruleModifier(grammar=grammar,ruleDescriptor=ruleDescriptor)> <return Type(ruleDescriptor)>
<ruleDescriptor.name; format="id"><ruleDescriptor.parameterScope:parameterScope()><!throws
RecognitionException !>{ <if(ruleDescriptor.hasReturnValue)>return
<endif><ruleDescriptor.grammar:delegateName()>.<ruleDescriptor.name;
format="id"><ruleDescriptor.parameterScope><ruleDescriptor.parameterScope.attributes:{ a|<a.name;
format="id">}; separator=", "><endif>}; \} }; separator="\n">
#endregion Delegated rules
<endif>

<insertSynpreds(synpreds)>

<if(cyclicDFAs)>
#region DFA
<cyclicDFAs:{ dfa | private DFA<dfa.decisionNumber> dfa<dfa.decisionNumber>; }; separator="\n">

protected override void InitDFAs()
{
base.InitDFAs();
<cyclicDFAs:{ dfa | dfa<dfa.decisionNumber> = new DFA<dfa.decisionNumber>( this<if(dfa.specialStateSTs)>,
SpecialStateTransition<dfa.decisionNumber><endif> ); }; separator="\n">
}

<cyclicDFAs:cyclicDFA()><! dump tables for all DFA !>
#endregion DFA
<endif>

<if(bitsets)>
#region Follow sets
private static class Follow
{
<bitsets: { it|<bitset(name={ _<it.name>_in_<it.inName><it.tokenIndex> }, words64=it.bits)> }; separator="\n">
}

```

```

#endregion Follow sets
<endif>
}
>>

@genericParser.members() ::= <<
<! WARNING. bug in ST: this is cut-n-paste into Dbg.stg !>
<actions.(actionScope).ctorModifier; null="public"> <grammar.recognizerName><(inputStreamType>
input<grammar.delegates:{g|, <g.recognizerName> <g.delegateName()>}>)
: this(input, new RecognizerSharedState()<grammar.delegates:{g|, <g.delegateName()>}>)
{
}
<actions.(actionScope).ctorModifier; null="public"> <grammar.recognizerName><(inputStreamType> input,
RecognizerSharedState state<grammar.delegates:{g|, <g.recognizerName> <g.delegateName()>}>)
: base(input, state)
{
<if(grammar.directDelegates)>
<grammar.directDelegates:
{g|<g.delegateName()> = new <g.recognizerName>(input, state<trunc(g.delegates):{p|, <p.delegateName()>}>,
this);}; separator="\n">
<endif>
<if(grammar.indirectDelegates)>
<grammar.indirectDelegates:{g | <g.delegateName()> = <g.delegator.delegateName()>.<g.delegateName()>};
separator="\n">
<endif>
<if(grammar.delegates)>
<last(grammar.delegates):{g|gParent = <g.delegateName()>};>
<endif>
<parserCtorBody()>
OnCreated();
}
>>

// imported grammars are 'public' (can't be internal because their return scope classes must be accessible)
parserModifier(grammar, actions) ::= <<
<if(grammar.grammarIsRoot)><actions.(actionScope).modifier; null="public"><else>public<endif>
>>

parserCtorBody() ::= <<
<if(memoize)>
<if(grammar.grammarIsRoot)>
this.state.ruleMemo = new System.Collections.Generic.Dictionary<int,
int>[<length(grammar.allImportedRules)>+1];<\n><! index from 1..n !>
<endif>
<endif>
<grammar.delegates:
{g|this.<g.delegateName()> = <g.delegateName()>}; separator="\n">
>>

```

```

parser(grammar, name, scopes, tokens, tokenNames, rules, numRules, bitsets,
    ASTLabelType="object",
    superClass={ <if(actions.(actionScope).superClass)><actions.(actionScope).superClass><else>Antlr.Runtime.Parser
<endif>}, labelType="IToken",
    members={ <actions.parser.members>}) ::= <<
<genericParser(inputStreamType="ITokenStream", rewriteElementType="IToken", filterMode=false, ...)>
>>

```

```

/** How to generate a tree parser; same as parser except the input

```

```

* stream is a different type.

```

```

*/

```

```

treeParser(grammar, name, scopes, tokens, tokenNames, globalAction, rules,
    numRules, bitsets, filterMode, labelType={ <ASTLabelType>}, ASTLabelType="object",
    superClass={ <if(actions.(actionScope).superClass)><actions.(actionScope).superClass><else>Antlr.Runtime.Tree.<
if(filterMode)><if(buildAST)>TreeRewriter<else>TreeFilter<endif><else>TreeParser<endif><endif>},
    members={ <actions.treeparser.members>}) ::= <<
<genericParser(inputStreamType="ITreeNodeStream", rewriteElementType="Node", ...)>
>>

```

```

/** A simpler version of a rule template that is specific to the imaginary

```

```

* rules created for syntactic predicates. As they never have return values

```

```

* nor parameters etc..., just give simplest possible method. Don't do

```

```

* any of the normal memoization stuff in here either; it's a waste.

```

```

* As predicates cannot be inlined into the invoking rule, they need to

```

```

* be in a rule by themselves.

```

```

*/

```

```

synpredRule(ruleName, ruleDescriptor, block, description, nakedBlock) ::=

```

```

<<

```

```

partial void EnterRule_<ruleName>_fragment();

```

```

partial void LeaveRule_<ruleName>_fragment();

```

```

// $ANTLR start <ruleName>

```

```

<ruleModifier(grammar=grammar,ruleDescriptor=ruleDescriptor)> void

```

```

<ruleName>_fragment(<ruleDescriptor.parameterScope:parameterScope()>)

```

```

{

```

```

    <ruleLabelDefs(...)>

```

```

    EnterRule_<ruleName>_fragment();

```

```

    EnterRule("<ruleName>_fragment", <ruleDescriptor.index>);

```

```

    TraceIn("<ruleName>_fragment", <ruleDescriptor.index>);

```

```

    try

```

```

    {

```

```

        <block>

```

```

    }

```

```

    finally

```

```

    {

```

```

        TraceOut("<ruleName>_fragment", <ruleDescriptor.index>);

```

```

    LeaveRule("<ruleName>_fragment", <ruleDescriptor.index>);
    LeaveRule_<ruleName>_fragment();
}
}
// $ANTLR end <ruleName>
>>

insertLexerSynpreds(synpreds) ::= <<
<insertSynpreds(synpreds)>
>>

insertSynpreds(synpreds) ::= <<
<if(synpreds)>
#region Synpreds
private bool EvaluatePredicate(System.Action fragment)
{
    bool success = false;
    state.backtracking++;
    <@start()>
    try { DebugBeginBacktrack(state.backtracking);
    int start = input.Mark();
    try
    {
        fragment();
    }
    catch ( RecognitionException re )
    {
        System.Console.Error.WriteLine("impossible: "+re);
    }
    success = !state.failed;
    input.Rewind(start);
} finally { DebugEndBacktrack(state.backtracking, success); }
    <@stop()>
    state.backtracking--;
    state.failed=false;
    return success;
}
#endregion Synpreds
<endif>
>>

ruleMemoization(name) ::= <<
<if(memoize)>
if (state.backtracking > 0 && AlreadyParsedRule(input, <ruleDescriptor.index>)) { <returnFromRule()> }
<endif>
>>

/** How to test for failure and return from rule */

```

```

checkRuleBacktrackFailure() ::= <<
<if(backtracking)>if (state.failed) <returnFromRule(><endif>
>>

/** This rule has failed, exit indicating failure during backtrack */
ruleBacktrackFailure() ::= <<
<if(backtracking)>if (state.backtracking>0) {state.failed=true; <returnFromRule(>}<endif>
>>

ruleWrapperMap ::= [
"bottomup":{<ruleWrapperBottomup(>},
"topdown":{<ruleWrapperTopdown(>},
default:""
]

ruleWrapperBottomup() ::= <<
<if(TREE_PARSER && filterMode)>
protected override <if(buildAST)>IAstRuleReturnScope<else>void<endif> Bottomup() { <if(buildAST)>return
<endif>bottomup(); }
<endif>
>>

ruleWrapperTopdown() ::= <<
<if(TREE_PARSER && filterMode)>
protected override <if(buildAST)>IAstRuleReturnScope<else>void<endif> Topdown() { <if(buildAST)>return
<endif>topdown(); }
<endif>
>>

/** How to generate code for a rule. This includes any return type
 * data aggregates required for multiple return values.
 */
rule(ruleName,ruleDescriptor,block,emptyRule,description,exceptions,finally,memoize) ::= <<
<ruleAttributeScope(scope=ruleDescriptor.ruleScope)>
<returnScope(ruleDescriptor.returnScope)>
partial void EnterRule_<ruleName>();
partial void LeaveRule_<ruleName>();
<ruleWrapperMap.(ruleName)>
// $ANTLR start "<ruleName>"
// <fileName>:<description>
[GrammarRule("<ruleName>")]
<ruleModifier(grammar=grammar,ruleDescriptor=ruleDescriptor)> <returnType(ruleDescriptor)> <ruleName;
format="id">(<ruleDescriptor.parameterScope:parameterScope(>)
{
EnterRule_<ruleName>();
EnterRule("<ruleName>", <ruleDescriptor.index>);
TraceIn("<ruleName>", <ruleDescriptor.index>);
<ruleScopeSetUp(>

```

```

<ruleDeclarations(>
<ruleLabelDefs(...)>
<ruleDescriptor.actions.init>
try { DebugEnterRule(GrammarFileName, "<ruleName>");
DebugLocation(<ruleDescriptor.tree.line>, <ruleDescriptor.EORNode.charPositionInLine>);
<@preamble(>
try
{
<ruleMemoization(name=ruleName)>
<block>
<ruleCleanUp(>
<(ruleDescriptor.actions.after):execAction(>
}
<if(exceptions)>
<exceptions: {e|<catch(decl=e.decl,action=e.action)><\n>}>
<else>
<if(!emptyRule)>
<if(actions.(actionScope).rulecatch)>
<actions.(actionScope).rulecatch>
<else>
catch (RecognitionException re)
{
ReportError(re);
Recover(input,re);
<@setErrorReturnValue(>
}
<endif>
<endif>
<endif>
finally
{
TraceOut("<ruleName>", <ruleDescriptor.index>);
LeaveRule("<ruleName>", <ruleDescriptor.index>);
LeaveRule_<ruleName>());
<memoize(>
<ruleScopeCleanUp(>
<finally>
}
DebugLocation(<ruleDescriptor.EORNode.line>, <ruleDescriptor.EORNode.charPositionInLine>);
} finally { DebugExitRule(GrammarFileName, "<ruleName>"); }
<@postamble(>
<returnFromRule(><\n>
}
// $ANTLR end "<ruleName>"
>>

// imported grammars need to have internal rules
ruleModifier(grammar,ruleDescriptor) ::= <<

```

```

<if(grammar.grammarIsRoot)><csharpVisibilityMap.(ruleDescriptor.modifier);
null="private"><else>internal<endif>
>>

// imported grammars need to have public return scopes
returnScopeModifier(grammar,ruleDescriptor) ::= <<
<if(grammar.grammarIsRoot)><csharpVisibilityMap.(ruleDescriptor.modifier);
null="private"><else>public<endif>
>>

catch(decl,action) ::= <<
catch (<e.decl>)
{
<e.action>
}
>>

ruleDeclarations() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<returnType(ruleDescriptor)> retval = new
<returnType(ruleDescriptor)><(if(ruleDescriptor.returnScope.attributes)>this<endif>);
retval.Start = (<labelType>)input.LT(1);
<elseif(ruleDescriptor.returnScope)>
<ruleDescriptor.returnScope.attributes: { a |
<a.type> <a.name; format="id"> = <if(a.initValue)><a.initValue><else><initValue(a.type)><endif>;
}>
<endif>
<if(memoize)>
int <ruleDescriptor.name>_startIndex = input.Index;
<endif>
>>

ruleScopeSetUp() ::= <<
<ruleDescriptor.useScopes: { it|<it>_stack.Push(new <it>_scope(this));<it>_scopeInit(<it>_stack.Peek());};
separator="\n">
<ruleDescriptor.ruleScope: { it|<it.name>_stack.Push(new
<it.name>_scope(this));<it.name>_scopeInit(<it.name>_stack.Peek());}; separator="\n">
>>

ruleScopeCleanUp() ::= <<
<ruleDescriptor.useScopes: { it|<it>_scopeAfter(<it>_stack.Peek());<it>_stack.Pop();}; separator="\n">
<ruleDescriptor.ruleScope: { it|<it.name>_scopeAfter(<it.name>_stack.Peek());<it.name>_stack.Pop();};
separator="\n">
>>

ruleLabelDefs(ruleDescriptor, labelType, ASTLabelType, rewriteElementType) ::= <<
<[ruleDescriptor.tokenLabels,ruleDescriptor.tokenListLabels,ruleDescriptor.wildcardTreeLabels,ruleDescriptor.wildcardTreeListLabels]

```

```

: {it|<labelType> <it.label.text> = default(<labelType>);}; separator="\n"
>
<ruleDescriptor.tokenListLabels
: {it|List\<<labelType>> list_<it.label.text> = null;}; separator="\n"
>
<[ruleDescriptor.ruleListLabels,ruleDescriptor.wildcardTreeListLabels]
: {it|List\<<ASTLabelType>> list_<it.label.text> = null;}; separator="\n"
>
<ruleDescriptor.ruleLabels:ruleLabelDef(); separator="\n">
<ruleDescriptor.ruleListLabels:ruleLabelDef(); separator="\n">
>>

lexerRuleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,
ruleDescriptor.tokenListLabels,
ruleDescriptor.ruleLabels]
: {it|<labelType> <it.label.text> = default(<labelType>);}; separator="\n"
>
<[ruleDescriptor.charListLabels,
ruleDescriptor.charLabels]
: {it|int <it.label.text> = 0;}; separator="\n"
>
<[ruleDescriptor.tokenListLabels,
ruleDescriptor.ruleListLabels]
: {it|List\<<labelType>> list_<it.label.text> = null;}; separator="\n"
>
<ruleDescriptor.charListLabels: {it|List\<int> list_<it.label.text> = null;}; separator="\n"
>
>>

returnFromRule() ::= <%
return
<if(!ruleDescriptor.isSynPred)>
<if(ruleDescriptor.hasReturnValue)>
<if(ruleDescriptor.hasSingleReturnValue)>
<! This comment is a hack to make sure the following
single space appears in the output. !> <ruleDescriptor.singleValueReturnName>
<else>
<!!> retval
<endif>
<endif>
<endif>
;
%>

ruleCleanUp() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(!TREE_PARSER)>

```

```

retval.Stop = (<labelType>)input.LT(-1);
<endif>
<endif>
>>

memoize() ::= <<
<if(memoize)>
<if(backtracking)>
if (state.backtracking > 0) { Memoize(input, <ruleDescriptor.index>, <ruleDescriptor.name>_StartIndex); }
<endif>
<endif>
>>

/** How to generate a rule in the lexer; naked blocks are used for
 * fragment rules.
 */
lexerRule(ruleName,nakedBlock,ruleDescriptor,block,memoize) ::= <<

partial void EnterRule_<ruleName>();
partial void LeaveRule_<ruleName>();

// $ANTLR start "<ruleName>"
[GrammarRule("<ruleName>")]
<ruleModifier(grammar=grammar,ruleDescriptor=ruleDescriptor)> void
m<ruleName>(<ruleDescriptor.parameterScope:parameterScope(>>)
{
  EnterRule_<ruleName>();
  EnterRule("<ruleName>", <ruleDescriptor.index>);
  TraceIn("<ruleName>", <ruleDescriptor.index>);
  <ruleScopeSetUp(>
  <ruleDeclarations(>
  try
  {
    <if(nakedBlock)>
    <ruleMemoization(name=ruleName)>
    <lexerRuleLabelDefs(>
    <ruleDescriptor.actions.init>
    <block>
    <else>
    int _type = <ruleName>;
    int _channel = DefaultTokenChannel;
    <ruleMemoization(name=ruleName)>
    <lexerRuleLabelDefs(>
    <ruleDescriptor.actions.init>
    <block>
    <ruleCleanUp(>
    state.type = _type;
    state.channel = _channel;

```

```

    <(ruleDescriptor.actions.after):execAction()>
<endif>
}
finally
{
    TraceOut("<ruleName>", <ruleDescriptor.index>);
    LeaveRule("<ruleName>", <ruleDescriptor.index>);
    LeaveRule_<ruleName>();
    <ruleScopeCleanUp()>
    <memoize()>
}
}
// $ANTLR end "<ruleName>"
>>

/** How to generate code for the implicitly-defined lexer grammar rule
 * that chooses between lexer rules.
 */
tokensRule(ruleName,nakedBlock,args,block,ruleDescriptor) ::= <<

public override void mTokens()
{
    <block><\n>
}
>>

// S U B R U L E S

/** A (...) subrule with multiple alternatives */
block(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
// <fileName>:<description>
int alt<decisionNumber>=<maxAlt>;
<decls>
<@predecision()>
try { DebugEnterSubRule(<decisionNumber>);
try { DebugEnterDecision(<decisionNumber>, false<!<decision.dfa.hasSynPred>!>);
<decision>
} finally { DebugExitDecision(<decisionNumber>); }
<@postdecision()>
<@prebranch()>
switch (alt<decisionNumber>)
{
    <alts:{a|<altSwitchCase(i,a)>}>
}
} finally { DebugExitSubRule(<decisionNumber>); }
<@postbranch()>
>>

```

```

/** A rule block with multiple alternatives */
ruleBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
// <fileName>:<description>
int alt<decisionNumber>=<maxAlt>;
<decls>
<@predecision()>
try { DebugEnterDecision(<decisionNumber>, false<!<decision.dfa.hasSynPred>!>);
<decision>
} finally { DebugExitDecision(<decisionNumber>); }
<@postdecision()>
switch (alt<decisionNumber>)
{
<alts:{a|altSwitchCase(i,a)}>
}
>>

ruleBlockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
// <fileName>:<description>
<decls>
<@prealt()>
DebugEnterAlt(1);
<alts>
<@postalt()>
>>

/** A special case of a (...) subrule with a single alternative */
blockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
// <fileName>:<description>
<decls>
<@prealt()>
DebugEnterAlt(1);
<alts>
<@postalt()>
>>

/** A (..)+ block with 1 or more alternatives */
positiveClosureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
// <fileName>:<description>
int cnt<decisionNumber>=0;
<decls>
<@preloop()>
try { DebugEnterSubRule(<decisionNumber>);
while (true)
{
int alt<decisionNumber>=<maxAlt>;
<@predecision()>
try { DebugEnterDecision(<decisionNumber>, false<!<decision.dfa.hasSynPred>!>);

```



```

}

loop<decisionNumber>:
;

} finally { DebugExitSubRule(<decisionNumber>); }
<@postloop()>
>>

closureBlockSingleAlt ::= closureBlock

/** Optional blocks (x)? are translated to (x|) by before code generation
 * so we can just use the normal block template
 */
optionalBlock ::= block

optionalBlockSingleAlt ::= block

/** A case in a switch that jumps to an alternative given the alternative
 * number. A DFA predicts the alternative and then a simple switch
 * does the jump to the code that actually matches that alternative.
 */
altSwitchCase(altNum,alt) ::= <<
case <altNum>:
<@prealt()>
DebugEnterAlt(<altNum>);
<alt>
break;<\n>
>>

/** An alternative is just a list of elements; at outermost level */
alt(elements,altNum,description,autoAST,outerAlt,treeLevel,rew) ::= <<
// <fileName>:<description>
{
<@declarations()>
<elements:element()>
<rew>
<@cleanup()>
}
>>

/** What to emit when there is no rewrite. For auto build
 * mode, does nothing.
 */
noRewrite(rewriteBlockLevel, treeLevel) ::= ""

// E L E M E N T S

```

```

/** Dump the elements one per line */
element(it) ::= <%
<@prematch(>
DebugLocation(<it.line>, <it.pos>);<\n>
<it.el><\n>
%>

/** match a token optionally with a label in front */
tokenRef(token,label,elementIndex,terminalOptions={ }) ::= <<
<if(label)><label>=(<labelType>)<endif>Match(input,<token>,Follow._<token>_in_<ruleName><elementIndex>);
<checkRuleBacktrackFailure(>
>>

/** ids+=ID */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions={ }) ::= <<
<tokenRef(...)>
<listLabelElem(elem=label,elemType=labelType,...)>
>>

listLabel(label,elem) ::= <<
#error The listLabel template should not be used with this target.<\n>
>>

listLabelElem(label,elem,elemType) ::= <<
if (list_<label>==null) list_<label>=new List<<elemType; null={ <labelType> }>>();
list_<label>.Add(<elem>);<\n>
>>

/** match a character */
charRef(char,label) ::= <<
<if(label)>
<label> = input.LA(1);<\n>
<endif>
Match(<char>); <checkRuleBacktrackFailure(>
>>

/** match a character range */
charRangeRef(a,b,label) ::= <<
<if(label)>
<label> = input.LA(1);<\n>
<endif>
MatchRange(<a>,<b>); <checkRuleBacktrackFailure(>
>>

/** For now, sets are interval tests and must be tested inline */
matchSet(s,label,elementIndex,postmatchCode="",terminalOptions={ }) ::= <<
<if(label)>
<matchSetLabel(>

```

```

<endif>
if (<s>)
{
input.Consume();
<postmatchCode>
<if(!LEXER)>state.errorRecovery=false;<endif><if(backtracking)>state.failed=false;<endif>
}
else
{
<ruleBacktrackFailure()>
MismatchedSetException mse = new MismatchedSetException(null,input);
DebugRecognitionException(mse);
<@mismatchedSetException()>
<if(LEXER)>
Recover(mse);
throw mse;
<else>
throw mse;
<! use following code to make it recover inline; remove throw mse;
recoverFromMismatchedSet(input,mse,Follow._set_in_<ruleName><elementIndex>);
!>
<endif>
}<\n>
>>

```

```

matchSetUnchecked(s,label,elementIndex,postmatchCode=false) ::= <%
<if(label)>
<matchSetLabel()><\n>
<endif>
input.Consume();<\n>
<if(postmatchCode)>
<postmatchCode><\n>
<endif>
<if(!LEXER)>state.errorRecovery=false;<endif><if(backtracking)>state.failed=false;<endif>
%>

```

```

matchSetLabel() ::= <%
<if(LEXER)>
<label>= input.LA(1);
<else>
<label>=(<labelType>)input.LT(1);
<endif>
%>

```

```

matchRuleBlockSet ::= matchSet

```

```

matchSetAndListLabel(s,label,elementIndex,postmatchCode) ::= <<
<matchSet(...)>

```

```

<listLabelElem(elem=label,elemType=labelType,...)>
>>

/** Match a string literal */
lexerStringRef(string,label,elementIndex) ::= <%
<if(label)>
int <label>Start = CharIndex;<\n>
Match(<string>); <checkRuleBacktrackFailure()><\n>
int <label>StartLine<elementIndex> = Line;<\n>
int <label>StartCharPos<elementIndex> = CharPositionInLine;<\n>
<label> = new <labelType>(input, TokenTypes.Invalid, TokenChannels.Default, <label>Start, CharIndex-1);<\n>
<label>.Line = <label>StartLine<elementIndex>;<\n>
<label>.CharPositionInLine = <label>StartCharPos<elementIndex>;
<else>
Match(<string>); <checkRuleBacktrackFailure()><\n>
<endif>
%>

wildcard(token,label,elementIndex,terminalOptions={ }) ::= <<
<if(label)>
<label>=(<labelType>)input.LT(1);<\n>
<endif>
MatchAny(input); <checkRuleBacktrackFailure()>
>>

wildcardAndListLabel(token,label,elementIndex,terminalOptions={ }) ::= <<
<wildcard(...)>
<listLabelElem(elem=label,elemType=labelType,...)>
>>

/** Match . wildcard in lexer */
wildcardChar(label, elementIndex) ::= <<
<if(label)>
<label> = input.LA(1);<\n>
<endif>
MatchAny(); <checkRuleBacktrackFailure()>
>>

wildcardCharListLabel(label, elementIndex) ::= <<
<wildcardChar(...)>
<listLabelElem(elem=label,elemType=labelType,...)>
>>

/** Match a rule reference by invoking it possibly with arguments
* and a return value or values. The 'rule' argument was the
* target rule name, but now is type Rule, whose toString is
* same: the rule name. Now though you can access full rule
* descriptor stuff.

```

```

*/
ruleRef(rule,label,elementIndex,args,scope) ::= <<
PushFollow(Follow._<rule.name>_in_<ruleName><elementIndex>);
<if(label)><label>=<endof><if(scope)><scope.delegateName()>.<endof><rule.name; format="id">(<args;
separator=", ">);
PopFollow();
<checkRuleBacktrackFailure()>
>>

/** ids+=r */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabelElem(elem=label,elemType={<ASTLabelType>},...)>
>>

/** A lexer rule reference.
*
* The 'rule' argument was the target rule name, but now
* is type Rule, whose toString is same: the rule name.
* Now though you can access full rule descriptor stuff.
*/
lexerRuleRef(rule,label,args,elementIndex,scope) ::= <%
<if(label)>
int <label>Start<elementIndex> = CharIndex;<\n>
int <label>StartLine<elementIndex> = Line;<\n>
int <label>StartCharPos<elementIndex> = CharPositionInLine;<\n>
<if(scope)><scope.delegateName()>.<endof>m<rule.name>(<args; separator=", ">);
<checkRuleBacktrackFailure()><\n>
<label> = new <labelType>(input, TokenTypes.Invalid, TokenChannels.Default, <label>Start<elementIndex>,
CharIndex-1);<\n>
<label>.Line = <label>StartLine<elementIndex>;<\n>
<label>.CharPositionInLine = <label>StartCharPos<elementIndex>;
<else>
<if(scope)><scope.delegateName()>.<endof>m<rule.name>(<args; separator=", ">);
<checkRuleBacktrackFailure()>
<endof>
%>

/** i+=INT in lexer */
lexerRuleRefAndListLabel(rule,label,args,elementIndex,scope) ::= <<
<lexerRuleRef(...)>
<listLabelElem(elem=label,elemType=labelType,...)>
>>

/** EOF in the lexer */
lexerMatchEOF(label,elementIndex) ::= <%
<if(label)>
int <label>Start<elementIndex> = CharIndex;<\n>

```

```

int <label>StartLine<elementIndex> = Line;<\n>
int <label>StartCharPos<elementIndex> = CharPositionInLine;<\n>
Match(EOF); <checkRuleBacktrackFailure()><\n>
<labelType> <label> = new <labelType>(input, EOF, TokenChannels.Default, <label>Start<elementIndex>,
CharIndex-1);<\n>
<label>.Line = <label>StartLine<elementIndex>;<\n>
<label>.CharPositionInLine = <label>StartCharPos<elementIndex>;
<else>
Match(EOF); <checkRuleBacktrackFailure()>
<endif>
%>

// used for left-recursive rules
recRuleDefArg()          ::= "int <recRuleArg()>"
recRuleArg()             ::= "_p"
recRuleAltPredicate(ruleName,opPrec) ::= "<recRuleArg()> \<= <opPrec>"
recRuleSetResultAction() ::= "root_0=$<ruleName>_primary.tree;"

/** match ^(root children) in tree parser */
tree(root, actionsAfterRoot, children, nullableChildList,
    enclosingTreeLevel, treeLevel) ::= <<
<root:element()>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if (input.LA(1) == TokenTypes.Down)
{
    Match(input, TokenTypes.Down, null); <checkRuleBacktrackFailure()>
    <children:element()>
    Match(input, TokenTypes.Up, null); <checkRuleBacktrackFailure()>
}
<else>
Match(input, TokenTypes.Down, null); <checkRuleBacktrackFailure()>
<children:element()>
Match(input, TokenTypes.Up, null); <checkRuleBacktrackFailure()>
<endif>
>>

/** Every predicate is used as a validating predicate (even when it is
 * also hoisted into a prediction expression).
 */
validateSemanticPredicate(pred,description) ::= <<
if (!(<evalPredicate(...)>))
{
    <ruleBacktrackFailure()>
    throw new FailedPredicateException(input, "<ruleName>", "<description>");
}
>>

```

```
// F i x e d D F A (if-then-else)
```

```
dfaState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
int LA<decisionNumber>_<k> = input.LA(<k>);<\n>
<edges; separator="\nelse ">
else
{
<if(eotPredictsAlt)>
alt<decisionNumber> = <eotPredictsAlt>;
<else>
<ruleBacktrackFailure()>
NoViableAltException nvae = new NoViableAltException("<description>", <decisionNumber>, <stateNumber>,
input, <k>);
DebugRecognitionException(nvae);
<@noViableAltException()>
throw nvae;
<endif>
}
>>
```

```
/** Same as a normal DFA state except that we don't examine lookahead
 * for the bypass alternative. It delays error detection but this
 * is faster, smaller, and more what people expect. For (X)? people
 * expect "if ( LA(1)==X ) match(X);" and that's it.
 */
```

```
dfaOptionalBlockState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
int LA<decisionNumber>_<k> = input.LA(<k>);<\n>
<edges; separator="\nelse ">
>>
```

```
/** A DFA state that is actually the loopback decision of a closure
 * loop. If end-of-token (EOT) predicts any of the targets then it
 * should act like a default clause (i.e., no error can be generated).
 * This is used only in the lexer so that for ('a')* on the end of a rule
 * anything other than 'a' predicts exiting.
 */
```

```
dfaLoopbackState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
int LA<decisionNumber>_<k> = input.LA(<k>);<\n>
<edges; separator="\nelse "><\n>
<if(eotPredictsAlt)>
<if(!edges)>
alt<decisionNumber> = <eotPredictsAlt>;<! if no edges, don't gen ELSE !>
<else>
else
{
alt<decisionNumber> = <eotPredictsAlt>;
}<\n>
<endif>
```

```

<endif>
>>

/** An accept state indicates a unique alternative has been predicted */
dfaAcceptState(alt) ::= "alt<decisionNumber> = <alt>";

/** A simple edge with an expression. If the expression is satisfied,
 * enter to the target state. To handle gated productions, we may
 * have to evaluate some predicates for this edge.
 */
dfaEdge(labelExpr, targetState, predicates) ::= <<
if ((<labelExpr><if(predicates)> && (<predicates><endif>))
{
    <targetState>
}
>>

// F i x e d D F A (switch case)

/** A DFA state where a SWITCH may be generated. The code generator
 * decides if this is possible: CodeGenerator.canGenerateSwitch().
 */
dfaStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch (input.LA(<k>))
{
    <edges; separator="\n">
    default:
    <if(eotPredictsAlt)>
    alt<decisionNumber>=<eotPredictsAlt>;
    break;<\n>
    <else>
    {
        <ruleBacktrackFailure()>
        NoViableAltException nvae = new NoViableAltException("<description>", <decisionNumber>, <stateNumber>,
input, <k>);
        DebugRecognitionException(nvae);
        <@noViableAltException()>
        throw nvae;
    }
    <endif>
}
<\n>
>>

dfaOptionalBlockStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch (input.LA(<k>))
{
    <edges; separator="\n">
}
<\n>

```

```

>>

dfaLoopbackStateSwitch(k, edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch (input.LA(<k>))
{
<edges; separator="\n">
<if(eotPredictsAlt)>
default:
alt<decisionNumber>=<eotPredictsAlt>;
break;<\n>
<endif>
}<\n>
>>

```

```

dfaEdgeSwitch(labels, targetState) ::= <<
<labels: {it|case <it>:}; separator="\n">
{
<targetState>
}
break;
>>

```

```
// C y c l i c D F A
```

```

/** The code to initiate execution of a cyclic DFA; this is used
 * in the rule to predict an alt just like the fixed DFA case.
 * The <name> attribute is inherited via the parser, lexer, ...
 */

```

```

dfaDecision(decisionNumber,description) ::= <<
try
{
alt<decisionNumber> = dfa<decisionNumber>.Predict(input);
}
catch (NoViableAltException nvae)
{
DebugRecognitionException(nvae);
throw;
}
>>

```

```

/* Dump DFA tables as run-length-encoded Strings of octal values.
 * Can't use hex as compiler translates them before compilation.
 * These strings are split into multiple, concatenated strings.
 * Java puts them back together at compile time thankfully.
 * Java cannot handle large static arrays, so we're stuck with this
 * encode/decode approach. See analysis and runtime DFA for
 * the encoding methods.
 */

```

```

cyclicDFA(dfa) ::= <<
private class DFA<dfa.decisionNumber> : DFA
{
private const string DFA<dfa.decisionNumber>_eotS =
"<dfa.javaCompressedEOT; wrap="\n\t\t"">";
private const string DFA<dfa.decisionNumber>_eofS =
"<dfa.javaCompressedEOF; wrap="\n\t\t"">";
private const string DFA<dfa.decisionNumber>_minS =
"<dfa.javaCompressedMin; wrap="\n\t\t"">";
private const string DFA<dfa.decisionNumber>_maxS =
"<dfa.javaCompressedMax; wrap="\n\t\t"">";
private const string DFA<dfa.decisionNumber>_acceptS =
"<dfa.javaCompressedAccept; wrap="\n\t\t"">";
private const string DFA<dfa.decisionNumber>_specialS =
"<dfa.javaCompressedSpecial; wrap="\n\t\t"">>";
private static readonly string[] DFA<dfa.decisionNumber>_transitionS =
{
<dfa.javaCompressedTransition: {s|"<s; wrap="\n\t\t"">"; separator=",\n">
};

private static readonly short[] DFA<dfa.decisionNumber>_eot =
DFA.UnpackEncodedString(DFA<dfa.decisionNumber>_eotS);
private static readonly short[] DFA<dfa.decisionNumber>_eof =
DFA.UnpackEncodedString(DFA<dfa.decisionNumber>_eofS);
private static readonly char[] DFA<dfa.decisionNumber>_min =
DFA.UnpackEncodedStringToUnsignedChars(DFA<dfa.decisionNumber>_minS);
private static readonly char[] DFA<dfa.decisionNumber>_max =
DFA.UnpackEncodedStringToUnsignedChars(DFA<dfa.decisionNumber>_maxS);
private static readonly short[] DFA<dfa.decisionNumber>_accept =
DFA.UnpackEncodedString(DFA<dfa.decisionNumber>_acceptS);
private static readonly short[] DFA<dfa.decisionNumber>_special =
DFA.UnpackEncodedString(DFA<dfa.decisionNumber>_specialS);
private static readonly short[][] DFA<dfa.decisionNumber>_transition;

static DFA<dfa.decisionNumber>()
{
int numStates = DFA<dfa.decisionNumber>_transitionS.Length;
DFA<dfa.decisionNumber>_transition = new short[numStates][];
for ( int i=0; i < numStates; i++ )
{
DFA<dfa.decisionNumber>_transition[i] =
DFA.UnpackEncodedString(DFA<dfa.decisionNumber>_transitionS[i]);
}
}

public DFA<dfa.decisionNumber>( BaseRecognizer recognizer<if(dfa.specialStateSTs)>,
SpecialStateTransitionHandler specialStateTransition<endif> )
<if(dfa.specialStateSTs)>

```

```

: base(specialStateTransition)
<endif>
{
this.recognizer = recognizer;
this.decisionNumber = <dfa.decisionNumber>;
this.eot = DFA<dfa.decisionNumber>_eot;
this.eof = DFA<dfa.decisionNumber>_eof;
this.min = DFA<dfa.decisionNumber>_min;
this.max = DFA<dfa.decisionNumber>_max;
this.accept = DFA<dfa.decisionNumber>_accept;
this.special = DFA<dfa.decisionNumber>_special;
this.transition = DFA<dfa.decisionNumber>_transition;
}

public override string Description { get { return "<dfa.description>"; } }

public override void Error(NoViableAltException nvae)
{
    DebugRecognitionException(nvae);
}
}<\n>
<if(dfa.specialStateSTs)>
private int SpecialStateTransition<dfa.decisionNumber>(DFA dfa, int s, IIntStream _input)<! throws
NoViableAltException!>
{
    <if(LEXER)>
    IIntStream input = _input;
    <endif>
    <if(PARSER)>
    ITokenStream input = (ITokenStream)_input;
    <endif>
    <if(TREE_PARSER)>
    ITreeNodeStream input = (ITreeNodeStream)_input;
    <endif>
    int _s = s;
    s = -1;
    <! pull these outside the switch cases to save space on locals !>
    int LA<dfa.decisionNumber>_1 = input.LA(1);
    int index<dfa.decisionNumber>_1 = input.Index;
    switch (_s)
    {
    <dfa.specialStateSTs: {state |case <i0>:<! compressed special state numbers 0..n-1 !>
    <state>}; separator="\n">

    default:
    break;
    }
}

```

```

if (s >= 0)
    return s;

<if(backtracking)>
if (state.backtracking > 0) {state.failed=true; return -1;}
<endif>
NoViableAltException nvae = new NoViableAltException(dfa.Description, <dfa.decisionNumber>, _s, input);
dfa.Error(nvae);
throw nvae;
}
<endif>
>>

/** A state in a cyclic DFA; it's a special state and part of a big switch on
 * state.
 */
cyclicDFAState(decisionNumber,stateNumber,edges,needErrorClause,semPredState) ::= <<
{
<if(semPredState)>
<! get next lookahead symbol to test edges, then rewind !>
input.Rewind();
<endif>
<edges; separator="\nelse ">
<if(semPredState)>
<! return input cursor to state before we rewound !>
input.Seek(index<decisionNumber>_1);
<endif>
break;
}
>>

/** Just like a fixed DFA edge, test the lookahead and indicate what
 * state to jump to next if successful.
 */
cyclicDFAEdge(labelExpr, targetStateNumber, edgeNumber, predicates) ::= <<
if ((<labelExpr>)<if(predicates)> && (<predicates>)<endif>) {s = <targetStateNumber>;}<\n>
>>

/** An edge pointing at end-of-token; essentially matches any char;
 * always jump to the target.
 */
eotDFAEdge(targetStateNumber,edgeNumber, predicates) ::= <<
s = <targetStateNumber>;<\n>
>>

// D F A E X P R E S S I O N S

```

```

andPredicates(left,right) ::= "<left>&&<right>"

orPredicates(operands) ::= "<operands; separator=\\|\\>"

notPredicate(pred) ::= "!(<evalPredicate(...)>)"

evalPredicate(pred,description) ::= "<pred>"

evalSynPredicate(pred,description) ::= "EvaluatePredicate(<pred>_fragment)"

lookaheadTest(atom,k,atomAsInt) ::= "LA<decisionNumber>_<k>===<atom>"

/** Sometimes a lookahead test cannot assume that LA(k) is in a temp variable
 * somewhere. Must ask for the lookahead directly.
 */
isolatedLookaheadTest(atom,k,atomAsInt) ::= "input.LA(<k>)===<atom>"

lookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= <%
(LA<decisionNumber>_<k><ge()><lower> && LA<decisionNumber>_<k><le()><upper>)
%>

isolatedLookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::=
"(input.LA(<k>)<ge()><lower> && input.LA(<k>)<le()><upper>)"

le() ::= "\<="
ge() ::= ">="

setTest(ranges) ::= <<
<ranges; separator=\\|\\>
>>

// A T T R I B U T E S

attributeScope(scope) ::= <<
<if(scope)>
<if(scope.attributes)>
protected sealed partial class <scope.name>_scope
{
<scope.attributes:{ it|public <it.decl>; }; separator=\\n">

public <scope.name>_scope(<grammar.recognizerName> grammar) { OnCreated(grammar); }
partial void OnCreated(<grammar.recognizerName> grammar);
}
<if(scope.actions.scopeinit)>
protected void <scope.name>_scopeInit( <scope.name>_scope scope )
{
<scope.actions.scopeinit>
}

```

```

<else>
partial void <scope.name>_scopeInit( <scope.name>_scope scope );
<endif>
<if(scope.actions.scopeafter)>
protected void <scope.name>_scopeAfter( <scope.name>_scope scope )
{
    <scope.actions.scopeafter>
}
<else>
partial void <scope.name>_scopeAfter( <scope.name>_scope scope );
<endif>
protected readonly ListStack<<<scope.name>_scope> <scope.name>_stack = new
ListStack<<<scope.name>_scope>();
<endif>
<endif>
>>

globalAttributeScope(scope) ::= <<
<attributeScope(...)>
>>

ruleAttributeScope(scope) ::= <<
<attributeScope(...)>
>>

returnStructName(it) ::= "<it.name>_return"

returnType(ruleDescriptor) ::= <%
<if(ruleDescriptor.returnScope.attributes && ruleDescriptor.hasMultipleReturnValues)>
    <ruleDescriptor.grammar.recognizerName>.<ruleDescriptor:returnStructName()>
<elseif(ruleDescriptor.hasMultipleReturnValues)>
    <ruleReturnBaseType()>
<elseif(ruleDescriptor.hasSingleReturnValue)>
    <ruleDescriptor.singleValueReturnType>
<else>
    void
<endif>
%>

/** Generate the C# type associated with a single or multiple return
 * values.
 */
ruleLabelType(referencedRule) ::= <%
<if(referencedRule.hasMultipleReturnValues)>
    <ruleReturnBaseType()>
<elseif(referencedRule.hasSingleReturnValue)>
    <referencedRule.singleValueReturnType>
<else>

```

```

void
<endif>
%>

delegateName(it) ::= <<
<if(it.label)><it.label><else>g<it.name><endif>
>>

/** Using a type to init value map, try to init a type; if not in table
 * must be an object, default value is "null".
 */
initValue(typeName) ::= <<
default(<typeName>)
>>

/** Define a rule label including default value */
ruleLabelDef(label) ::= <%
<ruleLabelType(label.referencedRule)> <label.label.text> = <initValue(ruleLabelType(label.referencedRule))>;
%>

/** Define a return struct for a rule if the code needs to access its
 * start/stop tokens, tree stuff, attributes, ... Leave a hole for
 * subgroups to stick in members.
 */
returnScope(scope) ::= <<
<if(scope.attributes && ruleDescriptor.hasMultipleReturnValues)>
<returnScopeModifier(grammar=grammar,ruleDescriptor=ruleDescriptor)> sealed partial class
<ruleDescriptor:returnStructName()> : <ruleReturnBaseType()><@ruleReturnInterfaces()>
{
<scope.attributes: {it|public <it.decl>;}; separator="\n">
<@ruleReturnMembers()>
}
<endif>
>>

ruleReturnBaseType() ::= <%
<if(TREE_PARSER)>Tree<else>Parser<endif>RuleReturnScope\<<labelType>>
%>

@returnScope.ruleReturnMembers() ::= <<
public <ruleDescriptor:returnStructName()>(<grammar.recognizerName> grammar) { OnCreated(grammar);}
partial void OnCreated(<grammar.recognizerName> grammar);
>>

parameterScope(scope) ::= <<
<scope.attributes: {it|<it.decl>;}; separator=", ">
>>

```

```

parameterAttributeRef(attr) ::= <<
<attr.name; format="id">
>>

parameterSetAttributeRef(attr,expr) ::= <<
<attr.name; format="id"> =<expr>;
>>

scopeAttributeRef(scope,attr,index,negIndex) ::= <%
<if(negIndex)>
<scope>_stack[<scope>_stack.Count - <negIndex> - 1].<attr.name; format="id">
<else>
<if(index)>
<scope>_stack[<index>].<attr.name; format="id">
<else>
<scope>_stack.Peek().<attr.name; format="id">
<endif>
<endif>
%>

scopeSetAttributeRef(scope,attr,expr,index,negIndex) ::= <%
<if(negIndex)>
<scope>_stack[<scope>_stack.Count - <negIndex> - 1].<attr.name; format="id"> = <expr>;
<else>
<if(index)>
<scope>_stack[<index>].<attr.name; format="id"> = <expr>;
<else>
<scope>_stack.Peek().<attr.name; format="id"> = <expr>;
<endif>
<endif>
%>

/** $x is either global scope or x is rule with dynamic scope; refers
* to stack itself not top of stack. This is useful for predicates
* like {$function.Count>0 && $function::name.Equals("foo")}?
*/
isolatedDynamicScopeRef(scope) ::= "<scope>_stack"

/** reference an attribute of rule; might only have single return value */
ruleLabelRef(referencedRule,scope,attr) ::= <%
<if(referencedRule.hasMultipleReturnValues)>
(<scope>!=null?(<returnType(referencedRule)><scope>).<attr.name; format="id">:<initValue(attr.type)>)
<else>
<scope>
<endif>
%>

returnAttributeRef(ruleDescriptor,attr) ::= <%

```

```

<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name; format="id">
<else>
<attr.name; format="id">
<endif>
%>

returnSetAttributeRef(ruleDescriptor,attr,expr) ::= <%
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name; format="id"> =<expr>;
<else>
<attr.name; format="id"> =<expr>;
<endif>
%>

/** How to translate $tokenLabel */
tokenLabelRef(label) ::= "<label>"

/** ids+=ID {$ids} or e+=expr {$e} */
listLabelRef(label) ::= "list_<label>"

// not sure the next are the right approach

tokenLabelPropertyRef_text(scope,attr) ::= "<scope>!=null?<scope>.Text:default(string)"
tokenLabelPropertyRef_type(scope,attr) ::= "<scope>!=null?<scope>.Type:0"
tokenLabelPropertyRef_line(scope,attr) ::= "<scope>!=null?<scope>.Line:0"
tokenLabelPropertyRef_pos(scope,attr) ::= "<scope>!=null?<scope>.CharPositionInLine:0)"
tokenLabelPropertyRef_channel(scope,attr) ::= "<scope>!=null?<scope>.Channel:0"
tokenLabelPropertyRef_index(scope,attr) ::= "<scope>!=null?<scope>.TokenIndex:0)"
tokenLabelPropertyRef_tree(scope,attr) ::= "<scope>_tree"
tokenLabelPropertyRef_int(scope,attr) ::= "<scope>!=null?int.Parse(<scope>.Text):0)"

ruleLabelPropertyRef_start(scope,attr) ::= "<scope>!=null?((<labelType>)<scope>.Start):default(<labelType>)"
ruleLabelPropertyRef_stop(scope,attr) ::= "<scope>!=null?((<labelType>)<scope>.Stop):default(<labelType>)"
ruleLabelPropertyRef_tree(scope,attr) ::=
"<scope>!=null?((<ASTLabelType>)<scope>.Tree):default(<ASTLabelType>)"
ruleLabelPropertyRef_text(scope,attr) ::= <%
<if(TREE_PARSER)>
(<scope>!=null?(input.TokenStream.ToString(
input.TreeAdaptor.GetTokenStartIndex(<scope>.Start),
input.TreeAdaptor.GetTokenStopIndex(<scope>.Start))):default(string))
<else>
(<scope>!=null?input.ToString(<scope>.Start,<scope>.Stop):default(string))
<endif>
%>

ruleLabelPropertyRef_st(scope,attr) ::= "<scope>!=null?<scope>.Template:null)"

```

```

/** Isolated $RULE ref ok in lexer as it's a Token */
lexerRuleLabel(label) ::= "<label>"

lexerRuleLabelPropertyRef_type(scope,attr) ::=
  "<scope>!=null?<scope>.Type:0)"

lexerRuleLabelPropertyRef_line(scope,attr) ::=
  "<scope>!=null?<scope>.Line:0)"

lexerRuleLabelPropertyRef_pos(scope,attr) ::=
  "<scope>!=null?<scope>.CharPositionInLine:-1)"

lexerRuleLabelPropertyRef_channel(scope,attr) ::=
  "<scope>!=null?<scope>.Channel:0)"

lexerRuleLabelPropertyRef_index(scope,attr) ::=
  "<scope>!=null?<scope>.TokenIndex:0)"

lexerRuleLabelPropertyRef_text(scope,attr) ::=
  "<scope>!=null?<scope>.Text:default(string))"

lexerRuleLabelPropertyRef_int(scope,attr) ::=
  "<scope>!=null?int.Parse(<scope>.Text):0)"

// Somebody may ref $template or $tree or $stop within a rule:
rulePropertyRef_start(scope,attr) ::= "retval.Start"
rulePropertyRef_stop(scope,attr) ::= "retval.Stop"
rulePropertyRef_tree(scope,attr) ::= "retval.Tree"
rulePropertyRef_text(scope,attr) ::= <%
<if(TREE_PARSER)>
input.TokenStream.ToString(
  input.TreeAdaptor.GetTokenStartIndex(retval.Start),
  input.TreeAdaptor.GetTokenStopIndex(retval.Start))
<else>
input.ToString(retval.Start,input.LT(-1))
<endif>
%>
rulePropertyRef_st(scope,attr) ::= "retval.Template"

lexerRulePropertyRef_text(scope,attr) ::= "Text"
lexerRulePropertyRef_type(scope,attr) ::= "_type"
lexerRulePropertyRef_line(scope,attr) ::= "state.tokenStartLine"
lexerRulePropertyRef_pos(scope,attr) ::= "state.tokenStartCharPositionInLine"
lexerRulePropertyRef_index(scope,attr) ::= "-1" // undefined token index in lexer
lexerRulePropertyRef_channel(scope,attr) ::= "_channel"
lexerRulePropertyRef_start(scope,attr) ::= "state.tokenStartCharIndex"
lexerRulePropertyRef_stop(scope,attr) ::= "(CharIndex-1)"

```

```

lexerRulePropertyRef_int(scope,attr) ::= "int.Parse(<scope>.Text)"

// setting $st and $tree is allowed in local rule. everything else
// is flagged as error
ruleSetPropertyRef_tree(scope,attr,expr) ::= "retval.Tree = <expr>";
ruleSetPropertyRef_st(scope,attr,expr) ::= "retval.Template =<expr>";

/** How to execute an action (only when not backtracking) */
execAction(action) ::= <%
<if(backtracking)>
if (<actions.(actionScope).synpredgate><\n>
{<\n>
<@indentedAction()><\n>
}
<else>
<action>
<endif>
%>

@execAction.indentedAction() ::= <<
<action>
>>

/** How to always execute an action even when backtracking */
execForcedAction(action) ::= "<action>"

// M I S C (properties, etc...)

bitset(name, words64) ::= <<
public static readonly BitSet <name> = new BitSet(new ulong[] {<words64: {it<it>UL};separator=",">});
>>

codeFileExtension() ::= ".cs"

true_value() ::= "true"
false_value() ::= "false"

Found in path(s):
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-
jar/org/antlr/codegen/templates/CSharp3/CSharp3.stg
No license file was found, but licenses were detected in source scan.

/*
[The "BSD license"]
Copyright (c) 2005-2012 Terence Parr
All rights reserved.

Redistribution and use in source and binary forms, with or without

```

modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

/** Templates for building ASTs during normal parsing.

*

* Deal with many combinations. Dimensions are:

* Auto build or rewrite

* no label, label, list label (label/no-label handled together)

* child, root

* token, set, rule, wildcard

*

* The situation is not too bad as rewrite (->) usage makes ^ and !

* invalid. There is no huge explosion of combinations.

*/

```
finishedBacktracking(block) ::= <<
```

```
<if(!ruleDescriptor.isSynPred)>
```

```
<if(backtracking)>
```

```
if <actions.(actionScope).synpredgate>:
```

```
  <block>
```

```
<else>
```

```
<block>
```

```
<endif>
```

```
<endif>
```

```
>>
```

```
@ruleBody.setErrorReturnValue() ::= <<
```

```
retval.tree = self._adaptor.errorNode(self.input, retval.start, self.input.LT(-1), re)
```

```
>>
```

```

// TOKEN AST STUFF

/** ID and output=AST */
tokenRef(token,label,elementIndex,terminalOptions={}) ::= <<
<super.tokenRef(...)>
<finishedBacktracking({
<label>_tree = <createNodeFromToken(...)>
self._adaptor.addChild(root_0, <label>_tree)
})>
>>

/** ID! and output=AST (same as plain tokenRef) */
tokenRefBang(token,label,elementIndex,terminalOptions={}) ::= "<super.tokenRef(...)>"

/** ID^ and output=AST */
tokenRefRuleRoot(token,label,elementIndex,terminalOptions={}) ::= <<
<super.tokenRef(...)>
<finishedBacktracking({
<label>_tree = <createNodeFromToken(...)>
root_0 = self._adaptor.becomeRoot(<label>_tree, root_0)
})>
>>

/** ids+=ID! and output=AST */
tokenRefBangAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRefBang(...)>
<listLabel(elem=label,...)>
>>

/** label+=TOKEN when output=AST but not rewrite alt */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

/** Match label+=TOKEN^ when output=AST but not rewrite alt */
tokenRefRuleRootAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRefRuleRoot(...)>
<listLabel(elem=label,...)>
>>

// SET AST

// the match set stuff is interesting in that it uses an argument list
// to pass code to the default matchSet; another possible way to alter
// inherited code. I don't use the region stuff because I need to pass
// different chunks depending on the operator. I don't like making

```

```

// the template name have the operator as the number of templates gets
// large but this is the most flexible--this is as opposed to having
// the code generator call matchSet then add root code or ruleroot code
// plus list label plus ... The combinations might require complicated
// rather than just added on code. Investigate that refactoring when
// I have more time.

matchSet(s,label,elementIndex,postmatchCode,terminalOptions={}) ::= <%
<super.matchSet(postmatchCode={<finishedBacktracking({self._adaptor.addChild(root_0,
<createNodeFromToken(...)>)}>}, ...)>
%>

matchRuleBlockSet(s,label,elementIndex,postmatchCode,treeLevel="0",terminalOptions={}) ::= <<
<matchSet(...)>
>>

matchSetBang(s,label,elementIndex,postmatchCode,terminalOptions={}) ::= "<super.matchSet(...)>"

// note there is no matchSetTrack because -> rewrites force sets to be
// plain old blocks of alts: (A|B|...|C)

matchSetRuleRoot(s,label,elementIndex,debug,terminalOptions={}) ::= <<
<if(label)>
<label> = self.input.LT(1)<\n>
<endif>
<super.matchSet(postmatchCode={<finishedBacktracking({root_0 =
self._adaptor.becomeRoot(<createNodeFromToken(...)>, root_0)}>}, ...)>
>>

// RULE REF AST

/** rule when output=AST */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<finishedBacktracking({self._adaptor.addChild(root_0, <label>.tree)}>)}>
>>

/** rule! is same as normal rule ref */
ruleRefBang(rule,label,elementIndex,args,scope) ::= "<super.ruleRef(...)>"

/** rule^ */
ruleRefRuleRoot(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<finishedBacktracking({root_0 = self._adaptor.becomeRoot(<label>.tree, root_0)}>)}>
>>

/** x+=rule when output=AST */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<

```

```

<ruleRef(...)>
<listLabel(label, {<label>.tree})>
>>

/** x+=rule! when output=AST is a rule ref with list addition */
ruleRefBangAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefBang(...)>
<listLabel(label, {<label>.tree})>
>>

/** x+=rule^ */
ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<listLabel(label, {<label>.tree})>
>>

// WILDCARD AST

wildcard(token,label,elementIndex,terminalOptions={ }) ::= <<
<super.wildcard(...)>
<finishedBacktracking({
<label>_tree = self._adaptor.createWithPayload(<label>)
self._adaptor.addChild(root_0, <label>_tree)
})>
>>

wildcardBang(label,elementIndex) ::= "<super.wildcard(...)>"

wildcardRuleRoot(token,label,elementIndex,terminalOptions={ }) ::= <<
<super.wildcard(...)>
<finishedBacktracking({
<label>_tree = self._adaptor.createWithPayload(<label>)
root_0 = self._adaptor.becomeRoot(<label>_tree, root_0)
})>
>>

createNodeFromToken(label,terminalOptions={ }) ::= <%
<if(terminalOptions.node)>
<terminalOptions.node>(<label>) <! new MethodNode(IDLabel) !>
<else>
self._adaptor.createWithPayload(<label>)
<endif>
%>

ruleCleanup() ::= <<
<super.ruleCleanup()>
<finishedBacktracking({
retval.tree = self._adaptor.rulePostProcessing(root_0)

```

```
self._adaptor.setTokenBoundaries(retval.tree, retval.start, retval.stop)
})>
>>
```

Found in path(s):

```
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-
jar/org/antlr/codegen/templates/Python3/ASTParser.stg
No license file was found, but licenses were detected in source scan.
```

```
/*
[The "BSD license"]
Copyright (c) 2005-2009 Jim Idle, Temporal Wave LLC
http://www.temporal-wave.com
http://www.linkedin.com/in/jimidle
```

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```
*/
```

```
/** Template overrides to add debugging to AST stuff. Dynamic inheritance
```

```
* hierarchy is set up as ASTDbg : AST : Dbg : Java by code generator.
```

```
*/
```

```
parserMembers() ::= <<
protected DebugTreeAdaptor adaptor =
    new DebugTreeAdaptor(null,new CommonTreeAdaptor());
public void setTreeAdaptor(TreeAdaptor adaptor) {
    this.adaptor = new DebugTreeAdaptor(dbg,adaptor);
```

```
}  
public TreeAdaptor getTreeAdaptor() {  
    return adaptor;  
}<\n>  
>>
```

```
parserCtorBody() ::= <<  
>>
```

```
createListenerAndHandshake() ::= <<  
<super.createListenerAndHandshake(>  
>>
```

```
ctorForPredefinedListener() ::= <<  
>>
```

```
@rewriteElement.pregen() ::= ""
```

Found in path(s):

```
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-  
jar/org/antlr/codegen/templates/C/ASTDbg.stg
```

No license file was found, but licenses were detected in source scan.

```
/*
```

```
[The "BSD license"]
```

```
Copyright (c) 2006 Kay Roepke 2010 Alan Condit
```

```
All rights reserved.
```

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF

THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

Found in path(s):

* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/ObjC/Dbg.stg

No license file was found, but licenses were detected in source scan.

/*

[The "BSD license"]

Copyright (c) 2006, 2007 Kay Roepke 2010 Alan Condit

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

/*

* Template group file for the Objective C code generator.

* Heavily based on Java.stg

*

* Written by Kay Roepke <kroepke(at)classdump.org>

* Modified by Alan Condit <acondit(at)ipns.com>

*

* This file is part of ANTLR and subject to the same license as ANTLR itself.

*/

objcTypeInitMap ::= [

"int" : "0", // Integers start out being 0

"long" : "0", // Longs start out being 0

```

"float"    : "0.0",      // Floats   start out being 0
"double"   : "0.0",      // Doubles  start out being 0
"BOOL"     : "NO",       // Booleans  start out being Antlr ObjC for false
"byte"     : "0",        // Bytes    start out being 0
"short"    : "0",        // Shorts   start out being 0
"char"     : "0",        // Chars    start out being 0
"id"       : "nil",      // ids      start out being nil
default    : "nil"       // anything other than an atomic type
]

// System.Boolean.ToString() returns "True" and "False", but the proper C# literals are "true" and "false"
// The Java version of Boolean returns "true" and "false", so they map to themselves here.
booleanLiteral ::= [
  "True":"true",
  "False":"false",
  "true":"YES",
  "false":"NO",
  default:"NO"
]

className() ::= "<name><!<if(LEXER)>Lexer<else><if(TREE_PARSER)>Tree<endif>Parser<endif>!>"
leadIn(type) ::=
<<
/** \file
 * This <type> file was generated by $ANTLR version <ANTLRVersion>
 *
 * - From the grammar source file : <fileName>
 * -           On : <generatedTimestamp>
<if(LEXER)>
 * -           for the lexer : <name>Lexer
<endif>
<if(PARSER)>
 * -           for the parser : <name>Parser
<endif>
<if(TREE_PARSER)>
 * -           for the tree parser : <name>TreeParser
<endif>
 *
 * Editing it, at least manually, is not wise.
 *
 * ObjC language generator and runtime by Alan Condit, acondit|hereisanat|ipns|dotgoeshere|com.
 *
 *
 *
>>

/** The overall file structure of a recognizer; stores methods for rules
 * and cyclic DFAs plus support code.

```

```

*/
outputFile( LEXER,
    PARSEr,
    TREE_PARSER,
    actionScope,
    actions,
    docComment,
    recognizer,
    name,
    tokens,
    tokenNames,
    rules,
    cyclicDFAs,
    bitsets,
    buildTemplate,
    buildAST,
    rewriteMode,
    profile,
    backtracking,
    synpreds,
    memoize,
    numRules,
    fileName,
    ANTLRVersion,
    generatedTimestamp,
    trace,
    scopes,
    superClass,
    literals
) ::=

<<
<leadIn("OBJC source")>
*/
// $ANTLR <ANTLRVersion> <fileName> <generatedTimestamp>

<! <if(actions.(actionScope).header)>
/* =====
* This is what the grammar programmer asked us to put at the top of every file.
*/
<actions.(actionScope).header>
/* End of Header action.
* =====
*/
<endif> !>

/* -----
* Include the ANTLR3 generated header file.
*/

```

```

#import " <name><!<if(LEXER)>Lexer<else><if(TREE_PARSER)>Tree<endif>Parser<endif>!>.h"
<actions.(actionScope).postinclude>
/* ----- */

<docComment>

<if(literals)>
/** String literals used by <name> that we must do things like MATCHS() with.
 * C will normally just lay down 8 bit characters, and you can use L"xxx" to
 * get wchar_t, but wchar_t is 16 bits on Windows, which is not UTF32 and so
 * we perform this little trick of defining the literals as arrays of UINT32
 * and passing in the address of these.
 */
<literals:{it | static ANTLR3_UCHAR lit_<i>[] = <it>;}; separator="\n">

<endif>

/* ===== */
/* ===== */
 * Start of recognizer
 */
<recognizer>
>>
headerFileExtension() ::= ".h"

headerFile( LEXER,
            PARSEr,
            TREE_PARSER,
            actionScope,
            actions,
            docComment,
            recognizer,
            name,
            tokens,
            tokenNames,
            rules,
            cyclicDFAs,
            bitsets,
            buildTemplate,
            buildAST,
            rewriteMode,
            profile,
            backtracking,
            synpreds,
            memoize,
            numRules,
            fileName,
            ANTLRVersion,

```

```

        generatedTimestamp,
        trace,
        scopes,
        superClass,
        literals
    ) ::=
<<
// $ANTLR <ANTLRVersion> <fileName> <generatedTimestamp>

<@imports>
<actions.(actionScope).preincludes>
/* =====
* Standard antlr OBJC runtime definitions
*/
#import \<Cocoa/Cocoa.h>
#import \<ANTLR/ANTLR.h>
/* End of standard antlr3 runtime definitions
* =====
*/
<actions.(actionScope).includes>
<@end>

<if(LEXER)>
<lexerHeaderFile(...)>
<endif>
<if(PARSER)>
<parserHeaderFile(...)>
<endif>
<if(TREE_PARSER)>
<treeParserHeaderFile(...)>
<endif>
<docComment>
>>

lexerHeaderFile( LEXER,
    PARSER,
    TREE_PARSER,
    actionScope,
    actions,
    docComment,
    recognizer,
    name,
    tokens,
    tokenNames,
    rules,
    cyclicDFAs,
    bitsets,
    buildTemplate,

```

```

        profile,
        backtracking,
        synpreds,
        memoize,
        numRules,
        fileName,
        ANTLRVersion,
        generatedTimestamp,
        trace,
        scopes,
        superClass="Lexer"
    ) ::=
<<

<if(actions.(actionScope).header)>
/* =====
* This is what the grammar programmer asked us to put at the top of every file.
*/
<actions.(actionScope).header>
/* End of Header action.
* =====
*/
<endif>

/* Start cyclicDFAInterface */
<cyclicDFAs:cyclicDFAInterface()>

#pragma mark Rule return scopes Interface start
<rules:{rule |
<rule.ruleDescriptor:{ruleDescriptor | <returnScopeInterface(scope=ruleDescriptor.returnScope)>>>}>
#pragma mark Rule return scopes Interface end
#pragma mark Tokens
#ifndef TOKENLISTAlreadyDefined
#define TOKENLISTAlreadyDefined 1
#ifdef EOF
#undef EOF
#endif
<tokens:{it | #define <it.name> <it.type>; separator="\n">
#endif
/* interface lexer class */
@interface <className()> <@superClassName>: <superClass><@end> { // line 283
<cyclicDFAs:{dfa | DFA<dfa.decisionNumber> *dfa<dfa.decisionNumber>;}; separator="\n">
<synpreds:{pred | SEL <pred>Selector;}; separator="\n">
/* ObjC start of actions.lexer.memVars */
<actions.lexer.memVars>
/* ObjC end of actions.lexer.memVars */
}
+ (void) initialize;

```

```

+ (<className()> *)new<className()>WithCharStream:(id\<CharStream>)anInput;
/* ObjC start actions.lexer.methodsDecl */
<actions.lexer.methodsDecl>
/* ObjC end actions.lexer.methodsDecl */
<rules:{ rule |
- (<rule.ruleDescriptor:{ ruleDescriptor|<returnType()>}>)
<if(!rule.ruleDescriptor.isSynPred)>m<rule.ruleName><else><rule.ruleName>_fragment<endif>
<if(rule.ruleDescriptor.parameterScope)><rule.ruleDescriptor.parameterScope:parameterScope()><endif>; };
separator="\n"><\n>
@end /* end of <className()> interface */<\n>
>>

```

```

headerReturnScope(ruleDescriptor) ::= "<returnScopeInterface(...)>"
headerReturnType(ruleDescriptor) ::= <<
<if(LEXER)>
<if(!r.ruleDescriptor.isSynPred)>
void
<else>
<ruleDescriptor:returnType()>
<endif>
<else>
<ruleDescriptor:returnType()>
<endif>
>>

```

```
// Produce the lexer output
```

```

lexer( grammar,
    name,
    tokens,
    scopes,
    rules,
    numRules,
    filterMode,
    labelType="CommonToken",
    superClass="Lexer"
) ::= <<
<cyclicDFAs:cyclicDFA()>

```

```
/** As per Terence: No returns for lexer rules! */
```

```

<!
#pragma mark Rule return scopes start
<rules:{ rule | <rule.ruleDescriptor:{ ruleDescriptor |
<returnScopeImplementation(scope=ruleDescriptor.returnScope)>}>
}>
#pragma mark Rule return scopes end
!>
@implementation <grammar.recognizerName> // line 330

```

```
+ (void) initialize
```

```

{
    [BaseRecognizer setGrammarFileName:@"<fileName>"];
}

+ (NSString *) tokenNameForType:(NSInteger)aTokenType
{
    return [[self getTokenNames] objectAtIndex:aTokenType];
}

+ (<grammar.recognizerName> *)new<grammar.recognizerName>WithCharStream:(id<CharStream>)anInput
{
    return [[<grammar.recognizerName> alloc] initWithCharStream:anInput];
}

- (id) initWithCharStream:(id<CharStream>)anInput
{
    self = [super initWithCharStream:anInput State:[RecognizerSharedState
newRecognizerSharedStateWithRuleLen:<numRules>+1]];
    if ( self != nil ) {
<if(memoize)>
        if ( state.ruleMemo == nil ) {
            state.ruleMemo = [[RuleStack newRuleStackWithSize:<numRules>+1] retain];
        }
        if ( [state.ruleMemo count] == 0 ) {
            // initialize the memoization cache - the indices are 1-based in the runtime code!
            <! [state.ruleMemo addObject:[NSNull null]]; /* dummy entry to ensure 1-basedness. */ !>
            for (NSInteger i = 0; i < <numRules>; i++) {
                [state.ruleMemo addObject:[HashRule newHashRuleWithLen:17]];
            }
        }
    }
<endif>
    <synpreds: {pred | <lexerSynpred(name=pred)>};separator="\n">
    <cyclicDFAs: {dfa | dfa<dfa.decisionNumber> = [DFA<dfa.decisionNumber>
newDFA<dfa.decisionNumber>WithRecognizer:self];}; separator="\n">
    <actions.lexer.init>
}
    return self;
}

- (void) dealloc
{
    <cyclicDFAs: {dfa | [dfa<dfa.decisionNumber> release];}; separator="\n">
<actions.lexer.dealloc>
    [super dealloc];
}

/* ObjC Start of actions.lexer.methods */
<actions.lexer.methods>

```

```

/* ObjC end of actions.lexer.methods */
/* ObjC start methods() */
<@methods()>
/* ObjC end methods() */

<if(actions.lexer.reset)>
- (void) reset
{
    <actions.lexer.reset>
    [super reset];
}
<endif>

<if(filterMode)>
<filteringNextToken()>
<endif>
/* Start of Rules */
<rules; separator="\n">

@end /* end of <grammar.recognizerName> implementation line 397 */
>>

/** A override of Lexer.nextToken() that backtracks over mTokens() looking
 * for matches. No error can be generated upon error; just rewind, consume
 * a token and then try again. backtracking needs to be set as well.
 * Make rule memoization happen only at levels above 1 as we start mTokens
 * at backtracking==1.
 */
filteringNextToken() ::= <<
- (id\<Token>) nextToken
{
    while (YES) {
        if ( [input LA:1] == CharStreamEOF ) {
            return [<labelType> eofToken];
        }
        state.token = nil;
        state.channel = TokenChannelDefault;
        state.tokenStartCharIndex = input.index;
        state.tokenStartCharPositionInLine = input.charPositionInLine;
        state.tokenStartLine = input.line;
        state.text = nil;
        @try {
            NSInteger m = [input mark];
            state.backtracking = 1; /* means we won't throw slow exception */
            state.failed = NO;
            [self mTokens];
            state.backtracking = 0;
            /* mTokens backtracks with synpred at backtracking==2

```

```

        and we set the synpredgate to allow actions at level 1. */
    if ( state.failed ) {
        [input rewind:m];
        [input consume]; /* advance one char and try again */
    } else {
        [self emit];
        return state.token;
    }
}
}
@catch (RecognitionException *re) {
    // shouldn't happen in backtracking mode, but...
    [self reportError:re];
    [self recover:re];
}
}
}
}

```

```

- (void)memoize:(id<IntStream>)anInput
  RuleIndex:(NSInteger)ruleIndex
  StartIndex:(NSInteger)ruleStartIndex
{
    if ( state.backtracking > 1 ) [super memoize:anInput RuleIndex:ruleIndex StartIndex:ruleStartIndex];
}

```

```

- (BOOL)alreadyParsedRule:(id<IntStream>)anInput RuleIndex:(NSInteger)ruleIndex
{
    if ( state.backtracking > 1 ) return [super alreadyParsedRule:anInput RuleIndex:ruleIndex];
    return NO;
}
>>

```

```

actionGate() ::= "state.backtracking == 0"

```

```

filteringActionGate() ::= "state.backtracking == 1"

```

```

parserHeaderFile( LEXER,
    PARSER,
    TREE_PARSER,
    actionScope,
    actions,
    docComment,
    recognizer,
    name,
    tokens,
    tokenNames,
    rules,
    cyclicDFAs,
    bitsets,

```

```

    buildTemplate,
    profile,
    backtracking,
    synpreds,
    memoize,
    numRules,
    fileName,
    ANTLRVersion,
    generatedTimestamp,
    trace,
    scopes,
    literals,
    superClass="Parser"
  ) ::= <<
/* parserHeaderFile */
<genericParserHeaderFile(inputStreamType="id\<TokenStream>",...)>
>>

treeParserHeaderFile( LEXER,
  PARSEr,
  TREE_PARSEr,
  actionScope,
  actions,
  docComment,
  recognizer,
  name,
  tokens,
  tokenNames,
  rules,
  cyclicDFAs,
  bitsets,
  buildTemplate,
  profile,
  backtracking,
  synpreds,
  memoize,
  numRules,
  fileName,
  ANTLRVersion,
  generatedTimestamp,
  trace,
  scopes,
  literals,
  superClass="TreeParser"
) ::= <<
/* treeParserHeaderFile */
<genericParserHeaderFile(inputStreamType="id\<TreeNodeStream>",...)>
>>

```

```

genericParserHeaderFile( LEXER,
    PARSER,
    TREE_PARSER,
    actionScope,
    actions,
    docComment,
    recognizer,
    name,
    tokens,
    tokenNames,
    rules,
    cyclicDFAs,
    bitsets,
    buildTemplate,
    profile,
    backtracking,
    synpreds,
    memoize,
    numRules,
    fileName,
    ANTLRVersion,
    generatedTimestamp,
    trace,
    scopes,
    superClass,
    literals,
    inputStreamType
) ::=

<<
<if(actions.(actionScope).header)>
/* =====
* This is what the grammar programmer asked us to put at the top of every file.
*/
<actions.(actionScope).header>
/* End of Header action.
* =====
*/
<endif>

#ifndef ANTLR3TokenTypeAlreadyDefined
#define ANTLR3TokenTypeAlreadyDefined
typedef enum {
    ANTLR_EOF = -1,
    INVALID,
    EOR,
    DOWN,
    UP,

```

```

    MIN
} ANTLR3TokenType;
#endif

<cyclicDFAs:cyclicDFAInterface()>
#pragma mark Tokens
#ifndef TOKENLISTAlreadyDefined
#define TOKENLISTAlreadyDefined 1
#endif
EOF
#undef EOF
#endif
<tokens:{it | #define <it.name> <it.type>}; separator="\n">
#endif
#pragma mark Dynamic Global Scopes globalAttributeScopeInterface
<scopes:{it | <if(it.isDynamicGlobalScope)><globalAttributeScopeInterface(scope=it)><endif>}>
#pragma mark Dynamic Rule Scopes ruleAttributeScopeInterface
<rules:{rule |
<rule.ruleDescriptor:{ ruleDescriptor | <ruleAttributeScopeInterface(scope=ruleDescriptor.ruleScope)>>>}>
#pragma mark Rule Return Scopes returnScopeInterface
<rules:{rule |<rule.ruleDescriptor:{ ruleDescriptor |
<returnScopeInterface(scope=ruleDescriptor.returnScope)>>>}>

/* Interface grammar class */
@interface <className()> <@superClassName> : <superClass><@end> { /* line 572 */
#pragma mark Dynamic Rule Scopes ruleAttributeScopeDecl
<rules:{rule | <rule.ruleDescriptor.ruleScope:ruleAttributeScopeDecl(scope=rule.ruleDescriptor.ruleScope)>>}>
#pragma mark Dynamic Global Rule Scopes globalAttributeScopeMemVar
<scopes:{it | <if(it.isDynamicGlobalScope)><globalAttributeScopeMemVar(scope=it)><endif>}><\n>
/* ObjC start of actions.(actionScope).memVars */
<actions.(actionScope).memVars>
/* ObjC end of actions.(actionScope).memVars */
/* ObjC start of memVars */
<@memVars()>
/* ObjC end of memVars */

<cyclicDFAs:{ dfa | DFA<dfa.decisionNumber> *dfa<dfa.decisionNumber>;}; separator="\n">
<synpreds:{ pred | SEL <pred>Selector;}; separator="\n">
}

/* ObjC start of actions.(actionScope).properties */
<actions.(actionScope).properties>
/* ObjC end of actions.(actionScope).properties */
/* ObjC start of properties */
<@properties()>
/* ObjC end of properties */

+ (void) initialize;
+ (<className()> *) new<className()>:(<inputStreamType>)aStream;

```

```

/* ObjC start of actions.(actionScope).methodsDecl */
<actions.(actionScope).methodsDecl>
/* ObjC end of actions.(actionScope).methodsDecl */

/* ObjC start of methodsDecl */
<@methodsDecl()>
/* ObjC end of methodsDecl */

<rules:{rule |
-
(<rule.ruleDescriptor:{ruleDescriptor|<returnType()>})><if(!rule.ruleDescriptor.isSynPred)><rule.ruleName><else
><rule.ruleName>_fragment<endif><if(rule.ruleDescriptor.parameterScope)><rule.ruleDescriptor.parameterScope:
parameterScope()><endif>; }; separator="\n"><\n>

@end /* end of <className()> interface */<\n>
>>

parser( grammar,
    name,
    scopes,
    tokens,
    tokenNames,
    rules,
    numRules,
    bitsets,
    ASTLabelType="CommonTree",
    superClass="Parser",
    labelType="CommonToken",
    members={<actions.parser.members>}
) ::= <<
<genericParser(inputStreamType="id<TokenStream>", rewriteElementType="Token", ...)>
>>

/** How to generate a tree parser; same as parser except the input
 * stream is a different type.
 */
treeParser( grammar,
    name,
    scopes,
    tokens,
    tokenNames,
    globalAction,
    rules,
    numRules,
    bitsets,
    filterMode,
    labelType={<ASTLabelType>},
    ASTLabelType="CommonTree",

```

```

    superClass={<if(filterMode)><if(buildAST)>TreeRewriter<else>TreeFilter<endif><else>TreeParser<endif>},
    members={<actions.treeparser.members>}
) ::= <<
<genericParser(inputStreamType="id\<TreeNodeStream>", rewriteElementType="Node", ...)>
>>

/** How to generate a parser */
genericParser( grammar,
    name,
    scopes,
    tokens,
    tokenNames,
    rules,
    numRules,
    cyclicDFAs,    // parser init -- initializes the DFAs
    bitsets,
    labelType,
    ASTLabelType,
    superClass,
    members,
    filterMode,
    rewriteElementType,
    inputStreamType
) ::= <<
<cyclicDFAs:cyclicDFA(>

#pragma mark Bitsets
<bitsets:{it | <bitset(name={FOLLOW_<it.name>_in_<it.inName><it.tokenIndex>}, words64=it.bits)>>>

#pragma mark Dynamic Global globalAttributeScopeImplementation
<scopes:{it | <if(it.isDynamicGlobalScope)><globalAttributeScopeImplementation(scope=it)><endif>>>

#pragma mark Dynamic Rule Scopes ruleAttributeScopeImplementation
<rules:{rule |
<rule.ruleDescriptor:{ ruleDescriptor | <ruleAttributeScopeImplementation(scope=ruleDescriptor.ruleScope)>>>>

#pragma mark Rule Return Scopes returnScopeImplementation
<rules:{rule | <rule.ruleDescriptor:{ ruleDescriptor |
<returnScopeImplementation(scope=ruleDescriptor.returnScope)>>>>

@implementation <grammar.recognizerName> // line 637

/* ObjC start of ruleAttributeScope */
#pragma mark Dynamic Rule Scopes ruleAttributeScope
<rules:{rule | <rule.ruleDescriptor.ruleScope:ruleAttributeScope(>>>
/* ObjC end of ruleAttributeScope */
#pragma mark global Attribute Scopes globalAttributeScope
/* ObjC start globalAttributeScope */

```

```

<scopes:{it | <if(it.isDynamicGlobalScope)><globalAttributeScope()><endif>>
/* ObjC end globalAttributeScope */
/* ObjC start actions.(actionScope).synthesize */
<actions.(actionScope).synthesize>
/* ObjC start synthesize() */
<@synthesize()>

+ (void) initialize
{
    #pragma mark Bitsets
    <bitsets:{it | <bitsetInit(name={FOLLOW_<it.name>_in_<it.inName><it.tokenIndex>}, words64=it.bits)>>
    [BaseRecognizer setTokenNames:[[AMutableArray arrayWithObjects:@"\<invalid>", @"\<EOR>",
@"\<DOWN>", @"\<UP>", <tokenNames:{it | @<it>}; separator=", ", wrap="\n ", nil] retain]];
    [BaseRecognizer setGrammarFileName:@"<fileName>"];
    <synpreds:{pred | <synpred(pred)>>
}

+ (<grammar.recognizerName> *)new<grammar.recognizerName>:(<inputStreamType>)aStream
{
<if(PARSER)>
    return [[<grammar.recognizerName> alloc] initWithTokenStream:aStream];
<else><! TREE_PARSER !>
    return [[<grammar.recognizerName> alloc] initWithStream:aStream];
<endif>
}

<if(PARSER)>
- (id) initWithTokenStream:(<inputStreamType>)aStream
{
    self = [super initWithTokenStream:aStream State:[[RecognizerSharedState
newRecognizerSharedStateWithRuleLen:<numRules>+1] retain]];
    if ( self != nil ) {
<else><! TREE_PARSER !>
- (id) initWithStream:(<inputStreamType>)aStream
{
    self = [super initWithStream:aStream State:[[RecognizerSharedState
newRecognizerSharedStateWithRuleLen:<numRules>+1] retain]];
    if ( self != nil ) {
<endif>
        <! <parserCtorBody()> !>
        <cyclicDFAs:{dfa | dfa<dfa.decisionNumber> = [DFA<dfa.decisionNumber>
newDFA<dfa.decisionNumber>WithRecognizer:self];}; separator="\n">
        <scopes:{it | <if(it.isDynamicGlobalScope)><globalAttributeScopeInit(scope=it)><endif>>
        <rules:{rule | <rule.ruleDescriptor.ruleScope:ruleAttributeScopeInit()>>
        /* start of actions-actionScope-init */
        <actions.(actionScope).init>
        /* start of init */
        <@init()>

```

```

    }
    return self;
}

- (void) dealloc
{
    <cyclicDFAs:{ dfa | [dfa<dfa.decisionNumber> release];}; separator="\n">
    <scopes:{it | <if(it.isDynamicGlobalScope)><globalAttributeScopeDealloc(scope=it)><endif>}>
    <actions.(actionScope).dealloc>
    <@dealloc()>
    [super dealloc];
}

/* ObjC start actions.(actionScope).methods */
<actions.(actionScope).methods>
/* ObjC end actions.(actionScope).methods */
/* ObjC start methods() */
<@methods()>
/* ObjC end methods() */
/* ObjC start rules */
<rules; separator="\n">
/* ObjC end rules */

@end /* end of <grammar.recognizerName> implementation line 692 */<\n>
>>

parserCtorBody() ::= <<
<if(memoize)> /* parserCtorBody */
<if(grammar.grammarIsRoot)>
state.ruleMemo = [[RuleStack newRuleStack:<numRules>+1] retain];<\n> <! index from 1..n !>
<endif>
<endif>
<grammar.delegators:
{g|this.<g:delegateName()> = <g:delegateName()>;}; separator="\n">
>>

/** A simpler version of a rule template that is specific to the imaginary
 * rules created for syntactic predicates. As they never have return values
 * nor parameters etc..., just give simplest possible method. Don't do
 * any of the normal memoization stuff in here either; it's a waste.
 * As predicates cannot be inlined into the invoking rule, they need to
 * be in a rule by themselves.
 */
synpredRule(ruleName, ruleDescriptor, block, description, nakedBlock) ::=
<<
// $ANTLR start <ruleName>_fragment
- (void) <ruleName>_fragment
{

```

```

<ruleLabelDefs(>
<if(trace)>
    [self traceIn:\@"<ruleName>_fragment" Index:<ruleDescriptor.index>];
    @try {
        <block>
    }
    @finally {
        [self traceOut:\@"<ruleName>_fragment" Index:<ruleDescriptor.index>];
    }
<else>
    <block>
<endif>
} // $ANTLR end <ruleName>_fragment
>>

```

```

synpred(name) ::= <<
SEL <name>Selector = @selector(<name>_fragment);
<! // $ANTLR start <name>
- (BOOL) <name>
{
    state.backtracking++;
    <@start(>
    NSInteger start = [input mark];
    @try {
        [self <name>_fragment]; // can never throw exception
    }
    @catch (RecognitionException *re) {
        NSLog(@"impossible: %@\n", re.name);
    }
    BOOL success = (state.failed == NO);
    [input rewind:start];
    <@stop(>
    state.backtracking--;
    state.failed=NO;
    return success;
} // $ANTLR end <name> <\n> !>
>>

```

```

lexerSynpred(name) ::= <<
<synpred(name)>
>>

```

```

ruleMemoization(name) ::= <<
<if(memoize)>
if ( state.backtracking > 0 && [self alreadyParsedRule:input RuleIndex:<ruleDescriptor.index>] ) { return
<ruleReturnValue(>; }
<endif>
>>

```

```

/** How to test for failure and return from rule */
checkRuleBacktrackFailure() ::= <<
<if (backtracking)>if ( state.failed ) return <ruleReturnValue(>;<endif>
>>

/** This rule has failed, exit indicating failure during backtrack */
ruleBacktrackFailure() ::= <<
<if(backtracking)>if ( state.backtracking > 0 ) { state.failed = YES; return <ruleReturnValue(>; }<\n><endif>
>>

/** How to generate code for a rule.
* The return type aggregates are declared in the header file (headerFile template)
*/
rule(ruleName,ruleDescriptor,block,emptyRule,description,exceptions,finally,memoize) ::= <<

/*
* $ANTLR start <ruleName>
* <fileName>:<description>
*/
- (<returnType(>) <ruleName><ruleDescriptor.parameterScope:parameterScope(>)
{
  <if(trace)>[self traceIn:\@"<ruleName>" Index:<ruleDescriptor.index>];<endif>
  <if(trace)>NSLog(@"enter <ruleName> %@ failed=%@ backtracking=%d", [input LT:1],
(state.failed==YES)?@"YES":@"NO", state.backtracking);<endif>
  <ruleScopeSetUp(>
  <ruleDeclarations(>
  <ruleDescriptor.actions.init>
  <@preamble(>
  @try {
    <ruleMemoization(name=ruleName)>
    <ruleLabelDefs(>
    <block>
    <ruleCleanUp(>
    <(ruleDescriptor.actions.after):execAction(>
  }
  <if(exceptions)>
    <exceptions:{e|<catch(decl=e.decl,action=e.action)><\n>}>
  <else><if(!emptyRule)><if(actions.(actionScope).rulecatch)>
    <actions.(actionScope).rulecatch>
  <else>
    @catch (RecognitionException *re) {
      [self reportError:re];
      [self recover:input Exception:re];
      <@setErrorReturnValue(>
    }<\n>
  <endif><endif><endif>
  @finally {

```

```

    <if(trace)>[self traceOut:@"<ruleName>" Index:<ruleDescriptor.index>];<endif>
    <memoize()>
    <ruleScopeCleanUp()>
    <finally>
  }
  <@postamble()>
  return <ruleReturnValue()>;
}
/* $ANTLR end <ruleName> */
>>

finalCode(finalBlock) ::= <<
{
  <finalBlock>
}
>>

catch(decl,action) ::= <<
@catch (<e.decl>) {
  <e.action>
}
>>

ruleDeclarations() ::= <<
/* ruleDeclarations */
<if(ruleDescriptor.hasMultipleReturnValues)>
<returnType()> retval = [<ruleDescriptor:returnStructName()> new<ruleDescriptor:returnStructName()>];
[retval setStart:[input LT:1]];<n>
<else>
<ruleDescriptor.returnScope.attributes: { a |
<a.type> <a.name> = <if(a.initValue)><a.initValue><else><initValue(a.type)><endif>;
}>
<endif>
<if(memoize)>
NSInteger <ruleDescriptor.name>_startIndex = input.index;
<endif>
>>

ruleScopeSetUp() ::= <<
/* ruleScopeSetUp */
<ruleDescriptor.useScopes: { it | [<it>_stack push:[<it>_Scope new<it>_Scope];]}>
<ruleDescriptor.ruleScope: { it | [<it.name>_stack push:[<it.name>_Scope new<it.name>_Scope];]}>
>>

ruleScopeCleanUp() ::= <<
/* ruleScopeCleanUp */
<ruleDescriptor.useScopes: { it | [<it>_stack pop];}; separator="\n">
<ruleDescriptor.ruleScope: { it | [<it.name>_stack pop];}; separator="\n">

```

>>

```
ruleLabelDefs() ::= <%
/* ruleLabelDefs entry */<"\n">
<[ruleDescriptor.tokenLabels, ruleDescriptor.tokenListLabels,
ruleDescriptor.wildcardTreeLabels,ruleDescriptor.wildcardTreeListLabels]
: {it |<labelType> *<it.label.text> = nil;}; separator="\n"><"\n">
<[ruleDescriptor.tokenListLabels,ruleDescriptor.ruleListLabels,ruleDescriptor.wildcardTreeListLabels]
: {it | NSMutableArray *list_<it.label.text> = nil;}; separator="\n"><"\n">
<ruleDescriptor.ruleLabels:ruleLabelDef(); separator="\n"><"\n">
<ruleDescriptor.ruleListLabels:{|l|ParserRuleReturnScope *<l.label.text> = nil;}; separator="\n"><"\n">
%>
```

```
lexerRuleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,
ruleDescriptor.tokenListLabels,
ruleDescriptor.ruleLabels]
: {it |<labelType> *<it.label.text>=nil;}; separator="\n"
>
<ruleDescriptor.charLabels:{it |NSInteger <it.label.text>;}; separator="\n">
<[ruleDescriptor.tokenListLabels,
ruleDescriptor.ruleListLabels]:{it |NSMutableArray *list_<it.label.text>=nil; }; separator="\n">
>>
```

```
ruleReturnValue() ::= <%
<if(!ruleDescriptor.isSynPred)>
<if(ruleDescriptor.hasReturnValue)>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnName>
<else>
retval
<endif>
<endif>
<endif>
%>
```

```
ruleCleanup() ::= <<
/* token+rule list labels */
<[ruleDescriptor.tokenListLabels,ruleDescriptor.ruleListLabels]:{it |[list_<it.label.text> release];}; separator="\n">
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(!TREE_PARSER)>
[retval setStop:[input LT:-1]];<\n>
<endif><endif>
>>
```

```
memoize() ::= <<
<if(memoize)>
<if(backtracking)>
```

```

if (state.backtracking > 0) [self memoize:input RuleIndex:<ruleDescriptor.index>
StartIndex:<ruleDescriptor.name>_StartIndex];
<endif><endif>
>>

/** How to generate a rule in the lexer; naked blocks are used for
 * fragment rules.
 */
lexerRule(ruleName, nakedBlock, ruleDescriptor, block, memoize) ::= <<
// $ANTLR start "<ruleName>"
- (void)
m<ruleName><if(ruleDescriptor.parameterScope)><ruleDescriptor.parameterScope:parameterScope(scope=it)><en
dif>
{
//<if(trace)>[self traceIn:\@"<ruleName>" Index:<ruleDescriptor.index>];<endif>
<if(trace)>NSLog(@"enter <ruleName> %C line=%d:%d failed=%@ backtracking=%d",
[input LA:1],
self.line,
self.charPositionInLine,
(state.failed==YES) ? @"YES" : @"NO",
state.backtracking);
<endif>
<ruleScopeSetUp()>
<ruleDeclarations()>
@try {
<if(nakedBlock)>
<ruleMemoization(name=ruleName)>
<lexerRuleLabelDefs()>
<ruleDescriptor.actions.init>
<block><\n>
<else>
NSInteger _type = <ruleName>;
NSInteger _channel = TokenChannelDefault;
<ruleMemoization(name=ruleName)>
<lexerRuleLabelDefs()>
<ruleDescriptor.actions.init>
<block>
<ruleCleanUp()>
state.type = _type;
state.channel = _channel;
<(ruleDescriptor.actions.after):execAction()>
<endif>
}
@finally {
//<if(trace)>[self traceOut:[NSString stringWithFormat:@"<ruleName> %d\n",
<ruleDescriptor.index>]];<endif>
<if(trace)>NSLog(@"exit <ruleName> %C line=%d:%d failed=%@ backtracking=%d",
[input LA:1], self.line, self.charPositionInLine,

```

```

        (state.failed==YES) ? @"YES" : @"NO", state.backtracking);<endif>
    <ruleScopeCleanUp()>
    <memoize()>
    }
    return;
}
/* $ANTLR end "<ruleName>" */
>>

/** How to generate code for the implicitly-defined lexer grammar rule
 * that chooses between lexer rules.
 */
tokensRule(ruleName,nakedBlock,args,block,ruleDescriptor) ::= <<
- (void) mTokens
{
    <block><\n>
}
>>

// S U B R U L E S

/** A (...) subrule with multiple alternatives */
block(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
// <fileName>:<description> // block
NSInteger alt<decisionNumber>=<maxAlt>;
<decls>
<@predecision()>
<decision>
<@postdecision()>
<@prebranch()>
switch (alt<decisionNumber>) {
    <alts:{ a | <altSwitchCase(i, a)>}>
}
<@postbranch()>
>>

/** A rule block with multiple alternatives */
ruleBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
// <fileName>:<description> //ruleblock
NSInteger alt<decisionNumber>=<maxAlt>;
<decls>
<@predecision()>
<decision>
<@postdecision()>
switch (alt<decisionNumber>) {
    <alts:{ a | <altSwitchCase(i, a)>}>
}
>>

```

```

ruleBlockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
// <fileName>:<description> // ruleBlockSingleAlt
<decls>
<@prealt()>
<alts>
<@postalt()>
>>

/** A special case of a (...) subrule with a single alternative */
blockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
// <fileName>:<description> // blockSingleAlt
<decls>
<@prealt()>
<alts>
<@postalt()>
>>

/** A (..)+ block with 1 or more alternatives */
positiveClosureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
// <fileName>:<description> // positiveClosureBlock
NSInteger cnt<decisionNumber> = 0;
<decls>
<@preloop()>
do {
    NSInteger alt<decisionNumber> = <maxAlt>;
    <@predecision()>
    <decision>
    <@postdecision()>
    switch (alt<decisionNumber>) {
        <alts:{a | <altSwitchCase(i, a)>}>
        default :
            if ( cnt<decisionNumber> >= 1 )
                goto loop<decisionNumber>;
            <ruleBacktrackFailure()>
            EarlyExitException *eee =
                [EarlyExitException newException:input decisionNumber:<decisionNumber>];
            <@earlyExitException()>
            @throw eee;
        }
    cnt<decisionNumber>++;
} while (YES);
loop<decisionNumber>; ;
<@postloop()>
>>

positiveClosureBlockSingleAlt ::= positiveClosureBlock

```

```

/** A (..)* block with 0 or more alternatives */
closureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::=
<<
<decls>
<@preloop()>
do {
  NSInteger alt<decisionNumber>=<maxAlt>;
  <@predecision()>
  <decision>
  <@postdecision()>
  switch (alt<decisionNumber>) {
    <alts:{ a | <altSwitchCase(i, a)> }>
    default :
      goto loop<decisionNumber>;
  }
} while (YES);
loop<decisionNumber>; ;
<@postloop()>
>>

```

```
closureBlockSingleAlt ::= closureBlock
```

```

/** Optional blocks (x)? are translated to (x|) by before code generation
* so we can just use the normal block template
*/

```

```
optionalBlock ::= block
```

```
optionalBlockSingleAlt ::= block
```

```

/** A case in a switch that jumps to an alternative given the alternative
* number. A DFA predicts the alternative and then a simple switch
* does the jump to the code that actually matches that alternative.
*/

```

```

altSwitchCase(altNum, alt) ::= <<
case <altNum> : ;
  <@prealt()>
  <alt>
  break;<\n>
>>

```

```

/** An alternative is just a list of elements; at outermost level */
alt(elements,altNum,description,autoAST,outerAlt,treeLevel,rew) ::= <<
// <fileName>:<description> // alt
{
  <@declarations()>
  <elements:element()>
  <rew>
}

```

```

<@cleanup(>
}
>>

/** What to emit when there is no rewrite. For auto build
 * mode, does nothing.
 */
noRewrite(rewriteBlockLevel, treeLevel) ::= ""

// E L E M E N T S

/** Dump the elements one per line */
element(e) ::= << <@prematch(><\n><e.el><\n> >>

/** match a token optionally with a label in front */
tokenRef(token,label,elementIndex,terminalOptions) ::= <<
<if(label)><label>=(<labelType> *)<endif>[self match:input TokenType:<token>
Follow:FOLLOW_<token>_in_<ruleName><elementIndex>]; <checkRuleBacktrackFailure(>
>>

/** ids+=ID */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

listLabel(label,elem) ::= <<
if (list_<label> == nil) list_<label> = [[AMutableArray arrayWithCapacity:5] retain];
[list_<label> addObject:<elem>];<\n>
>>

/** match a character */
charRef(char,label) ::= <<
<if(label)>NSInteger <label> = [input LA:1];<\n><endif>
[self matchChar:<char>]; <checkRuleBacktrackFailure(><\n>
>>

/** match a character range */
charRangeRef(a,b,label) ::= <<
<if(label)><label> = [input LA:1];<\n><endif>
[self matchRangeFromChar:<a> to:<b>]; <checkRuleBacktrackFailure(>
>>

/** For now, sets are interval tests and must be tested inline */
matchSet(s,label,elementIndex,terminalOptions,postmatchCode="") ::= <<
<if(label)>
<if(LEXER)>
<label> = [input LA:1];<\n>

```

```

<else>
<label> = (<labelType> *)[input LT:1]; /* matchSet */<\n>
<endif><endif>
if (<s>) {
    [input consume];
    <postmatchCode>
<if(!LEXER)>
    [state setIsErrorRecovery:NO];
<endif>
    <if(backtracking)>state.failed = NO;<\n><endif>
} else {
    <ruleBacktrackFailure()>
    MismatchedSetException *mse = [MismatchedSetException newException:nil stream:input];
    <@mismatchedSetException()>
<if(LEXER)>
<if(label)>
    mse.c = <label>;
<endif>
    [self recover:mse];
    @throw mse;
<else>
    @throw mse;
    <! use following code to make it recover inline; remove throw mse;
    [self recoverFromMismatchedSet:input exception:mse follow:FOLLOW_set_in_<ruleName><elementIndex>]; !>
<endif>
}<\n>
>>

```

```

matchRuleBlockSet ::= matchSet

```

```

matchSetAndListLabel(s,label,elementIndex,postmatchCode) ::= <<
<matchSet(...)>
<listLabel(elem=label,...)>
>>

```

```

/** Match a string literal */
lexerStringRef(string,label,elementIndex="0") ::= <<
<if(label)>
    NSInteger <label>Start = input.index;
    [self matchString:<string>]; <checkRuleBacktrackFailure()>
    NSInteger StartLine<elementIndex> = self.line;
    NSInteger <label>StartCharPos<elementIndex> = self.charPositionInLine;
    <label> = [[<labelType> newToken:input Type:TokenTypeInvalid Channel:TokenChannelDefault
    Start:<label>Start Stop:input.index] retain];
    [self setLine:<label>StartLine<elementIndex>];
    [self setCharPositionInLine:<label>StartCharPos<elementIndex>];
<else>
    [self matchString:<string>]; <checkRuleBacktrackFailure()><\n>

```

```

<endif>
>>

wildcard(token,label,elementIndex,terminalOptions) ::= <<
<if(label)>
<label> = (<labelType> *)[input LT:1];<\n>
<endif>
[self matchAny:input]; <checkRuleBacktrackFailure()>
>>

wildcardAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<wildcard(...)>
<listLabel(elem=label,...)>
>>

/** Match . wildcard in lexer */
wildcardChar(label, elementIndex) ::= <<
<if(label)>
NSInteger <label> = [input LA:1];<\n>
<endif>
[self matchAny]; <checkRuleBacktrackFailure()><\n>
>>

wildcardCharListLabel(label, elementIndex) ::= <<
<wildcardChar(...)>
<listLabel(elem=label,...)>
>>

/** Match a rule reference by invoking it possibly with arguments
 * and a return value or values. The 'rule' argument was the
 * target rule name, but now is type Rule, whose toString is
 * same: the rule name. Now though you can access full rule
 * descriptor stuff.
 */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
/* ruleRef */
[self pushFollow:FOLLOW_<rule.name>_in_<ruleName><elementIndex>];
<if(label)><label> = <endif>[self <if(scope)><scope.delegateName()>.<endif><rule.name><if(args)>:<first(args)>
<rest(args):{ a | arg<i>:<rest(args)>}; separator=" "><endif>];<\n>
[self popFollow];
<checkRuleBacktrackFailure()><\n>
>>

/** ids+=r */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(elem=label,...)>
>>

```

```

/** A lexer rule reference.
 *
 * The 'rule' argument was the target rule name, but now
 * is type Rule, whose toString is same: the rule name.
 * Now though you can access full rule descriptor stuff.
 */
lexerRuleRef(rule,label,args,elementIndex,scope) ::= <<
<if(label)>
NSInteger <label>Start<elementIndex> = input.index;
[self m<rule.name><if(args)>:<args; separator=":"><endif>]; <checkRuleBacktrackFailure()><\n>
<label> = [[<labelType> newToken:input Type:TokenTypeInvalid Channel:TokenChannelDefault
Start:<label>Start<elementIndex> Stop:input.index-1] retain];
<label>.line = self.line;
<else>
[self <if(scope)><scope:delegateName()>.<endif>m<rule.name><if(args)>:<args; separator=":"><endif>];
<checkRuleBacktrackFailure()><\n>
<endif>
>>

/** i+=INT in lexer */
lexerRuleRefAndListLabel(rule,label,args,elementIndex,scope) ::= <<
<lexerRuleRef(...)>
<listLabel(elem=label,...)>
>>

/** EOF in the lexer */
lexerMatchEOF(label,elementIndex) ::= <<
<if(label)>
NSInteger <label>Start<elementIndex> = input.index;
[self matchChar:CharStreamEOF]; <checkRuleBacktrackFailure()><\n>
<labelType> <label> = [[<labelType> newToken:input Type:TokenTypeEOF Channel:TokenChannelDefault
Start:<label>Start<elementIndex> Stop:input.index-1] retain];
<label>.line = self.line;
<else>
[self matchChar:CharStreamEOF]; <checkRuleBacktrackFailure()><\n>
<endif>
>>

// used for left-recursive rules
recRuleDefArg()          ::= "int <recRuleArg()>"
recRuleArg()            ::= "_p"
recRuleAltPredicate(ruleName,opPrec) ::= "<recRuleArg()> \<= <opPrec>"
recRuleSetResultAction() ::= "root_0=$<ruleName>_primary.tree;"
recRuleSetReturnAction(src,name)  ::= "$<name>=$<src>.<name>;"

/** match ^(root children) in tree parser */
tree(root, actionsAfterRoot, children, nullableChildList, enclosingTreeLevel, treeLevel) ::= <<

```

```

<root:element()>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if ( [input LA:1] == DOWN ) {
  [self match:input TokenType:DOWN Follow:nil]; <checkRuleBacktrackFailure()>
  <children:element()>
  [self match:input TokenType:UP Follow:nil]; <checkRuleBacktrackFailure()>
}
<else>
  [self match:input TokenType:DOWN Follow:nil]; <checkRuleBacktrackFailure()>
  <children:element()>
  [self match:input TokenType:UP Follow:nil]; <checkRuleBacktrackFailure()>
<endif>
>>

/** Every predicate is used as a validating predicate (even when it is
 * also hoisted into a prediction expression).
 */
validateSemanticPredicate(pred,description) ::= <<
if ( !( <evalPredicate(...)> ) ) {
  <ruleBacktrackFailure()>
  @throw [FailedPredicateException newException:@ "<ruleName>" predicate:@ "<description>" stream:input];
}
>>

// F i x e d D F A (if-then-else)

dfaState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
NSInteger LA<decisionNumber>_<stateNumber> = [input LA:<k>];<n>
<edges; separator="\nelse ">
else {
<if(eotPredictsAlt)>
  alt<decisionNumber> = <eotPredictsAlt>;
<else>
  <ruleBacktrackFailure()>
  NoViableAltException *nvae = [NoViableAltException newException:<decisionNumber> state:<stateNumber>
stream:input];
  nvae.c = LA<decisionNumber>_<stateNumber>;
  <@noViableAltException()>
  @throw nvae;<n>
<endif>
}
>>

/** Same as a normal DFA state except that we don't examine lookahead
 * for the bypass alternative. It delays error detection but this
 * is faster, smaller, and more what people expect. For (X)? people
 * expect "if ( LA(1)==X ) match(X);" and that's it.

```

```

*/
dfaOptionalBlockState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
NSInteger LA<decisionNumber>_<stateNumber> = [input LA:<k>];<n>
<edges; separator="\nelse ">
>>

/** A DFA state that is actually the loopback decision of a closure
* loop. If end-of-token (EOT) predicts any of the targets then it
* should act like a default clause (i.e., no error can be generated).
* This is used only in the lexer so that for ('a'* on the end of a rule
* anything other than 'a' predicts exiting.
*/
*/
dfaLoopbackState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
NSInteger LA<decisionNumber>_<stateNumber> = [input LA:<k>];
<edges; separator="\nelse "><n>
<if(eotPredictsAlt)>
<if(!edges)>
alt<decisionNumber>=<eotPredictsAlt>; <! if no edges, don't gen ELSE !>
<else>
else {
    alt<decisionNumber> = <eotPredictsAlt>;
}<n>
<endif><endif>
>>

/** An accept state indicates a unique alternative has been predicted */
dfaAcceptState(alt) ::= "alt<decisionNumber>=<alt>";

/** A simple edge with an expression. If the expression is satisfied,
* enter to the target state. To handle gated productions, we may
* have to evaluate some predicates for this edge.
*/
*/
dfaEdge(labelExpr, targetState, predicates) ::= <<
if ( (<labelExpr> <if(predicates)>&& (<predicates>)<endif>) {
    <targetState>
}
>>

// F i x e d D F A (switch case)

/** A DFA state where a SWITCH may be generated. The code generator
* decides if this is possible: CodeGenerator.canGenerateSwitch().
*/
*/
dfaStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
unichar charLA<decisionNumber> = [input LA:<k>];
switch (charLA<decisionNumber>) {
    <edges; separator="\n"><n>
default: ;

```

```

<if(eotPredictsAlt)>
  alt<decisionNumber> = <eotPredictsAlt>;
<else>
  <ruleBacktrackFailure()>
  NoViableAltException *nvae = [NoViableAltException newException:<decisionNumber> state:<stateNumber>
stream:input];
  nvae.c = charLA<decisionNumber>;
  <@noViableAltException()>
  @throw nvae;<\n>
<endif>
}<\n>
>>

```

```

dfaOptionalBlockStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ([input LA:<k>]) { // dfaOptionalBlockStateSwitch
  <edges; separator="\n"><\n>
}<\n>
>>

```

```

dfaLoopbackStateSwitch(k, edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ([input LA:<k>]) { // dfaLoopbackStateSwitch
  <edges; separator="\n"><\n>
<if(eotPredictsAlt)>
default:
  alt<decisionNumber> = <eotPredictsAlt>;
  break;<\n>
<endif>
}<\n>
>>

```

```

dfaEdgeSwitch(labels, targetState) ::= <<
<labels:{it | case <it>: ;}; separator="\n">
{
  <targetState>
}
break;
>>

```

// C y c l i c D F A

```

/** The code to initiate execution of a cyclic DFA; this is used
 * in the rule to predict an alt just like the fixed DFA case.
 * The <name> attribute is inherited via the parser, lexer, ...
 */
dfaDecision(decisionNumber,description) ::= <<
alt<decisionNumber> = [dfa<decisionNumber> predict:input];
>>

```

```

/** Used in headerFile */
cyclicDFAInterface(dfa) ::= <<
#pragma mark Cyclic DFA interface start DFA<dfa.decisionNumber>
@interface DFA<dfa.decisionNumber> : DFA {
}
+ (DFA<dfa.decisionNumber> *) newDFA<dfa.decisionNumber>WithRecognizer:(BaseRecognizer
*)theRecognizer;
- initWithRecognizer:(BaseRecognizer *)recognizer;
@end /* end of DFA<dfa.decisionNumber> interface */<\n>
#pragma mark Cyclic DFA interface end DFA<dfa.decisionNumber><\n>
>>

```

```

/** Used in lexer/parser implementation files */
/* Dump DFA tables as run-length-encoded Strings of octal values.
* Can't use hex as compiler translates them before compilation.
* These strings are split into multiple, concatenated strings.
* Java puts them back together at compile time thankfully.
* Java cannot handle large static arrays, so we're stuck with this
* encode/decode approach. See analysis and runtime DFA for
* the encoding methods.
*/

```

```

cyclicDFA(dfa) ::= <<
#pragma mark Cyclic DFA implementation start DFA<dfa.decisionNumber>

@implementation DFA<dfa.decisionNumber>
const static NSInteger dfa<dfa.decisionNumber>_eot[<dfa.numberofStates>] =
    {<dfa.eot; wrap="\n    ", separator=",", null="-1">};
const static NSInteger dfa<dfa.decisionNumber>_eof[<dfa.numberofStates>] =
    {<dfa.eof; wrap="\n    ", separator=",", null="-1">};
const static unichar dfa<dfa.decisionNumber>_min[<dfa.numberofStates>] =
    {<dfa.min; wrap="\n    ", separator=",", null="-1">};
const static unichar dfa<dfa.decisionNumber>_max[<dfa.numberofStates>] =
    {<dfa.max; wrap="\n    ", separator=",", null="-1">};
const static NSInteger dfa<dfa.decisionNumber>_accept[<dfa.numberofStates>] =
    {<dfa.accept; wrap="\n    ", separator=",", null="-1">};
const static NSInteger dfa<dfa.decisionNumber>_special[<dfa.numberofStates>] =
    {<dfa.special; wrap="\n    ", separator=",", null="-1">};

```

```

/** Used when there is no transition table entry for a particular state */
#define dfa<dfa.decisionNumber>_T_empty    nil

```

```

<dfa.edgeTransitionClassMap.keys:{ table |
const static NSInteger dfa<dfa.decisionNumber>_T<i0>[] =
{
    <table; separator=", ", wrap="\n ", null="-1">
\};
}; null="">

```

```

const static NSInteger *dfa<dfa.decisionNumber>_transition[] =
{
    <dfa.transitionEdgeTables:{ whichTable|dfa<dfa.decisionNumber>_T<whichTable>}; separator=", ", wrap="\n",
    null="nil">
};

+ (DFA<dfa.decisionNumber> *) newDFA<dfa.decisionNumber>WithRecognizer:(BaseRecognizer *)aRecognizer
{
    return [[[DFA<dfa.decisionNumber> alloc] initWithRecognizer:aRecognizer] retain];
}

- (id) initWithRecognizer:(BaseRecognizer *) theRecognizer
{
    self = [super initWithRecognizer:theRecognizer];
    if ( self != nil ) {
        decisionNumber = <dfa.decisionNumber>;
        eot = dfa<dfa.decisionNumber>_eot;
        eof = dfa<dfa.decisionNumber>_eof;
        min = dfa<dfa.decisionNumber>_min;
        max = dfa<dfa.decisionNumber>_max;
        accept = dfa<dfa.decisionNumber>_accept;
        special = dfa<dfa.decisionNumber>_special;
        transition = dfa<dfa.decisionNumber>_transition;
    }
    return self;
}

<if(dfa.specialStateSTs)>
/* start dfa.specialStateSTs */
- (NSInteger) specialStateTransition:(NSInteger)s Stream:(id<IntStream\>)anInput
{
    <if(LEXER)>
        id<IntStream\> input = anInput;<\n>
    <endif>
    <if(PARSER)>
        id<TokenStream\> input = (id<TokenStream\>)anInput;<\n>
    <endif>
    <if(TREE_PARSER)>
        id<TreeNodeStream\> input = (id<TreeNodeStream\>)anInput;<\n>
    <endif>
    switch (s) {
        <dfa.specialStateSTs:{ state |
            case <i0> : ;<! compressed special state numbers 0..n-1 !>
                <state>}; separator="\n">
    }
    <if(backtracking)>
        if ( [recognizer getBacktrackingLevel] > 0 ) { [recognizer setFailed:YES]; return -1; }<\n>
    <endif>

```

```

    NoViableAltException *nvae = [NoViableAltException newException:<dfa.decisionNumber> state:s
stream:recognizer.input];
    // nvae.c = s;
    /* [self error:nvae]; */ <! for debugger - do later !>
    @throw nvae;
}<\n>
/* end dfa.specialStateSTs */
<endif>

- (void) dealloc
{
    //free(transition);
    [super dealloc];
}

- (NSString *) description
{
    return @"<dfa.description>";
}

<@errorMethod()>

@end /* end DFA<dfa.decisionNumber> implementation */<\n>
#pragma mark Cyclic DFA implementation end DFA<dfa.decisionNumber>
<\n>
>>
/** A state in a cyclic DFA; it's a special state and part of a big switch on
 * state.
 */
cyclicDFASState(decisionNumber, stateNumber, edges, needErrorClause, semPredState) ::= <<
/* cyclicDFASState */
NSInteger LA<decisionNumber>_<stateNumber> = [input LA:1];<\n>
<if(semPredState)> <! get next lookahead symbol to test edges, then rewind !>
NSInteger index<decisionNumber>_<stateNumber> = input.index;
[input rewind];<\n>
<endif>
s = -1;
<edges; separator="\nelse ">
<if(semPredState)> <! return input cursor to state before we rewound !>
[input seek:index<decisionNumber>_<stateNumber>];<\n>
<endif>
if ( s >= 0 )
    return s;
break;
>>

/** Just like a fixed DFA edge, test the lookahead and indicate what
 * state to jump to next if successful.

```

```

*/
cyclicDFAEdge(labelExpr, targetStateNumber, edgeNumber, predicates) ::= <<
/* cyclicDFAEdge */
if (<labelExpr><if(predicates)> && (<predicates>)<endif>) { s = <targetStateNumber>;}<\n>
>>

/** An edge pointing at end-of-token; essentially matches any char;
* always jump to the target.
*/
eotDFAEdge(targetStateNumber,edgeNumber, predicates) ::= <<
s = <targetStateNumber>;<\n> /* eotDFAEdge */
>>

// D F A E X P R E S S I O N S

andPredicates(left,right) ::= "<left>&&<right>"

orPredicates(operands) ::= "<operands; separator='||\>"

notPredicate(pred) ::= "!(<evalPredicate(pred, {})>)"

evalPredicate(pred,description) ::= "<pred>"

/*
* evalSynPredicate(pred,description) ::= "<pred>()"
*
* synpreds are broken in cyclic DFA special states
* Damn! For now, work around with using the selectors directly, and by providing a trampoline evalSynPred
method in
* DFA
*/
/* evalSynPredicate(pred,description) ::= "[self evaluateSyntacticPredicate:<pred>Selector stream:input]" */
evalSynPredicate(pred,description) ::= "[self evaluateSyntacticPredicate:@selector(<pred>_fragment)]"
/* evalSynPredicate(pred,description) ::= "[recognizer <pred>]" */

lookaheadTest(atom,k,atomAsInt) ::= "LA<decisionNumber>_<stateNumber>===<atom>"

/** Sometimes a lookahead test cannot assume that LA(k) is in a temp variable
* somewhere. Must ask for the lookahead directly.
*/
isolatedLookaheadTest(atom,k,atomAsInt) ::= "[input LA:<k>] == <atom>"

lookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= <%
(LA<decisionNumber>_<stateNumber> >= <lower> && LA<decisionNumber>_<stateNumber> \<= <upper>)
%>

isolatedLookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= "(((input LA:<k>] >=
<lower>) && ([input LA:<k>] \<= <upper>)))"

```

```

setTest(ranges) ::= <%
<ranges; separator="||">
%>

// A T T R I B U T E S

memVars(scope) ::= <% <scope.attributes:{a|<a.type> <a.name>;<\n>}; separator="\n"> %>

properties(scope) ::= <%
<scope.attributes:{a|@property (assign, getter=get<a.name>, setter=set<a.name>:) <a.type> <a.name>;<\n>};
separator="\n">
%>

methodsDecl(scope) ::= <%
<scope.attributes:{a|- (<a.type>)get<a.name>;<\n>- (void)set<a.name>:(<a.type>)aVal;<\n>}; separator="\n">
%>

synthesize(scope) ::= <% <scope.attributes:{a|@synthesize <a.name>;}; separator="\n"> %>

methods(scope) ::= <%
<scope.attributes:{a|
- (<a.type>)get<a.name> { return( <a.name> ); }\<\n>
- (void)set<a.name>:(<a.type>)aVal { <a.name> = aVal; }\<\n>}; separator="\n">
%>

globalAttributeScopeInterface(scope) ::= <%
/* globalAttributeScopeInterface */<\n>
@interface <scope.name>_Scope : SymbolsScope {<\n>
<if(scope.attributes)>
<memVars(scope)>
<endif>
}<\n>
<if(scope.attributes)>
/* start of globalAttributeScopeInterface properties */<\n>
<properties(scope)>
/* end globalAttributeScopeInterface properties */<\n>
<endif>

+ (<scope.name>_Scope *)new<scope.name>_Scope;<\n>
- (id) init;<\n>
<if(scope.attributes)>
/* start of globalAttributeScopeInterface methodsDecl */<\n>
<methodsDecl(scope)>
/* End of globalAttributeScopeInterface methodsDecl */<\n>
<endif>
@end /* end of <scope.name>_Scope interface */<\n>
%>

```

```

globalAttributeScopeMemVar(scope) ::= <%
/* globalAttributeScopeMemVar */<\n>
SymbolStack *<scope.name>_stack;<\n>
<scope.name>_Scope *<scope.name>_scope;<\n>
%>

globalAttributeScopeImplementation(scope) ::= <%
@implementation <scope.name>_Scope /* globalAttributeScopeImplementation */<\n>
<if(scope.attributes)>
/* start of synthesize -- OBJC-Line 1750 */<\n>
<synthesize(scope)><\n>
<endif>
<\n>
+ (<scope.name>_Scope *)new<scope.name>_Scope<\n>
{<\n>
    return [[<scope.name>_Scope alloc] init];<\n>
}<\n>
<\n>
- (id) init<\n>
{<\n>
    self = [super init];<\n>
    return self;<\n>
}<\n>
<\n>
<if(scope.attributes)>
/* start of iterate get and set functions */<\n>
<methods(scope)><\n>
/* End of iterate get and set functions */<\n>
<endif>
@end /* end of <scope.name>_Scope implementation */<\n><\n>
%>

globalAttributeScopeInit(scope) ::= <<
/* globalAttributeScopeInit */<\n>
<scope.name>_scope = [<scope.name>_Scope new<scope.name>_Scope];<\n>
<scope.name>_stack = [SymbolStack newSymbolStackWithLen:30];<\n>
>>

globalAttributeScopeDealloc(scope) ::= <% [<scope.name>_stack release];<\n> %>

globalAttributeScope(scope) ::= <%
<if(scope.name)>
static <scope.name>_stack;<\n>
<endif>
%>

ruleAttributeScopeMemVar(scope) ::= <%

```

```

/* ObjC ruleAttributeScopeMemVar */<\n>
<if(scope.attributes)>
<scope.name>_Scope *<scope.name>_scope; /* ObjC ruleAttributeScopeMemVar */<\n>
<endif>
%>

ruleAttributeScopeInterface(scope) ::= <%
<if(scope.attributes)>
/* start of ruleAttributeScopeInterface */<\n>
@interface <scope.name>_Scope : SymbolsScope {<\n>
    <memVars(scope)><\n>
}<\n>
<\n>
/* start property declarations */<\n>
<properties(scope)><\n>
/* start method declarations */<\n>
+ (<scope.name>_Scope *)new<scope.name>_Scope;<\n>
- (id) init;<\n>
<methodsDecl(scope)><\n>
@end /* end of ruleAttributeScopeInterface */<\n><\n>
<endif>
%>

ruleAttributeScopeImplementation(scope) ::= <%
<if(scope.attributes)>
@implementation <scope.name>_Scope /* start of ruleAttributeScopeImplementation */<\n>
<synthesize(scope)><\n>
<\n>
+ (<scope.name>_Scope *)new<scope.name>_Scope<\n>
{<\n>
    return [[<scope.name>_Scope alloc] init];<\n>
}<\n>
<\n>
- (id) init<\n>
{<\n>
    self = [super init];<\n>
    return self;<\n>
}<\n>
<\n>
/* start of <scope.name>_Scope get and set functions */<\n>
<methods(scope)><\n>
/* End of <scope.name>_Scope get and set functions */<\n>
@end /* end of ruleAttributeScopeImplementation */<\n><\n>
<endif>
%>

ruleAttributeScopeInit(scope) ::= <%
/* ruleAttributeScopeInit */<\n>

```

```

<scope.name>_scope = [<scope.name>_Scope new<scope.name>_Scope];<\n>
<scope.name>_stack = [SymbolStack newSymbolStackWithLen:30];<\n>
%>

ruleAttributeScopeDealloc(scope) ::= <% [<scope.name>_Scope release];<\n> %>

ruleAttributeScope(scope) ::= <%
<if(scope.attributes)>
/* ruleAttributeScope */<\n>
static SymbolStack *<scope.name>_stack;<\n>
static <scope.name>_Scope *<scope.name>_scope;
<endif>
%>

ruleAttributeScopeDecl(scope) ::= <%
/* ruleAttributeScopeDecl */<\n>
<if(scope.attributes)>
<scope.name>_Scope *<scope.name>_scope;<\n>
<endif>
%>

returnStructName(r) ::= "<className()>_<r.name>_return"

returnType() ::= <%
<if(!ruleDescriptor.isSynPred)>
<if(ruleDescriptor.hasMultipleReturnValues)>
<ruleDescriptor:returnStructName()> *
<else>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnType>
<else>
void
<endif>
<endif>
<else>
void
<endif>
%>

/** Generate the Objective-C type associated with a single or multiple return
 * values.
 */
ruleLabelType(referencedRule) ::= <%
<if(referencedRule.hasMultipleReturnValues)>
<className()>_<referencedRule.name>_return *<else>
<if(referencedRule.hasSingleReturnValue)><referencedRule.singleValueReturnType><else>
void<endif>
<endif>

```

```

%>

delegateName(d) ::= <% <if(d.label)><d.label><else>g<d.name><endif> %>

/** Using a type to init value map, try to init a type; if not in table
 * must be an object, default value is "null".
 */
initValue(typeName) ::= <% <objcTypeInitMap.(typeName)> %>

/** Define a rule label including default value */
ruleLabelDef(label) ::= <%
<ruleLabelType(referencedRule=label.referencedRule)> <label.label.text> =
<initValue(typeName=ruleLabelType(referencedRule=label.referencedRule))>;<\n>
%>

/** Define a return struct for a rule if the code needs to access its
 * start/stop tokens, tree stuff, attributes, ... Leave a hole for
 * subgroups to stick in members.
 */
returnScopeInterface(scope) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
/* returnScopeInterface <ruleDescriptor:returnStructName()> */
@interface <ruleDescriptor:returnStructName()> :
<if(TREE_PARSER)>Tree<else>Parser<endif>RuleReturnScope { /* returnScopeInterface line 1838 */
<@memVars()> /* ObjC start of memVars() */<\n>
<if(scope.attributes)>
<memVars(scope)><\n>
<endif>
}
/* start property declarations */
<@properties()><\n>
<if(scope.attributes)>
<properties(scope)><\n>
<endif>
/* start of method declarations */<\n>
+ (<ruleDescriptor:returnStructName()> *)new<ruleDescriptor:returnStructName()>;
/* this is start of set and get methods */
<@methodsDecl()> /* methodsDecl */<\n>
<if(scope.attributes)>
/* start of iterated get and set functions */<\n>
<methodsDecl(scope)><\n>
<endif>
@end /* end of returnScopeInterface interface */<\n>
<endif>
>>

returnScopeImplementation(scope) ::= <%
<if(ruleDescriptor.hasMultipleReturnValues)>

```

```

@implementation <ruleDescriptor:returnStructName()> /* returnScopeImplementation */<\n>
<@synthesize()> /* start of synthesize -- OBJC-Line 1837 */<\n>
<if(scope.attributes)>
  <synthesize(scope)><\n>
<endif>
+ (<ruleDescriptor:returnStructName()> *)new<ruleDescriptor:returnStructName()><\n>
{<\n>
  return [[[<ruleDescriptor:returnStructName()> alloc] init] retain];<\n>
}<\n>
<\n>
- (id) init<\n>
{<\n>
  self = [super init];<\n>
  return self;<\n>
}<\n>
<\n>
<@methods()><\n>
<if(scope.attributes)>
/* start of iterate get and set functions */<\n>
<methods(scope)><\n>
/* End of iterate get and set functions */<\n>
<endif>
<actions.(actionScope).ruleReturnMethods>
<@ruleReturnMembers()><\n>
@end /* end of returnScope implementation */<\n><\n>
<endif>
%>

parameterScope(scope) ::= <<
<! <scope.attributes:{ it | :(<it.type><it.name>); separator=" "> !>
<first(scope.attributes){ a | :(<a.type><a.name>}> <rest(scope.attributes){ a | arg<i>:(<a.type><a.name> }>
separator=" ">
>>

parameterAttributeRef(attr) ::= "<attr.name>"
parameterSetAttributeRef(attr,expr) ::= "<attr.name> = <expr>";

/** Note that the scopeAttributeRef does not have access to the
 * grammar name directly
 */
scopeAttributeRef(scope,attr,index,negIndex) ::= <%
<if(negIndex)>
(((<scope>_Scope *) [<scope>_stack objectAtIndex:<scope>_stack size-<negIndex>-1])).<attr.name>
<else>
<if(index)>
((<scope>_Scope *) [<scope>_stack objectAtIndex:<index>]).<attr.name>
<else>
((<scope>_Scope *) [<scope>_stack peek]).<attr.name>

```

```

<endif>
<endif>
%>

scopeSetAttributeRef(scope,attr,expr,index,negIndex) ::= <%
/* scopeSetAttributeRef */
<if(negIndex)>
((<scope>_Scope *)[<scope>_stack objectAtIndex:([<scope>_stack size]-<negIndex>-1])).<attr.name> = <expr>;
<else>
<if(index)>
((<scope>_Scope *)[<scope>_stack objectAtIndex:<index>]).<attr.name> = <expr>;
<else>
((<scope>_Scope *)[<scope>_stack peek]).<attr.name> = <expr>;
<endif>
<endif>
%>

scopeAttributeRefStack() ::= <<
/* scopeAttributeRefStack */
<if(negIndex)>
((<scope>_Scope *)[<scope>_stack objectAtIndex:[<scope>_stack count]-<negIndex>-1]).<attr.name> = <expr>;
<else>
<if(index)>
((<scope>_Scope *)[<scope>_stack objectAtIndex:<index>]).<attr.name> = <expr>;
<else>
((<scope>_Scope *)[<scope>_stack peek]).<attr.name> = <expr>;
<endif>
<endif>
>>

/** $x is either global scope or x is rule with dynamic scope; refers
 * to stack itself not top of stack. This is useful for predicates
 * like {$function.size()>0 && $function::name.equals("foo")}?
 */
isolatedDynamicScopeRef(scope) ::= "<scope>_stack"

/** reference an attribute of rule; might only have single return value */
ruleLabelRef(referencedRule,scope,attr) ::= <<
<if(referencedRule.hasMultipleReturnValues)>
(<scope>!=nil?<scope>.<attr.name>:<initValue(attr.type)>)
<else>
<scope>
<endif>
>>

returnAttributeRef(ruleDescriptor,attr) ::= <%
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name> /* added to returnAttributeRef */<\n>

```

```

<else>
<attr.name><\n>
<endif>
%>

returnSetAttributeRef(ruleDescriptor,attr,expr) ::= <%
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name> =<expr>; /* added to returnSetAttributeRef */<\n>
<else>
<attr.name> = <expr>;<\n>
<endif>
%>

/** How to translate $tokenLabel */
tokenLabelRef(label) ::= "<label>"

/** ids+=ID {$ids} or e+=expr {$e} */
listLabelRef(label) ::= "list_<label>"

/* not sure the next are the right approach; and they are evaluated early; */
/* they cannot see TREE_PARSER or PARSER attributes for example. :( */

tokenLabelPropertyRef_text(scope,attr) ::= "<scope>!=nil?<scope>.text:nil)"
tokenLabelPropertyRef_type(scope,attr) ::= "<scope>!=nil?<scope>.type:0)"
tokenLabelPropertyRef_line(scope,attr) ::= "<scope>!=nil?<scope>.line:0)"
tokenLabelPropertyRef_pos(scope,attr) ::= "<scope>!=nil?<scope>.charPositionInLine:0)"
tokenLabelPropertyRef_channel(scope,attr) ::= "<scope>!=nil?<scope>.channel:0)"
tokenLabelPropertyRef_index(scope,attr) ::= "<scope>!=nil?[<scope> getTokenIndex]:0)"
tokenLabelPropertyRef_tree(scope,attr) ::= "<scope>_tree"
tokenLabelPropertyRef_int(scope,attr) ::= "<scope>!=nil?[<scope>.text integerValue]:0)"

ruleLabelPropertyRef_start(scope,attr) ::= "<scope>!=nil?((<labelType> *)<scope>.start):nil)"
ruleLabelPropertyRef_stop(scope,attr) ::= "<scope>!=nil?((<labelType> *)<scope>.stopToken):nil)"
ruleLabelPropertyRef_tree(scope,attr) ::= "<scope>!=nil?((<ASTLabelType> *)<scope>.tree):nil)"
ruleLabelPropertyRef_text(scope,attr) ::= <%
<if(TREE_PARSER)>
<scope>!=nil?[[input getTokenStream] toStringFromStart:[[input getTreeAdaptor] getTokenStartIndex:[<scope>
getStart]]
ToEnd:[[input getTreeAdaptor] getTokenStopIndex:[<scope> getStart]]]:0)
<else>
<scope>!=nil?([input toStringFromToken:[<scope> getStart] ToToken:[<scope> getStop]]):0)
<endif>
%>
ruleLabelPropertyRef_st(scope,attr) ::= "<scope>!=nil?[<scope> st]:nil)"

/** Isolated $RULE ref ok in lexer as it's a Token */
lexerRuleLabel(label) ::= "<label>"

```

```

lexerRuleLabelPropertyRef_type(scope,attr) ::= "<scope>!=nil?<scope>.type:0)"
lexerRuleLabelPropertyRef_line(scope,attr) ::= "<scope>!=nil?<scope>.line:0)"
lexerRuleLabelPropertyRef_pos(scope,attr) ::= "<scope>!=nil?<scope>.charPositionInLine:-1)"
lexerRuleLabelPropertyRef_channel(scope,attr) ::= "<scope>!=nil?<scope>.channel:0)"
lexerRuleLabelPropertyRef_index(scope,attr) ::= "<scope>!=nil?[<scope> getTokenIndex]:0)"
lexerRuleLabelPropertyRef_text(scope,attr) ::= "<scope>!=nil?<scope>.text:nil)"
lexerRuleLabelPropertyRef_int(scope,attr) ::= "<scope>!=nil?[<scope>.text integerValue]:0)"

// Somebody may ref $template or $tree or $stop within a rule:
rulePropertyRef_start(scope,attr) ::= "((<labelType> *)retval.start)"
rulePropertyRef_stop(scope,attr) ::= "((<labelType> *)retval.stopToken)"
rulePropertyRef_tree(scope,attr) ::= "((<ASTLabelType> *)retval.tree)"
rulePropertyRef_text(scope,attr) ::= <<
<if(TREE_PARSER)>
[[input getTokenStream] toStringFromStart:[[input getTreeAdaptor]
getTokenStartIndex:retval.start.token.startIndex]
ToEnd:[[input getTreeAdaptor] getTokenStopIndex:retval.start.token.stopIndex]]
<else>
[input toStringFromToken:retval.start ToToken:[input LT:-1]]
<endif>
>>
rulePropertyRef_st(scope,attr) ::= "retval.st"

/* hideous: find a way to cut down on the number of templates to support read/write access */
/* TODO: also, which ones are valid to write to? ask Ter */
lexerRuleSetPropertyRef_text(scope,attr,expr) ::= "state.text = <expr>;"
lexerRuleSetPropertyRef_type(scope,attr,expr) ::= "_type"
lexerRuleSetPropertyRef_line(scope,attr,expr) ::= "state.tokenStartLine"
lexerRuleSetPropertyRef_pos(scope,attr,expr) ::= "state.tokenStartCharPositionInLine"
lexerRuleSetPropertyRef_index(scope,attr,expr) ::= "-1" /* undefined token index in lexer */
lexerRuleSetPropertyRef_channel(scope,attr,expr) ::= "state.channel=<expr>;"
lexerRuleSetPropertyRef_start(scope,attr,expr) ::= "state.tokenStartCharIndex"
lexerRuleSetPropertyRef_stop(scope,attr,expr) ::= "(input.index-1)"

lexerRulePropertyRef_text(scope,attr) ::= "self.text"
lexerRulePropertyRef_type(scope,attr) ::= "state.type"
lexerRulePropertyRef_line(scope,attr) ::= "state.tokenStartLine"
lexerRulePropertyRef_pos(scope,attr) ::= "state.tokenStartCharPositionInLine"
lexerRulePropertyRef_index(scope,attr) ::= "-1" // undefined token index in lexer
lexerRulePropertyRef_channel(scope,attr) ::= "_channel"
lexerRulePropertyRef_start(scope,attr) ::= "state.tokenStartCharIndex"
lexerRulePropertyRef_stop(scope,attr) ::= "(input.index-1)"
lexerRulePropertyRef_int(scope,attr) ::= "[<scope>.text integerValue]"

// setting $st and $tree is allowed in local rule. everything else
// is flagged as error

```

```
ruleSetPropertyRef_tree(scope,attr,expr) ::= "retval.start =<expr>";
ruleSetPropertyRef_st(scope,attr,expr) ::= "retval.st =<expr>"; /* "<\n>#error StringTemplates are
unsupported<\n>" */
```

```
/** How to execute an action */
```

```
execAction(action) ::= <<
<if(backtracking)>
if ( <actions.(actionScope).synpredgate> ) {
    <action>
}
<else>
<action>
<endif>
>>
```

```
/** How to always execute an action even when backtracking */
```

```
execForcedAction(action) ::= "<action>"
```

```
// M I S C (properties, etc...)
```

```
bitset(name, words64) ::= <<
static ANTLRBitSet *<name>;
static const unsigned long long <name>_data[] = { <words64:{it | <it>LL};separator=", ">};<\n>
>>
```

```
bitsetInit(name, words64) ::= <<
<name> = [[ANTLRBitSet newBitSetWithBits:(const unsigned long long *)<name>_data
Count:(NSUInteger)<length(words64)>] retain];<\n>
>>
```

```
codeFileExtension() ::= ".m"
```

```
true_value() ::= "YES"
```

```
false_value() ::= "NO"
```

```
Found in path(s):
```

```
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-
jar/org/antlr/codegen/templates/ObjC/ObjC.stg
```

```
No license file was found, but licenses were detected in source scan.
```

```
/*
```

```
[The "BSD license"]
```

```
Copyright (c) 2005-2006 Terence Parr
```

```
Copyright (c) 2007-2008 Ronald Blaschke
```

```
All rights reserved.
```

```
Redistribution and use in source and binary forms, with or without
```

modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

group Perl5;

```
/** The overall file structure of a recognizer; stores methods for rules
 * and cyclic DFAs plus support code.
 */
```

```
outputFile(LEXER,PARSER,TREE_PARSER, actionScope, actions,
           docComment, recognizer,
           name, tokens, tokenNames, rules, cyclicDFAs,
           bitsets, buildTemplate, buildAST, rewriteMode, profile,
           backtracking, synpreds, memoize, numRules,
           fileName, ANTLRVersion, generatedTimestamp, trace,
           scopes, superClass, literals) ::=
```

```
<<
```

```
# $ANTLR <ANTLRVersion> <fileName> <generatedTimestamp>
```

```
<actions.(actionScope).header>
```

```
<@imports>
```

```
<if(TREE_PARSER)>
```

```
<endif>
```

```
<if(backtracking)>
```

```
<endif>
```

```
<@end>
```

```
<docComment>
```

```
<recognizer>
```

```
>>
```

```

lexer(grammar, name, tokens, scopes, rules, numRules, labelType="Token",
    filterMode, superClass="ANTLR::Runtime::Lexer") ::= <<
package <name>;

use Carp;
use English qw( -no_match_vars );
use Readonly;
use Switch;

use ANTLR::Runtime::BaseRecognizer;
use ANTLR::Runtime::DFA;
use ANTLR::Runtime::NoViableAltException;

use Moose;

extends 'ANTLR::Runtime::Lexer';

Readonly my $HIDDEN => ANTLR::Runtime::BaseRecognizer->HIDDEN;
sub HIDDEN { $HIDDEN }

use constant {
    <tokens:{ <it.name> => <it.type>, }; separator="\n">
};
<scopes:{<if(it.isDynamicGlobalScope)><globalAttributeScope(scope=it)><endif>}>
<actions.lexer.members>

sub BUILD {
    my ($self, $arg_ref) = @_ ;

    $self->init_dfas();
}

sub get_grammar_file_name {
    return "<fileName>";
}

<if(filterMode)>
<filteringNextToken()>
<endif>
<rules; separator="\n\n">

<synpreds:{p | <lexerSynpred(p)>}>

<cyclicDFAs:{ dfa | has 'dfa<dfa.decisionNumber>'; }; separator="\n">

sub init_dfas {
    my ($self) = @_ ;

```

```

<cyclicDFAs:{dfa |
$self->dfa<dfa.decisionNumber><(<name>::DFA<dfa.decisionNumber>->new({ recognizer => $self }));
}; separator="\n">

return;
}

<cyclicDFAs:cyclicDFA(> <! dump tables for all DFA !>

no Moose;
__PACKAGE__->meta->make_immutable();
1;

>>

perlTypeInitMap ::= [
"$":"undef",
"@":"()",
"%":"()",
default:"undef"
]

/** A override of Lexer.nextToken() that backtracks over mTokens() looking
* for matches. No error can be generated upon error; just rewind, consume
* a token and then try again. backtracking needs to be set as well.
* Make rule memoization happen only at levels above 1 as we start mTokens
* at backtracking==1.
*/
filteringNextToken() ::= <<
public Token nextToken() {
while (true) {
if ( input.LA(1)==CharStream.EOF ) {
return Token.EOF_TOKEN;
}
token = null;
channel = Token.DEFAULT_CHANNEL;
tokenStartCharIndex = input.index();
tokenStartCharPositionInLine = input.getCharPositionInLine();
tokenStartLine = input.getLine();
text = null;
try {
int m = input.mark();
backtracking=1; <! means we won't throw slow exception !>
failed=false;
mTokens();
backtracking=0;
<! mTokens backtracks with synpred at backtracking==2

```

```

        and we set the synpredgate to allow actions at level 1. !>
    if ( failed ) {
        input.rewind(m);
        input.consume(); <! advance one char and try again !>
    }
    else {
        emit();
        return token;
    }
}
}
catch (RecognitionException re) {
    // shouldn't happen in backtracking mode, but...
    reportError(re);
    recover(re);
}
}
}

public void memoize(IntStream input,
    int ruleIndex,
    int ruleStartIndex)
{
    if ( backtracking>1 ) super.memoize(input, ruleIndex, ruleStartIndex);
}

public boolean alreadyParsedRule(IntStream input, int ruleIndex) {
    if ( backtracking>1 ) return super.alreadyParsedRule(input, ruleIndex);
    return false;
}
}
>>

actionGate() ::= "$self->state->backtracking==0"

filteringActionGate() ::= "backtracking==1"

/** How to generate a parser */
genericParser(grammar, name, scopes, tokens, tokenNames, rules, numRules,
    bitsets, inputStreamType, superClass, filterMode,
    ASTLabelType="Object", labelType, members) ::= <<
package <name>;

use English qw( -no_match_vars );
use Readonly;
use Switch;
use Carp;
use ANTLR::Runtime::BitSet;

use Moose;

```

```

extends '<@superClassName><superClass><@end>';

Readonly my $token_names => [
    "\<invalid>", "\<EOR>", "\<DOWN>", "\<UP>", <tokenNames; separator=", ">
];

use constant {
    <tokens:{ <it.name> => <it.type>, }; separator="\n">
};

<bitsets:bitset(name={FOLLOW_<it.name>_in_<it.inName><it.tokenIndex>},
    words64=it.bits)>

<scopes:{<if(it.isDynamicGlobalScope)><globalAttributeScope(scope=it)><endif>}>
<@members>
<! WARNING. bug in ST: this is cut-n-paste into Dbg.stg !>

sub BUILD {
    my ($self, $arg_ref) = @_;

    <if(backtracking)>
        $self->state->rule_memo({});<\n>
    <endif>
}
<@end>

sub get_token_names {
    return $token_names;
}

sub get_grammar_file_name {
    return "<fileName>";
}

<members>

<rules; separator="\n\n">

<synpreds:{p | <synpred(p)>}>

<cyclicDFAs:{ dfa | dfa<dfa.decisionNumber> = __PACKAGE__::DFA<dfa.decisionNumber>->new($self);};
separator="\n">
<cyclicDFAs:cyclicDFA()> <! dump tables for all DFA !>

no Moose;
__PACKAGE__->meta->make_immutable();
1;

```

__END__

>>

```
parser(grammar, name, scopes, tokens, tokenNames, rules, numRules, bitsets, ASTLabelType,
superClass="ANTLR::Runtime::Parser", labelType="ANTLR::Runtime::Token",
members={<actions.parser.members>}) ::= <<
<genericParser(inputStreamType="ANTLR::Runtime::TokenStream", ...)>
>>
```

/** How to generate a tree parser; same as parser except the input

* stream is a different type.

*/

```
treeParser(grammar, name, scopes, tokens, tokenNames, globalAction, rules, numRules, bitsets,
labelType={<ASTLabelType>}, ASTLabelType="Object", superClass="ANTLR::Runtime::TreeParser",
members={<actions.treeparser.members>}, filterMode) ::= <<
<genericParser(inputStreamType="TreeNodeStream", ...)>
>>
```

/** A simpler version of a rule template that is specific to the imaginary

* rules created for syntactic predicates. As they never have return values

* nor parameters etc..., just give simplest possible method. Don't do

* any of the normal memoization stuff in here either; it's a waste.

* As predicates cannot be inlined into the invoking rule, they need to

* be in a rule by themselves.

*/

```
synpredRule(ruleName, ruleDescriptor, block, description, nakedBlock) ::=
```

```
<<
```

```
# $ANTLR start <ruleName>
```

```
sub <ruleName>_fragment {
```

```
# <ruleDescriptor.parameterScope:parameterScope(scope=it)>
```

```
<if(trace)>
```

```
  $self->traceIn("<ruleName>_fragment", <ruleDescriptor.index>);
```

```
  eval {
```

```
    <block>
```

```
  };
```

```
  $self->traceOut("<ruleName>_fragment", <ruleDescriptor.index>);
```

```
  if ($EVAL_ERROR) {
```

```
    croak $EVAL_ERROR;
```

```
  }
```

```
<else>
```

```
  <block>
```

```
<endif>
```

```
}
```

```
# $ANTLR end <ruleName>
```

```
>>
```

```
synpred(name) ::= <<
```

```

public final boolean <name>() {
    backtracking++;
    <@start()>
    int start = input.mark();
    try {
        <name>_fragment(); // can never throw exception
    } catch (RecognitionException re) {
        System.err.println("impossible: "+re);
    }
    boolean success = !failed;
    input.rewind(start);
    <@stop()>
    backtracking--;
    failed=false;
    return success;
}<\n>
>>

lexerSynpred(name) ::= <<
<synpred(name)>
>>

ruleMemoization(name) ::= <<
<if(memoize)>
if ( backtracking>0 && alreadyParsedRule(input, <ruleDescriptor.index>) ) { return <ruleReturnValue()>; }
<endif>
>>

/** How to test for failure and return from rule */
checkRuleBacktrackFailure() ::= <<
<if(backtracking)>
if ($self->state->failed) {
    return <ruleReturnValue()>;
}
<endif>
>>

/** This rule has failed, exit indicating failure during backtrack */
ruleBacktrackFailure() ::= <<
<if(backtracking)>if (backtracking>0) { failed=true; return <ruleReturnValue()>;}<endif>
>>

/** How to generate code for a rule. This includes any return type
 * data aggregates required for multiple return values.
 */
rule(ruleName,ruleDescriptor,block,emptyRule,description,exceptions,finally,memoize) ::= <<
<ruleAttributeScope(scope=ruleDescriptor.ruleScope)>
<returnScope(scope=ruleDescriptor.returnScope)>

```

```

# $ANTLR start <ruleName>
# <fileName>:<description>
sub <ruleName>() {
  my ($self, <ruleDescriptor.parameterScope:parameterScope(scope=it)>) = @_;
  <if(trace)>$self->traceIn("<ruleName>", <ruleDescriptor.index>);<endif>
  <ruleScopeSetUp()>
  <ruleDeclarations()>
  <ruleLabelDefs()>
  <ruleDescriptor.actions.init>
  <@preamble()>
  eval {
    <ruleMemoization(name=ruleName)>
    <block>
    <ruleCleanUp()>
    <(ruleDescriptor.actions.after):execAction()>
  };
  <if(exceptions)>
    <exceptions:{e|<catch(decl=e.decl,action=e.action)><\n}>>
  <else>
  <if(!emptyRule)>
  <if(actions.(actionScope).rulecatch)>
    <actions.(actionScope).rulecatch>
  <else>
    my $exception = $EVAL_ERROR;
    if (ref $exception && $exception->isa('ANTLR::Runtime::RecognitionException')) {
      $self->report_error($exception);
      $self->recover($self->input, $exception);
      $exception = undef;
    }<\n>
  <endif>
  <endif>
  <endif>
  <if(trace)>$self->traceOut("<ruleName>", <ruleDescriptor.index>);<endif>
  <memoize()>
  <ruleScopeCleanUp()>
  <finally>
  if ($exception) {
    croak $exception;
    # $exception->rethrow();
  }
  <@postamble()>
  return <ruleReturnValue()>;
}
# $ANTLR end <ruleName>
>>

```

```

catch(decl,action) ::= <<

```

```

catch (<e.decl>) {
  <e.action>
}
>>

ruleDeclarations() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
my $retval = <returnType()->new();
$retval->set_start($self->input->LT(1));<\n>
<else>
<ruleDescriptor.returnScope.attributes: { a |
my $<a.name> = <if(a.initValue)><a.initValue><else><initValue(a.type)><endif>;
}>
<endif>
<if(memoize)>
my $<ruleDescriptor.name>_start_index = $self->input->index();
<endif>
>>

ruleScopeSetUp() ::= <<
<ruleDescriptor.useScopes: { <it>_stack.push(new <it>_scope());}; separator="\n">
<ruleDescriptor.ruleScope: { <it.name>_stack.push(new <it.name>_scope());}; separator="\n">
>>

ruleScopeCleanUp() ::= <<
<ruleDescriptor.useScopes: { <it>_stack.pop();}; separator="\n">
<ruleDescriptor.ruleScope: { <it.name>_stack.pop();}; separator="\n">
>>

ruleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,ruleDescriptor.tokenListLabels]
: {my $<it.label.text> = undef;}; separator="\n"
>
<[ruleDescriptor.tokenListLabels,ruleDescriptor.ruleListLabels]
: {List list_<it.label.text>=null;}; separator="\n"
>
<ruleDescriptor.ruleLabels:ruleLabelDef(label=it); separator="\n">
<ruleDescriptor.ruleListLabels: {ll|RuleReturnScope <ll.label.text> = null;}; separator="\n">
>>

lexerRuleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,
ruleDescriptor.tokenListLabels,
ruleDescriptor.ruleLabels]
: {<labelType> <it.label.text>=null;}; separator="\n"
>
<ruleDescriptor.charLabels: {my $<it.label.text>;}; separator="\n">
<[ruleDescriptor.tokenListLabels,

```

```

ruleDescriptor.ruleListLabels,
ruleDescriptor.ruleListLabels]
: {List list_<it.label.text>=null;}; separator="\n"
>
>>

```

```

ruleReturnValue() ::= <<
<if(!ruleDescriptor.isSynPred)>
<if(ruleDescriptor.hasReturnValue)>
<if(ruleDescriptor.hasSingleReturnValue)>
$<ruleDescriptor.singleValueReturnName>
<else>
$retval
<endif>
<endif>
<endif>
>>

```

```

ruleCleanup() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(!TREE_PARSER)>
$retval->set_stop($self->input->LT(-1));<\n>
<endif>
<endif>
>>

```

```

memoize() ::= <<
<if(memoize)>
<if(backtracking)>
if ( backtracking>0 ) { memoize(input, <ruleDescriptor.index>, <ruleDescriptor.name>_startIndex); }
<endif>
<endif>
>>

```

```

/** How to generate a rule in the lexer; naked blocks are used for
 * fragment rules.
 */

```

```

lexerRule(ruleName,nakedBlock,ruleDescriptor,block,memoize) ::= <<
# $ANTLR start <ruleName>
sub m_<ruleName> {
# <ruleDescriptor.parameterScope:parameterScope(scope=it)>
my ($self) = @_ ;
<if(trace)>traceIn("<ruleName>", <ruleDescriptor.index>);<endif>
<ruleDeclarations()>
eval {
<if(nakedBlock)>
<ruleMemoization(name=ruleName)>
<lexerRuleLabelDefs()>

```

```

    <ruleDescriptor.actions.init>
    <block><\n>
<else>
    my $_type = <ruleName>;
    my $_channel = $self->DEFAULT_TOKEN_CHANNEL;
    <ruleMemoization(name=ruleName)>
    <lexerRuleLabelDefs()>
    <ruleDescriptor.actions.init>
    <block>
    <ruleCleanUp()>
    $self->state->type($_type);
    $self->state->channel($_channel);
    <(ruleDescriptor.actions.after):execAction()>
<endif>
};
<if(trace)>traceOut("<ruleName>", <ruleDescriptor.index>);<endif>
<memoize()>

if ($EVAL_ERROR) {
    croak $EVAL_ERROR;
}
}
# $ANTLR end <ruleName>
>>

/** How to generate code for the implicitly-defined lexer grammar rule
 * that chooses between lexer rules.
 */
tokensRule(ruleName,nakedBlock,args,block,ruleDescriptor) ::= <<
sub m_tokens {
    my ($self) = @_;
    <block><\n>
}
>>

// S U B R U L E S

/** A (...) subrule with multiple alternatives */
block(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
# <fileName>:<description>
my $alt<decisionNumber> = <maxAlt>;
<decls>
<@predecision()>
<decision>
<@postdecision()>
<@prebranch()>
switch ($alt<decisionNumber>) {
    <alts:altSwitchCase()>

```

```

}
<@postbranch()>
>>

/** A rule block with multiple alternatives */
ruleBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
# <fileName>:<description>
my $alt<decisionNumber> = <maxAlt>;
<decls>
<@predecision()>
<decision>
<@postdecision()>
switch ($alt<decisionNumber>) {
  <alts:altSwitchCase()>
}
>>

ruleBlockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
# <fileName>:<description>
<decls>
<@prealt()>
<alts>
<@postalt()>
>>

/** A special case of a (...) subrule with a single alternative */
blockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
# <fileName>:<description>
<decls>
<@prealt()>
<alts>
<@postalt()>
>>

/** A (..)+ block with 1 or more alternatives */
positiveClosureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
# <fileName>:<description>
my $cnt<decisionNumber> = 0;
<decls>
<@preloop()>
LOOP<decisionNumber>:
while (1) {
  my $alt<decisionNumber> = <maxAlt>;
  <@predecision()>
  <decision>
  <@postdecision()>
  switch ($alt<decisionNumber>) {

```

```

<alts:altSwitchCase()>
else {
  if ( $cnt<decisionNumber> >= 1 ) { last LOOP<decisionNumber> }
  <ruleBacktrackFailure()>
  my $eee =
    ANTLR::Runtime::EarlyExitException->new(<decisionNumber>, $self->input);
  <@earlyExitException()>
  croak $eee;
}
}
++$cnt<decisionNumber>;
}
<@postloop()>
>>

```

positiveClosureBlockSingleAlt ::= positiveClosureBlock

```

/** A (..)* block with 1 or more alternatives */
closureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::=
<<
# <fileName>:<description>
<decls>
<@preloop()>
LOOP<decisionNumber>:
while (1) {
  my $alt<decisionNumber> = <maxAlt>;
  <@predecision()>
  <decision>
  <@postdecision()>
  switch ($alt<decisionNumber>) {
    <alts:altSwitchCase()>
    else { last LOOP<decisionNumber> }
  }
}
<@postloop()>
>>

```

closureBlockSingleAlt ::= closureBlock

```

/** Optional blocks (x)? are translated to (x|) by before code generation
* so we can just use the normal block template
*/
optionalBlock ::= block

```

optionalBlockSingleAlt ::= block

```

/** A case in a switch that jumps to an alternative given the alternative
* number. A DFA predicts the alternative and then a simple switch

```

```

* does the jump to the code that actually matches that alternative.
*/
altSwitchCase() ::= <<
case <i> {
  <@prealt()>
  <it>
}<\n>
>>

/** An alternative is just a list of elements; at outermost level */
alt(elements,altNum,description,autoAST,outerAlt,treeLevel,rew) ::= <<
# <fileName>:<description>
{
  <@declarations()>
  <elements:element()>
  <rew>
  <@cleanup()>
}
>>

/** What to emit when there is no rewrite. For auto build
* mode, does nothing.
*/
noRewrite(rewriteBlockLevel, treeLevel) ::= ""

// E L E M E N T S

/** Dump the elements one per line */
element() ::= <<
<@prematch()>
<it.el><\n>
>>

/** match a token optionally with a label in front */
tokenRef(token,label,elementIndex,terminalOptions) ::= <<
<if(label)>$<label> =<endif>$self->match($self->input, <token>,
$FOLLOW_<token>_in_<ruleName><elementIndex>);
<checkRuleBacktrackFailure()>
>>

/** ids+=ID */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

listLabel(label,elem) ::= <<
if (list_<label>==null) list_<label>=new ArrayList();

```

```

list_<label>.add(<elem>);<\n>
>>

/** match a character */
charRef(char,label) ::= <<
<if(label)>
<label> = $self->input->LA(1);<\n>
<endif>
$self->match(<char>); <checkRuleBacktrackFailure()>
>>

/** match a character range */
charRangeRef(a,b,label) ::= <<
<if(label)>
<label> = $self->input->LA(1);<\n>
<endif>
$self->match_range(<a>,<b>); <checkRuleBacktrackFailure()>
>>

/** For now, sets are interval tests and must be tested inline */
matchSet(s,label,elementIndex,postmatchCode="") ::= <<
<if(label)>
<if(LEXER)>
<label>= $self->input->LA(1);<\n>
<else>
<label>=(<labelType>)input.LT(1);<\n>
<endif>
<endif>
if ( <s> ) {
    $self->input->consume();
    <postmatchCode>
<if(!LEXER)>
    $self->state->error_recovery(0);
<endif>
    <if(backtracking)>failed=false;<endif>
}
else {
    <ruleBacktrackFailure()>
    my $mse =
        ANTLR::Runtime::MismatchedSetException->new(undef, $self->input);
    <@mismatchedSetException()>
<if(LEXER)>
    $self->recover($mse);
    $mse->throw();
<else>
    $mse->throw();
    <! use following code to make it recover inline; remove throw mse;
    $self->recoverFromMismatchedSet($self->input, $mse, $FOLLOW_set_in_<ruleName><elementIndex>);

```

```

!>
<endif>
}<\n>
>>

matchRuleBlockSet ::= matchSet

matchSetAndListLabel(s,label,elementIndex,postmatchCode) ::= <<
<matchSet(...)>
<listLabel(elem=label,...)>
>>

/** Match a string literal */
lexerStringRef(string,label,elementIndex) ::= <<
<if(label)>
int <label>Start = getCharIndex();
$self->match(<string>); <checkRuleBacktrackFailure()>
<labelType> <label> = new CommonToken(input, Token.INVALID_TOKEN_TYPE,
Token.DEFAULT_CHANNEL, <label>Start, getCharIndex()-1);
<else>
$self->match(<string>); <checkRuleBacktrackFailure()><\n>
<endif>
>>

wildcard(label,elementIndex) ::= <<
<if(label)>
<label>=(<labelType>)input.LT(1);<\n>
<endif>
matchAny(input); <checkRuleBacktrackFailure()>
>>

wildcardAndListLabel(label,elementIndex) ::= <<
<wildcard(...)>
<listLabel(elem=label,...)>
>>

/** Match . wildcard in lexer */
wildcardChar(label, elementIndex) ::= <<
<if(label)>
<label> = $self->input->LA(1);<\n>
<endif>
matchAny(); <checkRuleBacktrackFailure()>
>>

wildcardCharListLabel(label, elementIndex) ::= <<
<wildcardChar(...)>
<listLabel(elem=label,...)>
>>

```

```

/** Match a rule reference by invoking it possibly with arguments
 * and a return value or values.
 */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
$self->push_follow($FOLLOW_<rule.name>_in_<ruleName><elementIndex>);
<if(label)>
$self-><rule.name>(<args; separator=", ">);<\n>
<else>
$self-><rule.name>(<args; separator=", ">);<\n>
<endif>
$self->state->_fsp($self->state->_fsp - 1);
<checkRuleBacktrackFailure()>
>>

/** ids+=r */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(elem=label,...)>
>>

/** A lexer rule reference.
 *
 * The 'rule' argument was the target rule name, but now
 * is type Rule, whose toString is same: the rule name.
 * Now though you can access full rule descriptor stuff.
 */
lexerRuleRef(rule,label,args,elementIndex,scope) ::= <<
<if(label)>
int <label>Start<elementIndex> = getCharIndex();
$self->m_<rule>(<args; separator=", ">); <checkRuleBacktrackFailure()>
<label> = new CommonToken(input, Token.INVALID_TOKEN_TYPE, Token.DEFAULT_CHANNEL,
<label>Start<elementIndex>, getCharIndex()-1);
<else>
$self->m_<rule.name>(<args; separator=", ">); <checkRuleBacktrackFailure()>
<endif>
>>

/** i+=INT in lexer */
lexerRuleRefAndListLabel(rule,label,args,elementIndex,scope) ::= <<
<lexerRuleRef(...)>
<listLabel(elem=label,...)>
>>

/** EOF in the lexer */
lexerMatchEOF(label,elementIndex) ::= <<
<if(label)>
int <label>Start<elementIndex> = getCharIndex();

```

```

match(EOF); <checkRuleBacktrackFailure()>
<labelType> <label> = new CommonToken(input, EOF, Token.DEFAULT_CHANNEL,
<label>Start<elementIndex>, getCharIndex()-1);
<else>
match(EOF); <checkRuleBacktrackFailure()>
<endif>
>>

```

```

/** match ^(root children) in tree parser */
tree(root, actionsAfterRoot, children, nullableChildList,
    enclosingTreeLevel, treeLevel) ::= <<
<root:element()>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if ( input.LA(1)==Token.DOWN ) {
    match(input, Token.DOWN, null); <checkRuleBacktrackFailure()>
    <children:element()>
    match(input, Token.UP, null); <checkRuleBacktrackFailure()>
}
<else>
match(input, Token.DOWN, null); <checkRuleBacktrackFailure()>
<children:element()>
match(input, Token.UP, null); <checkRuleBacktrackFailure()>
<endif>
>>

```

```

/** Every predicate is used as a validating predicate (even when it is
 * also hoisted into a prediction expression).
 */
validateSemanticPredicate(pred,description) ::= <<
if ( !(<evalPredicate(...)> ) ) {
    <ruleBacktrackFailure()>
    throw new FailedPredicateException(input, "<ruleName>", "<description>");
}
>>

```

// F i x e d D F A (if-then-else)

```

dfaState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
my $LA<decisionNumber>_<stateNumber> = $self->input->LA(<k>);<\n>
<edges; separator="\nls">
else {
<if(eotPredictsAlt)>
    $alt<decisionNumber> = <eotPredictsAlt>;
<else>
    <ruleBacktrackFailure()>
my $nvae =
    ANTLR::Runtime::NoViableAltException->new({

```

```

        grammar_decision_description => "<description>",
        decision_number => <decisionNumber>,
        state_number => <stateNumber>,
        input => $self->input,
    });<\n>
<@noViableAltException()>
croak $nvae;<\n>
<endif>
}
>>

/** Same as a normal DFA state except that we don't examine lookahead
 * for the bypass alternative. It delays error detection but this
 * is faster, smaller, and more what people expect. For (X)? people
 * expect "if ( LA(1)==X ) match(X);" and that's it.
 */
dfaOptionalBlockState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
my $LA<decisionNumber>_<stateNumber> = $self->input->LA(<k>);<\n>
<edges; separator="\nels">
>>

/** A DFA state that is actually the loopback decision of a closure
 * loop. If end-of-token (EOT) predicts any of the targets then it
 * should act like a default clause (i.e., no error can be generated).
 * This is used only in the lexer so that for ('a')* on the end of a rule
 * anything other than 'a' predicts exiting.
 */
dfaLoopbackState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
my $LA<decisionNumber>_<stateNumber> = $self->input->LA(<k>);<\n>
<edges; separator="\nels"><\n>
<if(eotPredictsAlt)>
<if(!edges)>
$alt<decisionNumber> = <eotPredictsAlt>; <! if no edges, don't gen ELSE !>
<else>
else {
    $alt<decisionNumber> = <eotPredictsAlt>;
}<\n>
<endif>
<endif>
>>

/** An accept state indicates a unique alternative has been predicted */
dfaAcceptState(alt) ::= "$alt<decisionNumber> = <alt>";

/** A simple edge with an expression. If the expression is satisfied,
 * enter to the target state. To handle gated productions, we may
 * have to evaluate some predicates for this edge.
 */

```

```

dfaEdge(labelExpr, targetState, predicates) ::= <<
if ( ( <labelExpr> <if(predicates)>&& (<predicates>)<endif> ) {
    <targetState>
}
>>

// F i x e d D F A (switch case)

/** A DFA state where a SWITCH may be generated. The code generator
 * decides if this is possible: CodeGenerator.canGenerateSwitch().
 */
dfaStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ( $self->input->LA(<k> ) ) {
    <edges; separator="\n">
    else {
        <if(eotPredictsAlt)>
            $alt<decisionNumber> = <eotPredictsAlt>;
        <else>
            <ruleBacktrackFailure()>
            my $nvae =
                ANTLR::Runtime::NoViableAltException->new({
                    grammar_decision_description => "<description>",
                    decision_number => <decisionNumber>,
                    state_number => <stateNumber>,
                    input => $self->input,
                });<\n>
            <@noViableAltException()>
            croak $nvae;<\n>
        <endif>
    }
}<\n>
>>

dfaOptionalBlockStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ( $self->input->LA(<k> ) ) {
    <edges; separator="\n">
}<\n>
>>

dfaLoopbackStateSwitch(k, edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ( $self->input->LA(<k> ) ) {
    <edges; separator="\n"><\n>
    <if(eotPredictsAlt)>
    else { $alt<decisionNumber> = <eotPredictsAlt> }<\n>
    <endif>
}<\n>
>>

```

```

dfaEdgeSwitch(labels, targetState) ::= <<
case [<labels: { <it> }; separator=", ">] { <targetState> }
>>

// C y c l i c D F A

/** The code to initiate execution of a cyclic DFA; this is used
 * in the rule to predict an alt just like the fixed DFA case.
 * The <name> attribute is inherited via the parser, lexer, ...
 */
dfaDecision(decisionNumber,description) ::= <<
$alt<decisionNumber> = $self->dfa<decisionNumber>->predict($self->input);
>>

/* Dump DFA tables as run-length-encoded Strings of octal values.
 * Can't use hex as compiler translates them before compilation.
 * These strings are split into multiple, concatenated strings.
 * Java puts them back together at compile time thankfully.
 * Java cannot handle large static arrays, so we're stuck with this
 * encode/decode approach. See analysis and runtime DFA for
 * the encoding methods.
 */
cyclicDFA(dfa) ::= <<
Readonly my $DFA<dfa.decisionNumber>_eot => ANTLR::Runtime::DFA->unpack_rle([
<dfa.javaCompressedEOT; separator=", "> ]);
Readonly my $DFA<dfa.decisionNumber>_eof => ANTLR::Runtime::DFA->unpack_rle([
<dfa.javaCompressedEOF; separator=", "> ]);
Readonly my $DFA<dfa.decisionNumber>_min => ANTLR::Runtime::DFA->unpack_rle([
<dfa.javaCompressedMin; separator=", "> ]);
Readonly my $DFA<dfa.decisionNumber>_max => ANTLR::Runtime::DFA->unpack_rle([
<dfa.javaCompressedMax; separator=", "> ]);
Readonly my $DFA<dfa.decisionNumber>_accept => ANTLR::Runtime::DFA->unpack_rle([
<dfa.javaCompressedAccept; separator=", "> ]);
Readonly my $DFA<dfa.decisionNumber>_special => ANTLR::Runtime::DFA->unpack_rle([
<dfa.javaCompressedSpecial; separator=", "> ]);
Readonly my $DFA<dfa.decisionNumber>_transition => [
<dfa.javaCompressedTransition: {s|ANTLR::Runtime::DFA->unpack_rle([ <s; separator=", "> ])}; separator=", "> ];

{
package <name>::DFA<dfa.decisionNumber>;
use ANTLR::Runtime::Class;

use strict;
use warnings;

extends 'ANTLR::Runtime::DFA';

sub BUILD {

```

```

my $self = shift;
my $param_ref = __PACKAGE__->unpack_params(@_, {
    spec => [
        {
            name => 'recognizer',
            isa => 'ANTLR::Runtime::BaseRecognizer'
        },
    ]
});

$self->recognizer($param_ref->{recognizer});
$self->decision_number(<dfa.decisionNumber>);
$self->eot($DFA<dfa.decisionNumber>_eot);
$self->eof($DFA<dfa.decisionNumber>_eof);
$self->min($DFA<dfa.decisionNumber>_min);
$self->max($DFA<dfa.decisionNumber>_max);
$self->accept($DFA<dfa.decisionNumber>_accept);
$self->special($DFA<dfa.decisionNumber>_special);
$self->transition($DFA<dfa.decisionNumber>_transition);
}

sub get_description {
    return "<dfa.description>";
}

<@errorMethod()>

<if(dfa.specialStateSTs)>
sub special_state_transition {
    my ($self, $param_ref) = unpack_params(@_, {
        spec => [
            {
                name => 's',
                type => SCALAR,
            },
            {
                name => 'input',
                isa => 'ANTLR::Runtime::IntStream',
            }
        ]
    });
    my $s = $param_ref->{s};
    my $input = $param_ref->{input};

    switch ($s) {
        <dfa.specialStateSTs:{state |
            case <i0> \{ <! compressed special state numbers 0..n-1 !>
                <state>; separator="\n">

```

```

    }
}

<if(backtracking)>
  if ($self->state->backtracking > 0) {
    $self->state->failed = 1;
    return -1;
  }<\n>
<endif>

my $nvae =
  ANTLR::Runtime::NoViableAltException->new({
    grammar_decision_description => $self->get_description(),
    decision_number => <dfa.decisionNumber>,
    state_number => $s,
    input => $input,
  });<\n>
  $self->error($nvae);
  $nvae->throw();
}<\n>
<endif>
}<\n>
>>

/** A state in a cyclic DFA; it's a special state and part of a big switch on
 * state.
 */
cyclicDFAState(decisionNumber,stateNumber,edges,needErrorClause,semPredState) ::= <<
my $input = $self->input;
my $LA<decisionNumber>_<stateNumber> = $input->LA(1);<\n>
<if(semPredState)> <! get next lookahead symbol to test edges, then rewind !>
my $index<decisionNumber>_<stateNumber> = $input->index();
$input->rewind();<\n>
<endif>
s = -1;
<edges; separator="\nls">
<if(semPredState)> <! return input cursor to state before we rewound !>
input.seek(index<decisionNumber>_<stateNumber>);<\n>
<endif>
if ( s>=0 ) return s;
break;
>>

/** Just like a fixed DFA edge, test the lookahead and indicate what
 * state to jump to next if successful.
 */
cyclicDFAEdge(labelExpr, targetStateNumber, edgeNumber, predicates) ::= <<
if ( (<labelExpr> <if(predicates)>&& (<predicates>)<endif> ) { s = <targetStateNumber>;}<\n>

```

```

>>

/** An edge pointing at end-of-token; essentially matches any char;
 * always jump to the target.
 */
eotDFAEdge(targetStateNumber,edgeNumber, predicates) ::= <<
s = <targetStateNumber>;<\n>
>>

// D F A E X P R E S S I O N S

andPredicates(left,right) ::= "<left> && <right>"

orPredicates(operands) ::= "<operands; separator=\\|\\>"

notPredicate(pred) ::= "!(<evalPredicate(...)>)"

evalPredicate(pred,description) ::= "<pred>"

evalSynPredicate(pred,description) ::= "<pred>()"

lookaheadTest(atom,k,atomAsInt) ::= "$LA<decisionNumber>_<stateNumber> eq <atom>"

/** Sometimes a lookahead test cannot assume that LA(k) is in a temp variable
 * somewhere. Must ask for the lookahead directly.
 */
isolatedLookaheadTest(atom,k,atomAsInt) ::= "$self->input->LA(<k>) eq <atom>"

lookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= <<
($LA<decisionNumber>_<stateNumber> ge <lower> && $LA<decisionNumber>_<stateNumber> le <upper>)
>>

isolatedLookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= "($self->input->LA(<k>) ge
<lower> && $self->input->LA(<k>) le <upper>)"

setTest(ranges) ::= "<ranges; separator=\\|\\>"

// A T T R I B U T E S

globalAttributeScope(scope) ::= <<
<if(scope.attributes)>
protected static class <scope.name>_scope {
  <scope.attributes:{<it.decl>;}; separator="\n">
}
protected Stack <scope.name>_stack = new Stack();<\n>
<endif>
>>

```

```

ruleAttributeScope(scope) ::= <<
<if(scope.attributes)>
protected static class <scope.name>_scope {
  <scope.attributes:{<it.decl>;}; separator="\n">
}
protected Stack <scope.name>_stack = new Stack();<\n>
<endif>
>>

returnType() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<ruleDescriptor.name>_return
<else>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnType>
<else>
void
<endif>
<endif>
>>

/** Generate the Java type associated with a single or multiple return
 * values.
 */
ruleLabelType(referencedRule) ::= <<
<if(referencedRule.hasMultipleReturnValues)>
<referencedRule.name>_return
<else>
<if(referencedRule.hasSingleReturnValue)>
<referencedRule.singleValueReturnType>
<else>
void
<endif>
<endif>
>>

/** Using a type to init value map, try to init a type; if not in table
 * must be an object, default value is "undef".
 */
initValue(typeName) ::= <<
<if(typeName)>
<perlTypeInitMap.(typeName)>
<else>
undef
<endif>
>>

```

```

/** Define a rule label including default value */
ruleLabelDef(label) ::= <<
my $<label.label.text> = <initValue(typeName=ruleLabelType(referencedRule=label.referencedRule))>;<\n>
>>

/** Define a return struct for a rule if the code needs to access its
 * start/stop tokens, tree stuff, attributes, ... Leave a hole for
 * subgroups to stick in members.
 */
returnScope(scope) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
{
package <returnType()>;
use ANTLR::Runtime::Class;

extends 'ANTLR::Runtime::<if(TREE_PARSER)>Tree<else>Parser<endif>RuleReturnScope';

scope.attributes:{public <it.decl>;}; separator="\n"
<@ruleReturnMembers()>
}
<endif>
>>

parameterScope(scope) ::= <<
<scope.attributes:{<$<it.name>}; separator=", ">
>>

parameterAttributeRef(attr) ::= "<$<attr.name>"
parameterSetAttributeRef(attr,expr) ::= "<$<attr.name> =<expr>";

scopeAttributeRef(scope,attr,index,negIndex) ::= <<
<if(negIndex)>
((<scope>_scope)<scope>_stack.elementAt(<scope>_stack.size()-<negIndex>-1)).<attr.name>
<else>
<if(index)>
((<scope>_scope)<scope>_stack.elementAt(<index>)).<attr.name>
<else>
((<scope>_scope)<scope>_stack.peek()).<attr.name>
<endif>
<endif>
>>

scopeSetAttributeRef(scope,attr,expr,index,negIndex) ::= <<
<if(negIndex)>
((<scope>_scope)<scope>_stack.elementAt(<scope>_stack.size()-<negIndex>-1)).<attr.name> =<expr>;
<else>
<if(index)>
((<scope>_scope)<scope>_stack.elementAt(<index>)).<attr.name> =<expr>;

```

```

<else>
((<scope>_scope)<scope>_stack.peek()).<attr.name> =<expr>;
<endif>
<endif>
>>

/** $x is either global scope or x is rule with dynamic scope; refers
 * to stack itself not top of stack. This is useful for predicates
 * like {$function.size()>0 && $function::name.equals("foo")}?
 */
isolatedDynamicScopeRef(scope) ::= "<scope>_stack"

/** reference an attribute of rule; might only have single return value */
ruleLabelRef(referencedRule,scope,attr) ::= <<
<if(referencedRule.hasMultipleReturnValues)>
$<scope>.<attr.name>
<else>
$<scope>
<endif>
>>

returnAttributeRef(ruleDescriptor,attr) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name>
<else>
$<attr.name>
<endif>
>>

returnSetAttributeRef(ruleDescriptor,attr,expr) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name> =<expr>;
<else>
$<attr.name> =<expr>;
<endif>
>>

/** How to translate $tokenLabel */
tokenLabelRef(label) ::= "$<label>"

/** ids+=ID {$ids} or e+=expr {$e} */
listLabelRef(label) ::= "list_<label>"

// not sure the next are the right approach

tokenLabelPropertyRef_text(scope,attr) ::= "$<scope>->get_text()"
tokenLabelPropertyRef_type(scope,attr) ::= "<scope>.getType()"

```

```

tokenLabelPropertyRef_line(scope,attr) ::= "<scope>.getLine()"
tokenLabelPropertyRef_pos(scope,attr) ::= "<scope>.getCharPositionInLine()"
tokenLabelPropertyRef_channel(scope,attr) ::= "<scope>.getChannel()"
tokenLabelPropertyRef_index(scope,attr) ::= "<scope>.getTokenIndex()"
tokenLabelPropertyRef_tree(scope,attr) ::= "<scope>_tree"

ruleLabelPropertyRef_start(scope,attr) ::= "((<labelType>)<scope>.start)"
ruleLabelPropertyRef_stop(scope,attr) ::= "((<labelType>)<scope>.stop)"
ruleLabelPropertyRef_tree(scope,attr) ::= "((<ASTLabelType>)<scope>.tree)"
ruleLabelPropertyRef_text(scope,attr) ::= <<
<if(TREE_PARSER)>
input.getTokenStream().toString(
input.getTreeAdaptor().getTokenStartIndex(<scope>.start),
input.getTreeAdaptor().getTokenStopIndex(<scope>.start))
<else>
substr($self->input, $<scope>->start, $<scope>->stop)
<endif>
>>

ruleLabelPropertyRef_st(scope,attr) ::= "<scope>.st"

/** Isolated $RULE ref ok in lexer as it's a Token */
lexerRuleLabel(label) ::= "$<label>"

lexerRuleLabelPropertyRef_type(scope,attr) ::= "<scope>.getType()"
lexerRuleLabelPropertyRef_line(scope,attr) ::= "<scope>.getLine()"
lexerRuleLabelPropertyRef_pos(scope,attr) ::= "<scope>.getCharPositionInLine()"
lexerRuleLabelPropertyRef_channel(scope,attr) ::= "<scope>.getChannel()"
lexerRuleLabelPropertyRef_index(scope,attr) ::= "<scope>.getTokenIndex()"
lexerRuleLabelPropertyRef_text(scope,attr) ::= "<scope>.getText()"

// Somebody may ref $template or $tree or $stop within a rule:
rulePropertyRef_start(scope,attr) ::= "((<labelType>)retval.start)"
rulePropertyRef_stop(scope,attr) ::= "((<labelType>)retval.stop)"
rulePropertyRef_tree(scope,attr) ::= "((<ASTLabelType>)retval.tree)"
rulePropertyRef_text(scope,attr) ::= <<
<if(TREE_PARSER)>
input.getTokenStream().toString(
input.getTreeAdaptor().getTokenStartIndex(retval.start),
input.getTreeAdaptor().getTokenStopIndex(retval.start))
<else>
input.toString(retval.start,input.LT(-1))
<endif>
>>
rulePropertyRef_st(scope,attr) ::= "retval.st"

lexerRulePropertyRef_text(scope,attr) ::= "getText()"
lexerRulePropertyRef_type(scope,attr) ::= "$_type"

```

```

lexerRulePropertyRef_line(scope,attr) ::= "tokenStartLine"
lexerRulePropertyRef_pos(scope,attr) ::= "tokenStartCharPositionInLine"
lexerRulePropertyRef_index(scope,attr) ::= "-1" // undefined token index in lexer
lexerRulePropertyRef_channel(scope,attr) ::= "$_channel"
lexerRulePropertyRef_start(scope,attr) ::= "tokenStartCharIndex"
lexerRulePropertyRef_stop(scope,attr) ::= "(getCharIndex()-1)"
lexerRulePropertyRef_self(scope,attr) ::= "$self"

// setting $st and $tree is allowed in local rule. everything else
// is flagged as error
ruleSetPropertyRef_tree(scope,attr,expr) ::= "retval.tree =<expr>";
ruleSetPropertyRef_st(scope,attr,expr) ::= "retval.st =<expr>";

/** How to execute an action */
execAction(action) ::= <<
<if(backtracking)>
<if(actions.(actionScope).synpredgate)>
if ( <actions.(actionScope).synpredgate> ) {
  <action>
}
<else>
if ( backtracking==0 ) {
  <action>
}
<endif>
<else>
<action>
<endif>
>>

// M I S C (properties, etc...)

bitset(name, words64) ::= <<
Readonly my $<name> => ANTLR::Runtime::BitSet->new({ words64 => [ <words64:{'<it>'};separator=", "> ]
});<\n>
>>

codeFileExtension() ::= ".pm"

true() ::= "1"
false() ::= "0"

Found in path(s):
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-
jar/org/antlr/codegen/templates/Perl5/Perl5.stg
No license file was found, but licenses were detected in source scan.

```

/*

[The "BSD license"]

Copyright (c) 2005-2006 Terence Parr

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

/** Templates for building ASTs during tree parsing.

*

* Deal with many combinations. Dimensions are:

* Auto build or rewrite

* no label, label, list label (label/no-label handled together)

* child, root

* token, set, rule, wildcard

*

* Each combination has its own template except that label/no label

* is combined into tokenRef, ruleRef, ...

*/

/** Add a variable to track last element matched */

```
ruleDeclarations() ::= <<
```

```
<super.ruleDeclarations()>
```

```
<if(!ruleDescriptor.isSynPred)>
```

```
<ASTLabelType> _first_0 = null;
```

```
<ASTLabelType> _last = null;<\n>
```

```
<endif>
```

```
>>
```

```

/** What to emit when there is no rewrite rule. For auto build
 * mode, does nothing.
 */
noRewrite(rewriteBlockLevel=false, treeLevel=false) ::= <<
<if(!ruleDescriptor.isSynPred)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<if(rewriteMode)>
retval.tree = _first_0;
if ( adaptor.getParent(retval.tree)!=null && adaptor.isNil( adaptor.getParent(retval.tree) ) )
retval.tree = (<ASTLabelType>)adaptor.getParent(retval.tree);
<endif>
<if(backtracking)>}<endif>
<endif>
>>

/** match ^(root children) in tree parser; override here to
 * add tree construction actions.
 */
tree(root, actionsAfterRoot, children, nullableChildList,
enclosingTreeLevel, treeLevel) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
{
<ASTLabelType> _save_last_<treeLevel> = _last;
<ASTLabelType> _first_<treeLevel> = null;
<if(!rewriteMode)>
<ASTLabelType> root_<treeLevel> = (<ASTLabelType>)adaptor.nil();
<endif>
<root:element()>
<if(rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> )<endif>
<if(root.el.rule)>
if ( _first_<enclosingTreeLevel>==null ) _first_<enclosingTreeLevel> =
(<ASTLabelType>)<root.el.label>.getTree();
<elseif(root.el.label)>
if ( _first_<enclosingTreeLevel>==null ) _first_<enclosingTreeLevel> = <root.el.label>;
<endif>
<endif>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if ( input.LA(1)==Token.DOWN ) {
match(input, Token.DOWN, null); <checkRuleBacktrackFailure()>
<children:element()>
match(input, Token.UP, null); <checkRuleBacktrackFailure()>
}
<else>
match(input, Token.DOWN, null); <checkRuleBacktrackFailure()>

```

```

<children:element()>
match(input, Token.UP, null); <checkRuleBacktrackFailure()>
<endif>
<if(!rewriteMode)>
adaptor.addChild(root_<enclosingTreeLevel>, root_<treeLevel>);
<endif>
_last = _save_last_<treeLevel>;
}<\n>
<else>
<super.tree(...)>
<endif>
>>

// TOKEN AST STUFF

/** ID! and output=AST (same as plain tokenRef) 'cept add
 * setting of _last
 */
tokenRefBang(token,label,elementIndex,terminalOptions={ }) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.tokenRef(...)>
<else>
<super.tokenRefBang(...)>
<endif>
>>

/** ID auto construct */
tokenRef(token,label,elementIndex,terminalOptions={ }) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.tokenRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<if(terminalOptions.node)>
<label>_tree = new <terminalOptions.node>(<label>);
<else>
<label>_tree = (<ASTLabelType>)adaptor.dupNode(<label>);
<endif><\n>
adaptor.addChild(root_<treeLevel>, <label>_tree);
<if(backtracking)>}<endif>
<else> <! rewrite mode !>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> )<endif>
if ( _first_<treeLevel>==null ) _first_<treeLevel> = <label>;
<endif>
<else>
<super.tokenRef(...)>
<endif>

```

>>

```
/** label+=TOKEN auto construct */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<if(!ruleDescriptor.isSynPred)>
<tokenRef(...)>
<listLabel(elem=label,...)>
<else>
<super.tokenRefAndListLabel(...)>
<endif>
>>
```

```
/** ^(ID ...) auto construct */
tokenRefRuleRoot(token,label,elementIndex,terminalOptions={}) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.tokenRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<if(terminalOptions.node)>
<label>_tree = new <terminalOptions.node>(<label>);
<else>
<label>_tree = (<ASTLabelType>)adaptor.dupNode(<label>);
<endif><\n>
root_<treeLevel> = (<ASTLabelType>)adaptor.becomeRoot(<label>_tree, root_<treeLevel>);
<if(backtracking)>}<endif>
<endif>
<else>
<super.tokenRefRuleRoot(...)>
<endif>
>>
```

```
/** Match ^(label+=TOKEN ...) auto construct */
tokenRefRuleRootAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<if(!ruleDescriptor.isSynPred)>
<tokenRefRuleRoot(...)>
<listLabel(elem=label,...)>
<else>
<super.tokenRefRuleRootAndListLabel(...)>
<endif>
>>
```

```
/** Match . wildcard and auto dup the node/subtree */
wildcard(token,label,elementIndex,terminalOptions={}) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.wildcard(...)>
<if(!rewriteMode)>
```

```

<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<label>_tree = (<ASTLabelType>)adaptor.dupTree(<label>);
adaptor.addChild(root_<treeLevel>, <label>_tree);
<if(backtracking)>}<endif>
<else> <! rewrite mode !>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> )<endif>
if ( _first_<treeLevel>==null ) _first_<treeLevel> = <label>;
<endif>
<else>
<super.wildcard(...)>
<endif>
>>

```

```
// SET AST
```

```

matchSet(s,label,elementIndex,postmatchCode,terminalOptions={}) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.matchSet(postmatchCode={
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<if(terminalOptions.node)>
<label>_tree = new <terminalOptions.node>(<label>);
<else>
<label>_tree = (<ASTLabelType>)adaptor.dupNode(<label>);
<endif><\n>
adaptor.addChild(root_<treeLevel>, <label>_tree);
<if(backtracking)>}<endif>
<endif>
}, ...
)>
<else>
<super.matchSet(...)>
<endif>
>>

```

```

matchRuleBlockSet(s,label,elementIndex,postmatchCode,treeLevel="0",terminalOptions={}) ::= <<
<if(!ruleDescriptor.isSynPred)>
<matchSet(...)>
<noRewrite(...)> <! set return tree !>
<else>
<super.matchRuleBlockSet(...)>
<endif>
>>

```

```

matchSetBang(s,label,elementIndex,postmatchCode,terminalOptions={}) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);

```

```

<super.matchSet(...)>
<else>
<super.matchSetBang(...)>
<endif>
>>

matchSetRuleRoot(s,label,elementIndex,debug,terminalOptions={ }) ::= <<
<if(!ruleDescriptor.isSynPred)>
<super.matchSet(postmatchCode={
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<if(terminalOptions.node)>
<label>_tree = new <terminalOptions.node>(<label>);
<else>
<label>_tree = (<ASTLabelType>)adaptor.dupNode(<label>);
<endif><\n>
root_<treeLevel> = (<ASTLabelType>)adaptor.becomeRoot(<label>_tree, root_<treeLevel>);
<if(backtracking)>\}<endif>
<endif>
}, ...
)>
<else>
<super.matchSetRuleRoot(...)>
<endif>
>>

// RULE REF AST

/** rule auto construct */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>
<if(!rewriteMode)>
adaptor.addChild(root_<treeLevel>, <label>.getTree());
<else> <! rewrite mode !>
if ( _first_<treeLevel>==null ) _first_<treeLevel> = (<ASTLabelType>)<label>.getTree();
<endif>
<else>
<super.ruleRef(...)>
<endif>
>>

/** x+=rule auto construct */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<if(!ruleDescriptor.isSynPred)>
<ruleRef(...)>

```

```

<listLabel(label, {<label>.getTree()})>
<else>
<super.ruleRefAndListLabel(...)>
<endif>
>>

/** ^(rule ...) auto construct */
ruleRefRuleRoot(rule,label,elementIndex,args,scope) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>root_<treeLevel> =
(<ASTLabelType>)adaptor.becomeRoot(<label>.getTree(), root_<treeLevel>);
<endif>
<else>
<super.ruleRefRuleRoot(...)>
<endif>
>>

/** ^(x+=rule ...) auto construct */
ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<if(!ruleDescriptor.isSynPred)>
<ruleRefRuleRoot(...)>
<listLabel(label, {<label>.getTree()})>
<else>
<super.ruleRefRuleRootAndListLabel(...)>
<endif>
>>

/** rule when output=AST and tracking for rewrite */
ruleRefTrack(rule,label,elementIndex,args,scope) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRefTrack(...)>
>>

/** x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRefTrackAndListLabel(...)>
<else>
<super.ruleRefTrackAndListLabel(...)>
<endif>
>>

/** ^(rule ...) rewrite */
ruleRefRuleRootTrack(rule,label,elementIndex,args,scope) ::= <<

```

```

<if(!ruleDescriptor.isSynPred)>
  _last = (<ASTLabelType>)input.LT(1);
<super.ruleRefRootTrack(...)>
<else>
<super.ruleRefRootTrack(...)>
<endif>
>>

/** ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<if(!ruleDescriptor.isSynPred)>
  _last = (<ASTLabelType>)input.LT(1);
<super.ruleRefRuleRootTrackAndListLabel(...)>
<else>
<super.ruleRefRuleRootTrackAndListLabel(...)>
<endif>
>>

/** Streams for token refs are tree nodes now; override to
 * change nextToken to nextNode.
 */
createRewriteNodeFromElement(token,args,terminalOptions={}) ::= <<
<if(terminalOptions.node)>
new <terminalOptions.node>(stream_<token>.nextNode())
<else>
stream_<token>.nextNode()
<endif>
>>

ruleCleanup() ::= <<
<super.ruleCleanup()>
<if(!ruleDescriptor.isSynPred)>
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<\n><endif>
retval.tree = (<ASTLabelType>)adaptor.rulePostProcessing(root_0);
<if(backtracking)>}<endif>
<endif>
<endif>
>>

```

Found in path(s):

```

* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-
jar/org/antlr/codegen/templates/Java/ASTTreeParser.stg

```

No license file was found, but licenses were detected in source scan.

```

/*

```

```

[The "BSD license"]

```

```

Copyright (c) 2010 Terence Parr

```

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

```
asTypeInitMap ::= [
  "int": "0",
  "uint": "0",
  "Number": "0.0",
  "Boolean": "false",
  default: "null" // anything other than an atomic type
]

/** The overall file structure of a recognizer; stores methods for rules
 * and cyclic DFAs plus support code.
 */
outputFile(LEXER,PARSER,TREE_PARSER, actionScope, actions,
  docComment, recognizer,
  name, tokens, tokenNames, rules, cyclicDFAs,
  bitsets, buildTemplate, buildAST, rewriteMode, profile,
  backtracking, synpreds, memoize, numRules,
  fileName, ANTLRVersion, generatedTimestamp, trace,
  scopes, superClass, literals) ::=
<<
// $ANTLR <ANTLRVersion> <fileName> <generatedTimestamp>
package<if(actions.(actionScope).package)> <actions.(actionScope).package><endif> {
  <actions.(actionScope).header>
  <@imports>
```

```

import org.antlr.runtime.*;
<if(TREE_PARSER)>
  import org.antlr.runtime.tree.*;
<endif>
<@end>

<docComment>
<recognizer>
}
>>

lexer(grammar, name, tokens, scopes, rules, numRules, filterMode, labelType="Token",
  superClass="Lexer") ::= <<
public class <grammar.recognizerName> extends
<if(actions.(actionScope).superClass)><actions.(actionScope).superClass><else><@superClassName><superClass>
><@end><endif> {
  <tokens:{it |public static const <it.name>:int=<it.type>;}; separator="\n">
  <scopes:{it |<if(it.isDynamicGlobalScope)><globalAttributeScope(it)><endif>}>
  <actions.lexer.members>

  // delegates
  <grammar.delegates:
    {g|public var <g.delegateName():><g.recognizerName>;}; separator="\n">
  // delegators
  <grammar.delegators:
    {g|public var <g.delegateName():><g.recognizerName>;}; separator="\n">
  <last(grammar.delegators):{g|gParent:<g.recognizerName>;}>

  public function <grammar.recognizerName>(<grammar.delegators:{g|<g.delegateName():><g.recognizerName>,
}>input:CharStream = null, state:RecognizerSharedState = null) {
    super(input, state);
    <cyclicDFAs:cyclicDFACTor(>
<if(memoize)>
<if(grammar.grammarIsRoot)>
  this.state.ruleMemo = new Array(<numRules>+1);<\n><! index from 1..n !>
<endif>
<endif>
  <grammar.directDelegates:
    {g|<g.delegateName()> = new <g.recognizerName>(<trunc(g.delegators):{p|<p.delegateName()>, }>this,
input, this.state);}; separator="\n">
  <grammar.delegators:
    {g|this.<g.delegateName()> = <g.delegateName()>;}; separator="\n">
  <last(grammar.delegators):{g|gParent = <g.delegateName()>;}>
  }
  public override function get grammarFileName():String { return "<fileName>"; }

<if(filterMode)>
  <filteringNextToken(>

```

```

<endif>
  <rules; separator="\n\n">

  <synpreds:{p | <lexerSynpred(p)>}>

  <cyclicDFAs:cyclicDFA()> <! dump tables for all DFA !>

}
>>

/** A override of Lexer.nextToken() that backtracks over mTokens() looking
 * for matches. No error can be generated upon error; just rewind, consume
 * a token and then try again. backtracking needs to be set as well.
 * Make rule memoization happen only at levels above 1 as we start mTokens
 * at backtracking==1.
 */
filteringNextToken() ::= <<
public override function nextToken():Token {
  while (true) {
    if ( input.LA(1)==CharStreamConstants.EOF ) {
      return TokenConstants.EOF_TOKEN;
    }
    this.state.token = null;
    this.state.channel = TokenConstants.DEFAULT_CHANNEL;
    this.state.tokenStartCharIndex = input.index;
    this.state.tokenStartCharPositionInLine = input.charPositionInLine;
    this.state.tokenStartLine = input.line;
    this.state.text = null;
    try {
      var m:int = input.mark();
      this.state.backtracking=1; <! means we won't throw slow exception !>
      this.state.failed=false;
      mTokens();
      this.state.backtracking=0;
      <! mTokens backtracks with synpred at backtracking==2
       and we set the synpredgate to allow actions at level 1. !>
      if ( this.state.failed ) {
        input.rewindTo(m);
        input.consume(); <! advance one char and try again !>
      }
      else {
        emit();
        return this.state.token;
      }
    }
    catch (re:RecognitionException) {
      // shouldn't happen in backtracking mode, but...
      reportError(re);
    }
  }
}

```

```

        recover(re);
    }
}
// Not reached - For ActionScript compiler
throw new Error();
}

public override function memoize(input:InputStream,
    ruleIndex:int,
    ruleStartIndex:int):void
{
if ( this.state.backtracking>1 ) super.memoize(input, ruleIndex, ruleStartIndex);
}

public override function alreadyParsedRule(input:InputStream, ruleIndex:int):Boolean {
if ( this.state.backtracking>1 ) return super.alreadyParsedRule(input, ruleIndex);
return false;
}
>>

actionGate() ::= "this.state.backtracking==0"

filteringActionGate() ::= "this.state.backtracking==1"

/** How to generate a parser */
genericParser(grammar, name, scopes, tokens, tokenNames, rules, numRules,
    bitsets, inputStreamType, superClass,
    labelType, members, rewriteElementType,
    filterMode, ASTLabelType="Object") ::= <<
public class <grammar.recognizerName> extends
<if(actions.(actionScope).superClass)><actions.(actionScope).superClass><else><@superClassName><superClass>
<@end><endif> {
<if(grammar.grammarIsRoot)>
    public static const tokenNames:Array = [
        "\<invalid>", "\<EOR>", "\<DOWN>", "\<UP>", <tokenNames; separator=", ">
    ];<\n>
<endif>
    <tokens:{it |public static const <it.name>:int=<it.type>;}; separator="\n">

    // delegates
    <grammar.delegates: {g|public var <g.delegateName()>:<g.recognizerName>;}; separator="\n">
    // delegators
    <grammar.delegators:
        {g|public var <g.delegateName()>:<g.recognizerName>;}; separator="\n">
    <last(grammar.delegators):{g|public var gParent:<g.recognizerName>;}>

    <scopes:{it |<if(it.isDynamicGlobalScope)><globalAttributeScope(it)><endif>}>
    <@members>

```

```

<! WARNING. bug in ST: this is cut-n-paste into Dbg.stg !>
public function <grammar.recognizerName>(<grammar.delegators:{g|<g.delegateName():<g.recognizerName>,
}>input:<inputStreamType>, state:RecognizerSharedState = null) {
    super(input, state);
    <cyclicDFAs:cyclicDFACTOR()>
    <parserCtorBody()>
    <grammar.directDelegates:
        {g|<g.delegateName()> = new <g.recognizerName>(<trunc(g.delegators):{p|<p.delegateName()>, }>this,
input, this.state);}; separator="\n">
    <grammar.indirectDelegates:{g | <g.delegateName()> = <g.delegator.delegateName()>.<g.delegateName()>;};
separator="\n">
    <last(grammar.delegators):{g|gParent = <g.delegateName()>;}>
    }
    <@end>

public override function get tokenNames():Array { return
<grammar.composite.rootGrammar.recognizerName>.tokenNames; }
public override function get grammarFileName():String { return "<fileName>"; }

<members>

<rules; separator="\n\n">

<! generate rule/method definitions for imported rules so they
appear to be defined in this recognizer. !>
// Delegated rules
<grammar.delegatedRules:{ruleDescriptor|
    public function <ruleDescriptor.name>(<ruleDescriptor.parameterScope:parameterScope()>):<returnType()> {
<if(ruleDescriptor.hasReturnValue)>return
<endif><ruleDescriptor.grammar:delegateName()>.<ruleDescriptor.name>(<ruleDescriptor.parameterScope.attribut
es:{a|<a.name>; separator=", ">; \}); separator="\n">

    <synpreds:{p | <synpred(p)>}>

    <cyclicDFAs:cyclicDFA()> <! dump tables for all DFA !>

    <bitsets:{it | <bitset(name={FOLLOW_<it.name>_in_<it.inName><it.tokenIndex>},
        words64=it.bits)>}>
    }
    >>

parserCtorBody() ::= <<
<if(memoize)>
<if(grammar.grammarIsRoot)>
this.state.ruleMemo = new Array(<length(grammar.allImportedRules)>+1);<\n> <! index from 1..n !>
<endif>
<endif>
<grammar.delegators:

```

```

{g|this.<g:delegateName()> = <g:delegateName()>; separator="\n">
>>

parser(grammar, name, scopes, tokens, tokenNames, rules, numRules, bitsets, ASTLabelType="Object",
superClass="Parser", labelType="Token", members={<actions.parser.members>}) ::= <<
<genericParser(grammar, name, scopes, tokens, tokenNames, rules, numRules,
    bitsets, "TokenStream", superClass,
    labelType, members, "Token",
    false, ASTLabelType)>
>>

/** How to generate a tree parser; same as parser except the input
 * stream is a different type.
 */
treeParser(grammar, name, scopes, tokens, tokenNames, globalAction, rules,
    numRules, bitsets, filterMode, labelType={<ASTLabelType>}, ASTLabelType="Object",
    superClass="TreeParser", members={<actions.treeparser.members>}) ::= <<
<genericParser(grammar, name, scopes, tokens, tokenNames, rules, numRules,
    bitsets, "TreeNodeStream", superClass,
    labelType, members, "Node",
    filterMode, ASTLabelType)>
>>

/** A simpler version of a rule template that is specific to the imaginary
 * rules created for syntactic predicates. As they never have return values
 * nor parameters etc..., just give simplest possible method. Don't do
 * any of the normal memoization stuff in here either; it's a waste.
 * As predicates cannot be inlined into the invoking rule, they need to
 * be in a rule by themselves.
 */
synpredRule(ruleName, ruleDescriptor, block, description, nakedBlock) ::=
<<
// $ANTLR start <ruleName>
public final function <ruleName>_fragment(<ruleDescriptor.parameterScope:parameterScope()>):void {
    <ruleLabelDefs()>
<if(trace)>
    traceIn("<ruleName>_fragment", <ruleDescriptor.index>);
    try {
        <block>
    }
    finally {
        traceOut("<ruleName>_fragment", <ruleDescriptor.index>);
    }
<else>
    <block>
<endif>
}
// $ANTLR end <ruleName>

```

>>

```
synpred(name) ::= <<
public final function <name>():Boolean {
    this.state.backtracking++;
    <@start()>
    var start:int = input.mark();
    try {
        <name>_fragment(); // can never throw exception
    } catch (re:RecognitionException) {
        trace("impossible: "+re);
    }
    var success:Boolean = !this.state.failed;
    input.rewindTo(start);
    <@stop()>
    this.state.backtracking--;
    this.state.failed=false;
    return success;
}<\n>
```

>>

```
lexerSynpred(name) ::= <<
<synpred(name)>
```

>>

```
ruleMemoization(name) ::= <<
<if(memoize)>
if ( this.state.backtracking>0 && alreadyParsedRule(input, <ruleDescriptor.index>) ) { return <ruleReturnValue()>;
}
<endif>
```

>>

```
/** How to test for failure and return from rule */
checkRuleBacktrackFailure() ::= <<
<if(backtracking)>if (this.state.failed) return <ruleReturnValue()>;<endif>
```

>>

```
/** This rule has failed, exit indicating failure during backtrack */
ruleBacktrackFailure() ::= <<
<if(backtracking)>if (this.state.backtracking>0) {this.state.failed=true; return <ruleReturnValue()>;}<endif>
```

>>

```
/** How to generate code for a rule. This includes any return type
* data aggregates required for multiple return values.
*/
rule(ruleName,ruleDescriptor,block,emptyRule,description,exceptions,finally,memoize) ::= <<
<ruleAttributeScope(scope=ruleDescriptor.ruleScope)>
// $ANTLR start <ruleName>
```

```

// <fileName>:<description>
public final function <ruleName>(<ruleDescriptor.parameterScope:parameterScope()>:<returnType()> {
    <if(trace)>traceIn("<ruleName>", <ruleDescriptor.index>);<endif>
    <ruleScopeSetUp()>
    <ruleDeclarations()>
    <ruleLabelDefs()>
    <ruleDescriptor.actions.init>
    <@preamble()>
    try {
        <ruleMemoization(name=ruleName)>
        <block>
        <ruleCleanUp()>
        <(ruleDescriptor.actions.after):execAction()>
    }
    <if(exceptions)>
        <exceptions:{e|<catch(decl=e.decl,action=e.action)><\n>}>
    <else>
    <if(!emptyRule)>
    <if(actions.(actionScope).rulecatch)>
        <actions.(actionScope).rulecatch>
    <else>
        catch (re:RecognitionException) {
            reportError(re);
            recoverStream(input,re);
            <@setErrorReturnValue()>
        }<\n>
    <endif>
    <endif>
    <endif>
    finally {
        <if(trace)>traceOut("<ruleName>", <ruleDescriptor.index>);<endif>
        <memoize()>
        <ruleScopeCleanUp()>
        <finally>
    }
    <@postamble()>
    return <ruleReturnValue()>;
}
// $ANTLR end <ruleName>
>>

catch(decl,action) ::= <<
catch (<e.decl>) {
    <e.action>
}
>>

ruleDeclarations() ::= <<

```

```

<if(ruleDescriptor.hasMultipleReturnValues)>
var retval:<returnType()> = new <returnType()>();
retval.start = input.LT(1);<\n>
<else>
<ruleDescriptor.returnScope.attributes:{ a |
var <a.name>:<a.type> = <if(a.initValue)><a.initValue><else><initValue(a.type)><endif>;
}>
<endif>
<if(memoize)>
var <ruleDescriptor.name>_startIndex:int = input.index;
<endif>
>>

ruleScopeSetUp() ::= <<
<ruleDescriptor.useScopes:{it |<it>_stack.push(new Object());}; separator="\n">
<ruleDescriptor.ruleScope:{it |<it.name>_stack.push(new Object());}; separator="\n">
>>

ruleScopeCleanUp() ::= <<
<ruleDescriptor.useScopes:{it |<it>_stack.pop();}; separator="\n">
<ruleDescriptor.ruleScope:{it |<it.name>_stack.pop();}; separator="\n">
>>

ruleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,ruleDescriptor.tokenListLabels,
ruleDescriptor.wildcardTreeLabels,ruleDescriptor.wildcardTreeListLabels]
: {it |var <it.label.text>:<labelType>=null;}; separator="\n"
>
<[ruleDescriptor.tokenListLabels,ruleDescriptor.ruleListLabels,ruleDescriptor.wildcardTreeListLabels]
: {it |var list_<it.label.text>:Array=null;}; separator="\n"
>
<ruleDescriptor.ruleLabels:ruleLabelDef(); separator="\n">
<ruleDescriptor.ruleListLabels:{ll|var <ll.label.text>:RuleReturnScope = null;}; separator="\n">
>>

lexerRuleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,
ruleDescriptor.tokenListLabels,
ruleDescriptor.ruleLabels]
: {it |var <it.label.text>:<labelType>=null;}; separator="\n"
>
<ruleDescriptor.charLabels:{it |var <it.label.text>:int;}; separator="\n">
<[ruleDescriptor.tokenListLabels,
ruleDescriptor.ruleListLabels]
: {it |var list_<it.label.text>:Array=null;}; separator="\n"
>
>>

```

```

ruleReturnValue() ::= <%
<if(!ruleDescriptor.isSynPred)>
<if(ruleDescriptor.hasReturnValue)>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnName>
<else>
retval
<endif>
<endif>
<endif>
%>

ruleCleanUp() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(!TREE_PARSER)>
retval.stop = input.LT(-1);<\n>
<endif>
<endif>
>>

memoize() ::= <<
<if(memoize)>
<if(backtracking)>
if ( this.state.backtracking>0 ) { memoize(input, <ruleDescriptor.index>, <ruleDescriptor.name>_StartIndex); }
<endif>
<endif>
>>

/** How to generate a rule in the lexer; naked blocks are used for
 * fragment rules.
 */
lexerRule(ruleName,nakedBlock,ruleDescriptor,block,memoize) ::= <<
// $ANTLR start <ruleName>
public final function m<ruleName>(<ruleDescriptor.parameterScope:parameterScope()):void {
  <if(trace)>traceIn("<ruleName>", <ruleDescriptor.index>);<endif>
  <ruleScopeSetUp()>
  <ruleDeclarations()>
  try {
<if(nakedBlock)>
    <ruleMemoization(name=ruleName)>
    <lexerRuleLabelDefs()>
    <ruleDescriptor.actions.init>
    <block><\n>
<else>
    var _type:int = <ruleName>;
    var _channel:int = DEFAULT_TOKEN_CHANNEL;
    <ruleMemoization(name=ruleName)>
    <lexerRuleLabelDefs()>

```

```

    <ruleDescriptor.actions.init>
    <block>
    <ruleCleanUp()>
    this.state.type = _type;
    this.state.channel = _channel;
    <(ruleDescriptor.actions.after):execAction()>
<endif>
}
finally {
    <if(trace)>traceOut("<ruleName>", <ruleDescriptor.index>);<endif>
    <ruleScopeCleanUp()>
    <memoize()>
}
}
// $ANTLR end <ruleName>
>>

```

```

/** How to generate code for the implicitly-defined lexer grammar rule
 * that chooses between lexer rules.
 */

```

```

tokensRule(ruleName,nakedBlock,args,block,ruleDescriptor) ::= <<
public override function mTokens():void {
    <block><\n>
}
>>

```

```

// S U B R U L E S

```

```

/** A (...) subrule with multiple alternatives */

```

```

block(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
// <fileName>:<description>
var alt<decisionNumber>:int=<maxAlt>;
<decls>
<@predecision()>
<decision>
<@postdecision()>
<@prebranch()>
switch (alt<decisionNumber>) {
    <alts:{a | <altSwitchCase(i, a)>}>
}
<@postbranch()>
>>

```

```

/** A rule block with multiple alternatives */

```

```

ruleBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
// <fileName>:<description>
var alt<decisionNumber>:int=<maxAlt>;
<decls>

```

```

<@predecision()>
<decision>
<@postdecision()>
switch (alt<decisionNumber>) {
  <alts:{ a | <altSwitchCase(i, a)>}>
}
>>

ruleBlockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
// <fileName>:<description>
<decls>
<@prealt()>
<alts>
<@postalt()>
>>

/** A special case of a (...) subrule with a single alternative */
blockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
// <fileName>:<description>
<decls>
<@prealt()>
<alts>
<@postalt()>
>>

/** A (..)+ block with 1 or more alternatives */
positiveClosureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
// <fileName>:<description>
var cnt<decisionNumber>:int=0;
<decls>
<@preloop()>
loop<decisionNumber>:
do {
  var alt<decisionNumber>:int=<maxAlt>;
  <@predecision()>
  <decision>
  <@postdecision()>
  switch (alt<decisionNumber>) {
  <alts:{ a | <altSwitchCase(i, a)>}>
  default :
    if ( cnt<decisionNumber> >= 1 ) break loop<decisionNumber>;
    <ruleBacktrackFailure()>
    throw new EarlyExitException(<decisionNumber>, input);
    <! Need to add support for earlyExitException debug hook !>
  }
  cnt<decisionNumber>++;
} while (true);

```

```

<@postloop()>
>>

positiveClosureBlockSingleAlt ::= positiveClosureBlock

/** A (..)* block with 1 or more alternatives */
closureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::=
<<
// <fileName>:<description>
<decls>
<@preloop()>
loop<decisionNumber>:
do {
    var alt<decisionNumber>:int=<maxAlt>;
    <@predecision()>
    <decision>
    <@postdecision()>
    switch (alt<decisionNumber>) {
    <alts:{a | <altSwitchCase(i, a)>}>
    default :
        break loop<decisionNumber>;
    }
} while (true);
<@postloop()>
>>

closureBlockSingleAlt ::= closureBlock

/** Optional blocks (x)? are translated to (x|) by before code generation
 * so we can just use the normal block template
 */
optionalBlock ::= block

optionalBlockSingleAlt ::= block

/** A case in a switch that jumps to an alternative given the alternative
 * number. A DFA predicts the alternative and then a simple switch
 * does the jump to the code that actually matches that alternative.
 */
altSwitchCase(altNum, alt) ::= <<
case <altNum> :
    <@prealt()>
    <alt>
    break;<\n>
>>

/** An alternative is just a list of elements; at outermost level */
alt(elements,altNum,description,autoAST,outerAlt,treeLevel,rew) ::= <<

```

```

// <fileName>:<description>
{
<@declarations()>
<elements:element()>
<rew>
<@cleanup()>
}
>>

/** What to emit when there is no rewrite. For auto build
 * mode, does nothing.
 */
noRewrite(rewriteBlockLevel, treeLevel) ::= ""

// E L E M E N T S

/** Dump the elements one per line */
element(e) ::= <<
<@prematch()>
<e.el><\n>
>>

/** match a token optionally with a label in front */
tokenRef(token,label,elementIndex,terminalOptions) ::= <<
<if(label)><label>=<labelType><(<endif>matchStream(input,<token>,FOLLOW_<token>_in_<ruleName><element
Index>)<if(label)><endif>; <checkRuleBacktrackFailure()>
>>

/** ids+=ID */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<tokenRef(token,label,elementIndex,terminalOptions)>
<listLabel(label, label)>
>>

listLabel(label,elem) ::= <<
if (list_<label>==null) list_<label>=new Array();
list_<label>.push(<elem>);<\n>
>>

/** match a character */
charRef(char,label) ::= <<
<if(label)>
<label> = input.LA(1);<\n>
<endif>
match(<char>); <checkRuleBacktrackFailure()>
>>

/** match a character range */

```

```

charRangeRef(a,b,label) ::= <<
<if(label)>
<label> = input.LA(1);<\n>
<endif>
matchRange(<a>,<b>); <checkRuleBacktrackFailure()>
>>

/** For now, sets are interval tests and must be tested inline */
matchSet(s,label,elementIndex,terminalOptions,postmatchCode="") ::= <<
<if(label)>
<if(LEXER)>
<label>= input.LA(1);<\n>
<else>
<label>=<labelType>(input.LT(1));<\n>
<endif>
<endif>
if ( <s> ) {
    input.consume();
    <postmatchCode>
<if(!LEXER)>
    this.state.errorRecovery=false;
<endif>
    <if(backtracking)>this.state.failed=false;<endif>
}
else {
    <ruleBacktrackFailure()>
    <@mismatchedSetException()>
<if(LEXER)>
    throw recover(new MismatchedSetException(null,input));<\n>
<else>
    throw new MismatchedSetException(null,input);
    <! use following code to make it recover inline; remove throw mse;
    recoverFromMismatchedSet(input,mse,FOLLOW_set_in_<ruleName><elementIndex>);
    !>
<endif>
}<\n>
>>

matchRuleBlockSet ::= matchSet

matchSetAndListLabel(s,label,elementIndex,postmatchCode) ::= <<
<matchSet(...)>
<listLabel(label, label)>
>>

/** Match a string literal */
lexerStringRef(string,label,elementIndex="0") ::= <<
<if(label)>

```

```

var <label>Start:int = charIndex;
matchString(<string>); <checkRuleBacktrackFailure()>
var <label>StartLine<elementIndex>:int = line;
var <label>StartCharPos<elementIndex>:int = charPositionInLine;
<label> = CommonToken.createFromStream(input, TokenConstants.INVALID_TOKEN_TYPE,
TokenConstants.DEFAULT_CHANNEL, <label>Start, charIndex-1);
<label>.line = <label>StartLine<elementIndex>;
<label>.charPositionInLine = <label>StartCharPos<elementIndex>;
<else>
matchString(<string>); <checkRuleBacktrackFailure()><\n>
<endif>
>>

wildcard(token,label,elementIndex,terminalOptions) ::= <<
<if(label)>
<label>=<labelType>(input.LT(1));<\n>
<endif>
matchAny(input); <checkRuleBacktrackFailure()>
>>

wildcardAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<wildcard(...)>
<listLabel(label, label)>
>>

/** Match . wildcard in lexer */
wildcardChar(label, elementIndex) ::= <<
<if(label)>
<label> = input.LA(1);<\n>
<endif>
matchAny(); <checkRuleBacktrackFailure()>
>>

wildcardCharListLabel(label, elementIndex) ::= <<
<wildcardChar(label, elementIndex)>
<listLabel(label, label)>
>>

/** Match a rule reference by invoking it possibly with arguments
 * and a return value or values. The 'rule' argument was the
 * target rule name, but now is type Rule, whose toString is
 * same: the rule name. Now though you can access full rule
 * descriptor stuff.
 *
 * GMS: Note: do not use post-decrement operator! ASC produces bad code for exceptions in this case.
 * See: https://bugs.adobe.com/jira/browse/ASC-3625
 */
ruleRef(rule,label,elementIndex,args,scope) ::= <<

```

```

pushFollow(FOLLOW_<rule.name>_in_<ruleName><elementIndex>);
<if(label)><label>=<endif><if(scope)><scope:delegateName()>.<endif><rule.name>(<args; separator=", ">);<\n>
state._fsp = state._fsp - 1;
<checkRuleBacktrackFailure()>
>>

```

```

/** ids+=r */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(rule,label,elementIndex,args,scope)>
<listLabel(label, label)>
>>

```

```

/** A lexer rule reference.

```

```

*
* The 'rule' argument was the target rule name, but now
* is type Rule, whose toString is same: the rule name.
* Now though you can access full rule descriptor stuff.
*/

```

```

lexerRuleRef(rule,label,args,elementIndex,scope) ::= <<
<if(label)>
var <label>Start<elementIndex>:int = charIndex;
var <label>StartLine<elementIndex>:int = line;
var <label>StartCharPos<elementIndex>:int = charPositionInLine;
<if(scope)><scope:delegateName()>.<endif>m<rule.name>(<args; separator=", ">);
<checkRuleBacktrackFailure()>
<label> = CommonToken.createFromStream(input, TokenConstants.INVALID_TOKEN_TYPE,
TokenConstants.DEFAULT_CHANNEL, <label>Start<elementIndex>, charIndex-1);
<label>.line = <label>StartLine<elementIndex>;
<label>.charPositionInLine = <label>StartCharPos<elementIndex>;
<else>
<if(scope)><scope:delegateName()>.<endif>m<rule.name>(<args; separator=", ">);
<checkRuleBacktrackFailure()>
<endif>
>>

```

```

/** i+=INT in lexer */
lexerRuleRefAndListLabel(rule,label,args,elementIndex,scope) ::= <<
<lexerRuleRef(rule,label,args,elementIndex,scope)>
<listLabel(label, label)>
>>

```

```

/** EOF in the lexer */

```

```

lexerMatchEOF(label,elementIndex) ::= <<
<if(label)>
var <label>Start<elementIndex>:int = charIndex;
var <label>StartLine<elementIndex>:int = line;
var <label>StartCharPos<elementIndex>:int = charPositionInLine;

```

```

match(EOF); <checkRuleBacktrackFailure()>
var <label>:<labelType> = CommonToken.createFromStream(input, EOF,
TokenConstants.DEFAULT_CHANNEL, <label>Start<elementIndex>, charIndex-1);
<label>.line = <label>StartLine<elementIndex>;
<label>.charPositionInLine = <label>StartCharPos<elementIndex>;
<else>
match(EOF); <checkRuleBacktrackFailure()>
<endif>
>>

// used for left-recursive rules
recRuleDefArg()          ::= "var <recRuleArg()>:int"
recRuleArg()            ::= "_p"
recRuleAltPredicate(ruleName,opPrec) ::= "<recRuleArg()> \<= <opPrec>"
recRuleSetResultAction() ::= "root_0=$<ruleName>_primary.tree;"
recRuleSetReturnAction(src,name)    ::= "$<name>=$<src>.<name>;"

/** match ^(root children) in tree parser */
tree(root, actionsAfterRoot, children, nullableChildList,
enclosingTreeLevel, treeLevel) ::= <<
<root:element()>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if ( input.LA(1)==TokenConstants.DOWN ) {
matchStream(input, TokenConstants.DOWN, null); <checkRuleBacktrackFailure()>
<children:element()>
matchStream(input, TokenConstants.UP, null); <checkRuleBacktrackFailure()>
}
<else>
matchStream(input, TokenConstants.DOWN, null); <checkRuleBacktrackFailure()>
<children:element()>
matchStream(input, TokenConstants.UP, null); <checkRuleBacktrackFailure()>
<endif>
>>

/** Every predicate is used as a validating predicate (even when it is
* also hoisted into a prediction expression).
*/
validateSemanticPredicate(pred,description) ::= <<
if ( !(<evalPredicate(pred,description)>) ) {
<ruleBacktrackFailure()>
throw new FailedPredicateException(input, "<ruleName>", "<description>");
}
>>

// F i x e d D F A (if-then-else)

dfaState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<

```

```

var LA<decisionNumber>_<stateNumber>:int = input.LA(<k>);<\n>
<edges; separator="\nelse ">
else {
<if(eotPredictsAlt)>
  alt<decisionNumber>=<eotPredictsAlt>;
<else>
  <ruleBacktrackFailure()>
  throw new NoViableAltException("<description>", <decisionNumber>, <stateNumber>, input);<\n>
  <! Need to add hook for noViableAltException() !>
<endif>
}
>>

```

```

/** Same as a normal DFA state except that we don't examine lookahead
 * for the bypass alternative. It delays error detection but this
 * is faster, smaller, and more what people expect. For (X)? people
 * expect "if ( LA(1)==X ) match(X);" and that's it.
 */
dfaOptionalBlockState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
var LA<decisionNumber>_<stateNumber>:int = input.LA(<k>);<\n>
<edges; separator="\nelse ">
>>

```

```

/** A DFA state that is actually the loopback decision of a closure
 * loop. If end-of-token (EOT) predicts any of the targets then it
 * should act like a default clause (i.e., no error can be generated).
 * This is used only in the lexer so that for ('a')* on the end of a rule
 * anything other than 'a' predicts exiting.
 */
dfaLoopbackState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
var LA<decisionNumber>_<stateNumber>:int = input.LA(<k>);<\n>
<edges; separator="\nelse "><\n>
<if(eotPredictsAlt)>
<if(!edges)>
alt<decisionNumber>=<eotPredictsAlt>; <! if no edges, don't gen ELSE !>
<else>
else {
  alt<decisionNumber>=<eotPredictsAlt>;
}<\n>
<endif>
<endif>
>>

```

```

/** An accept state indicates a unique alternative has been predicted */
dfaAcceptState(alt) ::= "alt<decisionNumber>=<alt>";

```

```

/** A simple edge with an expression. If the expression is satisfied,
 * enter to the target state. To handle gated productions, we may

```

```

* have to evaluate some predicates for this edge.
*/
dfaEdge(labelExpr, targetState, predicates) ::= <<
if ( (<labelExpr> <if(predicates)>&& (<predicates>)<endif> ) {
    <targetState>
}
>>

// F i x e d D F A (switch case)

/** A DFA state where a SWITCH may be generated. The code generator
* decides if this is possible: CodeGenerator.canGenerateSwitch().
*/
dfaStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ( input.LA(<k>) ) {
<edges; separator="\n">
default:
<if(eotPredictsAlt)>
    alt<decisionNumber>=<eotPredictsAlt>;
<else>
    <ruleBacktrackFailure()>
    throw new NoViableAltException("<description>", <decisionNumber>, <stateNumber>, input);<\n>
    <! Need to add hook for noViableAltException !>
<endif>
}<\n>
>>

dfaOptionalBlockStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ( input.LA(<k>) ) {
    <edges; separator="\n">
}<\n>
>>

dfaLoopbackStateSwitch(k, edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ( input.LA(<k>) ) {
<edges; separator="\n"><\n>
<if(eotPredictsAlt)>
default:
    alt<decisionNumber>=<eotPredictsAlt>;
    break;<\n>
<endif>
}<\n>
>>

dfaEdgeSwitch(labels, targetState) ::= <<
<labels: {it |case <it>;}; separator="\n">
{
    <targetState>
}
>>

```

```

    }
    break;
>>

// C y c l i c D F A

/** The code to initiate execution of a cyclic DFA; this is used
 * in the rule to predict an alt just like the fixed DFA case.
 * The <name> attribute is inherited via the parser, lexer, ...
 */
dfaDecision(decisionNumber,description) ::= <<
alt<decisionNumber> = dfa<decisionNumber>.predict(input);
>>

cyclicDFActor(dfa) ::= <<

dfa<dfa.decisionNumber> = new DFA(this, <dfa.decisionNumber>,
    "<dfa.description>",
    DFA<dfa.decisionNumber>_eot, DFA<dfa.decisionNumber>_eof, DFA<dfa.decisionNumber>_min,
    DFA<dfa.decisionNumber>_max, DFA<dfa.decisionNumber>_accept, DFA<dfa.decisionNumber>_special,
    DFA<dfa.decisionNumber>_transition<if(dfa.specialStateSTs)>,
    DFA<dfa.decisionNumber>_specialStateTransition<endif>);

>>

/* Dump DFA tables as run-length-encoded Strings of octal values.
 * Can't use hex as compiler translates them before compilation.
 * These strings are split into multiple, concatenated strings.
 * Java puts them back together at compile time thankfully.
 * Java cannot handle large static arrays, so we're stuck with this
 * encode/decode approach. See analysis and runtime DFA for
 * the encoding methods.
 */
cyclicDFA(dfa) ::= <<

private const DFA<dfa.decisionNumber>_eot:Array =
    DFA.unpackEncodedString("<dfa.javaCompressedEOT; wrap=\""+\n  \>");
private const DFA<dfa.decisionNumber>_eof:Array =
    DFA.unpackEncodedString("<dfa.javaCompressedEOF; wrap=\""+\n  \>");
private const DFA<dfa.decisionNumber>_min:Array =
    DFA.unpackEncodedString("<dfa.javaCompressedMin; wrap=\""+\n  \>", true);
private const DFA<dfa.decisionNumber>_max:Array =
    DFA.unpackEncodedString("<dfa.javaCompressedMax; wrap=\""+\n  \>", true);
private const DFA<dfa.decisionNumber>_accept:Array =
    DFA.unpackEncodedString("<dfa.javaCompressedAccept; wrap=\""+\n  \>");
private const DFA<dfa.decisionNumber>_special:Array =
    DFA.unpackEncodedString("<dfa.javaCompressedSpecial; wrap=\""+\n  \>");
private const DFA<dfa.decisionNumber>_transition:Array = [
    <dfa.javaCompressedTransition: {s|DFA.unpackEncodedString("<s; wrap=\""+\n\>"); separator=",\n">

```

```

];
<if(dfa.specialStateSTs)>
  private function DFA<dfa.decisionNumber>_specialStateTransition(dfa:DFA, s:int, _input:IntStream):int {
    <if(LEXER)>
      var input:IntStream = _input;
    <endif>
    <if(PARSER)>
      var input:TokenStream = TokenStream(_input);
    <endif>
    <if(TREE_PARSER)>
      var input:TreeNodeStream = TreeNodeStream(_input);
    <endif>
    var _s:int = s;
    switch ( s ) {
      <dfa.specialStateSTs:{state |
        case <i0> : <! compressed special state numbers 0..n-1 !>
          <state>}; separator="\n">
    }
  <if(backtracking)>
    if (this.state.backtracking>0) {this.state.failed=true; return -1;}<\n>
  <endif>
    throw dfa.error(new NoViableAltException(dfa.description, <dfa.decisionNumber>, _s, input));
  }<\n>
  <endif>

protected var dfa<dfa.decisionNumber>:DFA; // initialized in constructor

>>

/** A state in a cyclic DFA; it's a special state and part of a big switch on
 * state.
 */
cyclicDFAState(decisionNumber,stateNumber,edges,needErrorClause,semPredState) ::= <<
var LA<decisionNumber>_<stateNumber>:int = input.LA(1);<\n>
<if(semPredState)> <! get next lookahead symbol to test edges, then rewind !>
var index<decisionNumber>_<stateNumber>:int = input.index;
input.rewind();<\n>
<endif>
s = -1;
<edges; separator="\nelse ">
<if(semPredState)> <! return input cursor to state before we rewound !>
input.seek(index<decisionNumber>_<stateNumber>);<\n>
<endif>
if ( s>=0 ) return s;
break;
>>

/** Just like a fixed DFA edge, test the lookahead and indicate what

```

```

* state to jump to next if successful.
*/
cyclicDFAEdge(labelExpr, targetStateNumber, edgeNumber, predicates) ::= <<
if ( (<labelExpr> <if(predicates)>&& (<predicates>)<endif>) { s = <targetStateNumber>; }<\n>
>>

```

```

/** An edge pointing at end-of-token; essentially matches any char;
* always jump to the target.
*/
eotDFAEdge(targetStateNumber,edgeNumber, predicates) ::= <<
s = <targetStateNumber>;<\n>
>>

```

// D F A E X P R E S S I O N S

```
andPredicates(left,right) ::= "<left>&&<right>"
```

```
orPredicates(operands) ::= "<operands; separator='||'>"
```

```
notPredicate(pred) ::= "!(<evalPredicate(pred, { })>)"
```

```
evalPredicate(pred,description) ::= "<pred>"
```

```
evalSynPredicate(pred,description) ::= "<pred>()"
```

```
lookaheadTest(atom,k,atomAsInt) ::= "LA<decisionNumber>_<stateNumber>==<atomAsInt>"
```

```

/** Sometimes a lookahead test cannot assume that LA(k) is in a temp variable
* somewhere. Must ask for the lookahead directly.
*/

```

```
isolatedLookaheadTest(atom,k,atomAsInt) ::= "input.LA(<k>)==<atomAsInt>"
```

```

lookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= <%
(LA<decisionNumber>_<stateNumber> >= <lowerAsInt> && LA<decisionNumber>_<stateNumber> \<=
<upperAsInt>)
%>

```

```
isolatedLookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= "(input.LA(<k>) >=
<lowerAsInt> && input.LA(<k>) \<= <upperAsInt>)"
```

```

setTest(ranges) ::= <<
<ranges; separator="||">
>>

```

// A T T R I B U T E S

```
globalAttributeScope(scope) ::= <<
```

```

<if(scope.attributes)>
protected var <scope.name>_stack:Array = new Array();<\n>
<endif>
>>

ruleAttributeScope(scope) ::= <<
<if(scope.attributes)>
protected var <scope.name>_stack:Array = new Array();<\n>
<endif>
>>

returnStructName() ::= "<if(TREE_PARSER)>Tree<else>Parser<endif>RuleReturnScope"

returnType() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<returnStructName()>
<else>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnType>
<else>
void
<endif>
<endif>
>>

/** Generate the Java type associated with a single or multiple return
 * values.
 */
ruleLabelType(referencedRule) ::= <<
<if(referencedRule.hasMultipleReturnValues)>
<returnStructName()>
<else>
<if(referencedRule.hasSingleReturnValue)>
<referencedRule.singleValueReturnType>
<else>
void
<endif>
<endif>
>>

delegateName(d) ::= <<
<if(d.label)><d.label><else>g<d.name><endif>
>>

/** Using a type to init value map, try to init a type; if not in table
 * must be an object, default value is "null".
 */
initValue(typeName) ::= <<

```

```

<asTypeInitMap.(typeName)>
>>

/** Define a rule label including default value */
ruleLabelDef(label) ::= <<
var <label.label.text>:<ruleLabelType(referencedRule=label.referencedRule)> =
<initValue(typeName=ruleLabelType(referencedRule=label.referencedRule))>;<\n>
>>

/** Define a return struct for a rule if the code needs to access its
 * start/stop tokens, tree stuff, attributes, ... Leave a hole for
 * subgroups to stick in members.
 */
returnScope(scope) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
public static class <returnType()> extends <if(TREE_PARSER)>Tree<else>Parser<endif>RuleReturnScope {
  <scope.attributes:{it |public <it.decl>;}; separator="\n">
  <@ruleReturnMembers()>
};
<endif>
>>

parameterScope(scope) ::= <<
<scope.attributes:{it |<it.name>:<it.type>;}; separator=" ">
>>

parameterAttributeRef(attr) ::= "<attr.name>"
parameterSetAttributeRef(attr,expr) ::= "<attr.name> =<expr>;"

scopeAttributeRef(scope,attr,index,negIndex) ::= <<
<if(negIndex)>
<scope>_stack[<scope>_stack.length-<negIndex>-1].<attr.name>
<else>
<if(index)>
<scope>_stack[<index>].<attr.name>
<else>
<scope>_stack[<scope>_stack.length-1].<attr.name>
<endif>
<endif>
>>

scopeSetAttributeRef(scope,attr,expr,index,negIndex) ::= <<
<if(negIndex)>
<scope>_stack[<scope>_stack.length-<negIndex>-1].<attr.name> =<expr>;
<else>
<if(index)>
<scope>_stack[<index>].<attr.name> =<expr>;
<else>

```

```

<scope>_stack[<scope>_stack.length-1].<attr.name> =<expr>;
<endif>
<endif>
>>

/** $x is either global scope or x is rule with dynamic scope; refers
 * to stack itself not top of stack. This is useful for predicates
 * like {$function.size()>0 && $function::name.equals("foo")}?
 */
isolatedDynamicScopeRef(scope) ::= "<scope>_stack"

/** reference an attribute of rule; might only have single return value */
ruleLabelRef(referencedRule,scope,attr) ::= <<
<if(referencedRule.hasMultipleReturnValues)>
(<scope>!=null?<scope>.values.<attr.name>:<initValue(attr.type)>)
<else>
<scope>
<endif>
>>

returnAttributeRef(ruleDescriptor,attr) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.values.<attr.name>
<else>
<attr.name>
<endif>
>>

returnSetAttributeRef(ruleDescriptor,attr,expr) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.values.<attr.name> =<expr>;
<else>
<attr.name> =<expr>;
<endif>
>>

/** How to translate $tokenLabel */
tokenLabelRef(label) ::= "<label>"

/** ids+=ID {$ids} or e+=expr {$e} */
listLabelRef(label) ::= "list_<label>"

// not sure the next are the right approach

tokenLabelPropertyRef_text(scope,attr) ::= "(<scope>!=null?<scope>.text:null)"
tokenLabelPropertyRef_type(scope,attr) ::= "(<scope>!=null?<scope>.type:0)"
tokenLabelPropertyRef_line(scope,attr) ::= "(<scope>!=null?<scope>.line:0)"

```

```

tokenLabelPropertyRef_pos(scope,attr) ::= "<scope>!=null?<scope>.charPositionInLine:0)"
tokenLabelPropertyRef_channel(scope,attr) ::= "<scope>!=null?<scope>.channel:0)"
tokenLabelPropertyRef_index(scope,attr) ::= "<scope>!=null?<scope>.tokenIndex:0)"
tokenLabelPropertyRef_tree(scope,attr) ::= "<scope>_tree"
tokenLabelPropertyRef_int(scope,attr) ::= "<scope>!=null?int(<scope>.text):0)"

ruleLabelPropertyRef_start(scope,attr) ::= "<scope>!=null?<labelType>(<scope>.start):null)"
ruleLabelPropertyRef_stop(scope,attr) ::= "<scope>!=null?<labelType>(<scope>.stop):null)"
ruleLabelPropertyRef_tree(scope,attr) ::= "<scope>!=null?<ASTLabelType>(<scope>.tree):null)"
ruleLabelPropertyRef_text(scope,attr) ::= <<
<if(TREE_PARSER)>
(<scope>!=null?(input.tokenStream.toStringWithRange(
input.treeAdaptor.getTokenStartIndex(<scope>.start),
input.treeAdaptor.getTokenStopIndex(<scope>.start))):null)
<else>
(<scope>!=null?input.toStringWithTokenRange(<scope>.start,<scope>.stop):null)
<endif>
>>

ruleLabelPropertyRef_st(scope,attr) ::= "<scope>!=null?<scope>.st:null)"

/** Isolated $RULE ref ok in lexer as it's a Token */
lexerRuleLabel(label) ::= "<label>"

lexerRuleLabelPropertyRef_type(scope,attr) ::=
"<scope>!=null?<scope>.type:0)"
lexerRuleLabelPropertyRef_line(scope,attr) ::=
"<scope>!=null?<scope>.lien:0)"
lexerRuleLabelPropertyRef_pos(scope,attr) ::=
"<scope>!=null?<scope>.charPositionInLine:0)"
lexerRuleLabelPropertyRef_channel(scope,attr) ::=
"<scope>!=null?<scope>.channel:0)"
lexerRuleLabelPropertyRef_index(scope,attr) ::=
"<scope>!=null?<scope>.tokenIndex:0)"
lexerRuleLabelPropertyRef_text(scope,attr) ::=
"<scope>!=null?<scope>.text:null)"
lexerRuleLabelPropertyRef_int(scope,attr) ::=
"<scope>!=null?int(<scope>.text):0)"

// Somebody may ref $template or $tree or $stop within a rule:
rulePropertyRef_start(scope,attr) ::= "<labelType>(retval.start)"
rulePropertyRef_stop(scope,attr) ::= "<labelType>(retval.stop)"
rulePropertyRef_tree(scope,attr) ::= "<ASTLabelType>(retval.tree)"
rulePropertyRef_text(scope,attr) ::= <<
<if(TREE_PARSER)>
input.tokenStream.toStringWithRange(
input.treeAdaptor.getTokenStartIndex(retval.start),
input.treeAdaptor.getTokenStopIndex(retval.start))

```

```

<else>
input.toStringWithTokenRange(retval.start,input.LT(-1))
<endif>
>>
rulePropertyRef_st(scope,attr) ::= "retval.st"

lexerRulePropertyRef_text(scope,attr) ::= "text"
lexerRulePropertyRef_type(scope,attr) ::= "_type"
lexerRulePropertyRef_line(scope,attr) ::= "state.tokenStartLine"
lexerRulePropertyRef_pos(scope,attr) ::= "state.tokenStartCharPositionInLine"
lexerRulePropertyRef_index(scope,attr) ::= "-1" // undefined token index in lexer
lexerRulePropertyRef_channel(scope,attr) ::= "_channel"
lexerRulePropertyRef_start(scope,attr) ::= "state.tokenStartCharIndex"
lexerRulePropertyRef_stop(scope,attr) ::= "(charIndex-1)"
lexerRulePropertyRef_int(scope,attr) ::= "int(<scope>.text)"

// setting $st and $tree is allowed in local rule. everything else
// is flagged as error
ruleSetPropertyRef_tree(scope,attr,expr) ::= "retval.tree =<expr>";
ruleSetPropertyRef_st(scope,attr,expr) ::= "retval.st =<expr>";

/** How to execute an action (only when not backtracking) */
execAction(action) ::= <<
<if(backtracking)>
if ( <actions.(actionScope).synpredgate> ) {
  <action>
}
<else>
<action>
<endif>
>>

/** How to always execute an action even when backtracking */
execForcedAction(action) ::= "<action>"

// M I S C (properties, etc...)

bitset(name, words64) ::= <<
public static const <name>:BitSet = new BitSet([<words64:{ it |<it> };separator=", ">]);<n>
>>

codeFileExtension() ::= ".as"

true_value() ::= "true"
false_value() ::= "false"

Found in path(s):
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-

```

jar/org/antlr/codegen/templates/ActionScript/ActionScript.stg
No license file was found, but licenses were detected in source scan.

```
/*  
[The "BSD license"]  
Copyright (c) 2005-2006 Terence Parr  
All rights reserved.
```

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```
*/
```

```
/** The API version of the runtime that recognizers generated by this runtime  
* need.  
*/
```

```
apiVersion() ::= "1"
```

```
// System.Boolean.ToString() returns "True" and "False", but the proper C# literals are "true" and "false"  
// The Java version of Boolean returns "true" and "false", so they map to themselves here.
```

```
booleanLiteral ::= [  
    "True": "true",  
    "False": "false",  
    "true": "true",  
    "false": "false",  
    default: "false"  
]
```

```
/** The overall file structure of a recognizer; stores methods for rules  
* and cyclic DFAs plus support code.
```

```

*/
outputFile(LEXER,PARSER,TREE_PARSER, actionScope, actions,
           docComment, recognizer,
           name, tokens, tokenNames, rules, cyclicDFAs,
           bitsets, buildTemplate, buildAST, rewriteMode, profile,
           backtracking, synpreds, memoize, numRules,
           fileName, ANTLRVersion, generatedTimestamp, trace,
           scopes, superClass, literals) ::=
<<
# $ANTLR <ANTLRVersion> <fileName> <generatedTimestamp>

<@imports>
import sys
from antlr3 import *
<if(TREE_PARSER)>
from antlr3.tree import *<\n>
<endif>
from antlr3.compat import set, frozenset
<@end>

<actions.(actionScope).header>

<! <docComment> !>

# for convenience in actions
HIDDEN = BaseRecognizer.HIDDEN

# token types
<tokens:{it | <it.name>=<it.type>}; separator="\n">

<recognizer>

<if(actions.(actionScope).main)>
<actions.(actionScope).main>
<else>
def main(argv, stdin=sys.stdin, stdout=sys.stdout, stderr=sys.stderr):
<if(LEXER)>
    from antlr3.main import LexerMain
    main = LexerMain(<recognizer.name>)<\n>
<endif>
<if(PARSER)>
    from antlr3.main import ParserMain
    main = ParserMain("<recognizer.grammar.name>Lexer", <recognizer.name>)<\n>
<endif>
<if(TREE_PARSER)>
    from antlr3.main import WalkerMain
    main = WalkerMain(<recognizer.name>)<\n>
<endif>

```

```

    main.stdin = stdin
    main.stdout = stdout
    main.stderr = stderr
    main.execute(argv)<\n>
<endif>

<actions.(actionScope).footer>

if __name__ == '__main__':
    main(sys.argv)

>>

lexer(grammar, name, tokens, scopes, rules, numRules, filterMode,
    labelType="CommonToken", superClass="Lexer") ::= <<
<grammar.directDelegates:
{g|from <g.recognizerName> import <g.recognizerName>; separator="\n">

class <grammar.recognizerName>(<@superClassName><superClass><@end>):
    <scopes:{it|<if(it.isDynamicGlobalScope)><globalAttributeScope(scope=it)><endif>>>

    grammarFileName = "<fileName>"
    api_version = <apiVersion()>

    def __init__(self<grammar.delegators:{g|, <g:delegateName()>>, input=None, state=None):
        if state is None:
            state = RecognizerSharedState()
            super(<grammar.recognizerName>, self).__init__(input, state)

<if(memoize)>
<if(grammar.grammarIsRoot)>
    self._state.ruleMemo = {}
<endif>
<endif>

    <grammar.directDelegates:
    {g|self.<g:delegateName()> = <g.recognizerName>(<trunc(g.delegators):{p|<p:delegateName()>, }>self, input,
state)); separator="\n">
    <grammar.directDelegates:
    {g|<g.delegates:{h|self.<h:delegateName()> = self.<g:delegateName()>.<h:delegateName()>};
separator="\n">}; separator="\n">
    <grammar.delegators:
    {g|self.<g:delegateName()> = <g:delegateName()>; separator="\n">
    <last(grammar.delegators):
    {g|self.gParent = <g:delegateName()>; separator="\n">
    self.delegates = [<grammar.delegates: {g|self.<g:delegateName()>}; separator = ", ">]

    <cyclicDFAs:{dfa | <cyclicDFAInit(dfa)>}; separator="\n">

```

```

    <actions.lexer.init>

    <actions.lexer.members>

<if(filterMode)>
    <filteringNextToken()>
<endif>
    <rules; separator="\n\n">

    <synpreds:{p | <lexerSynpred(p)>}>

    <cyclicDFAs:cyclicDFA()> <! dump tables for all DFA !>

>>

/** A override of Lexer.nextToken() that backtracks over mTokens() looking
 * for matches. No error can be generated upon error; just rewind, consume
 * a token and then try again. backtracking needs to be set as well.
 * Make rule memoization happen only at levels above 1 as we start mTokens
 * at backtracking==1.
 */
filteringNextToken() ::= <<
def nextToken(self):
    while True:
        if self.input.LA(1) == EOF:
            return self.makeEOFToken()

        self._state.token = None
        self._state.channel = DEFAULT_CHANNEL
        self._state.tokenStartCharIndex = self.input.index()
        self._state.tokenStartCharPositionInLine = self.input.charPositionInLine
        self._state.tokenStartLine = self.input.line
        self._state._text = None
        try:
            m = self.input.mark()
            try:
                # means we won't throw slow exception
                self._state.backtracking = 1
                try:
                    self.mTokens()
                finally:
                    self._state.backtracking = 0
            except BacktrackingFailed:

```

```

        # mTokens backtracks with synpred at backtracking==2
        # and we set the synpredgate to allow actions at level 1.
        self.input.rewind(m)
        self.input.consume() # advance one char and try again

    else:
        self.emit()
        return self._state.token

except RecognitionException, re:
    # shouldn't happen in backtracking mode, but...
    self.reportError(re)
    self.recover(re)

def memoize(self, input, ruleIndex, ruleStartIndex, success):
    if self._state.backtracking > 1:
        # is Lexer always superclass?
        super(<grammar.recognizerName>, self).memoize(input, ruleIndex, ruleStartIndex, success)

def alreadyParsedRule(self, input, ruleIndex):
    if self._state.backtracking > 1:
        return super(<grammar.recognizerName>, self).alreadyParsedRule(input, ruleIndex)
    return False

>>

actionGate() ::= "self._state.backtracking == 0"

filteringActionGate() ::= "self._state.backtracking == 1"

/** How to generate a parser */

genericParser(grammar, name, scopes, tokens, tokenNames, rules, numRules,
              bitsets, inputStreamType, superClass, labelType, members,
              rewriteElementType, filterMode, init, ASTLabelType="Object") ::= <<
<if(grammar.grammarIsRoot)>
# token names
tokenNames = [
    "\<invalid>", "\<EOR>", "\<DOWN>", "\<UP>",
    <tokenNames; wrap, separator=", ">
]<\n>
<else>
from <grammar.composite.rootGrammar.recognizerName> import tokenNames<\n>
<endif>
<scopes:{it|<if(it.isDynamicGlobalScope)><globalAttributeScopeClass(scope=it)><endif>}>

```

```

<grammar.directDelegates:
{g|from <g.recognizerName> import <g.recognizerName>; separator="\n">

<rules:{it|<ruleAttributeScopeClass(scope=it.ruleDescriptor.ruleScope)>>

class <grammar.recognizerName>(<@superClassName><superClass><@end>):
    grammarFileName = "<fileName>"
    api_version = <apiVersion()>
    tokenNames = tokenNames

    def __init__(self<grammar.delegators:{g|, <g.delegateName()>>, input, state=None, *args, **kwargs):
        if state is None:
            state = RecognizerSharedState()

        <@args()>
        super(<grammar.recognizerName>, self).__init__(input, state, *args, **kwargs)

<if(memoize)>
<if(grammar.grammarIsRoot)>
    self._state.ruleMemo = {}
<endif>
<endif>

    <cyclicDFAs:{dfa | <cyclicDFAInit(dfa)>>; separator="\n">

    <scopes:{it | <if(it.isDynamicGlobalScope)><globalAttributeScopeStack(scope=it)><endif>>
<rules:{it | <ruleAttributeScopeStack(scope=it.ruleDescriptor.ruleScope)>>

    <init>

    <grammar.delegators:
    {g|self.<g.delegateName()> = <g.delegateName()>; separator="\n">
    <grammar.directDelegates:
    {g|self.<g.delegateName()> = <g.recognizerName>(<trunc(g.delegators):{p|<p.delegateName()>, }>self, input,
state)); separator="\n">
    <grammar.directDelegates:
    {g|<g.delegates:{h|self.<h.delegateName()> = self.<g.delegateName()>.<h.delegateName()>};
separator="\n">}; separator="\n">
    <last(grammar.delegators):
    {g|self.gParent = self.<g.delegateName()>; separator="\n">
    self.delegates = [<grammar.delegates: {g|self.<g.delegateName()>}; separator = ", ">]

    <@init><@end>

    <@members><@end>

```

```

<members>

<rules; separator="\n\n">

<! generate rule/method definitions for imported rules so they
  appear to be defined in this recognizer. !>
<grammar.delegatedRules:{ruleDescriptor| <delegateRule(ruleDescriptor)> }; separator="\n">

<synpreds:{p | <synpred(p)>>

<cyclicDFAs:cyclicDFA()> <! dump tables for all DFA !>

<bitsets:{it | FOLLOW_<it.name>_in_<it.inName><it.tokenIndex> = frozenset([<it.tokenTypes:{it |
<it>};separator=", ">])<\n>>

>>

delegateRule(ruleDescriptor) ::= <<
def <ruleDescriptor.name>(self, <ruleDescriptor.parameterScope:parameterScope()>):
<\> <if(ruleDescriptor.hasReturnValue)>return
<endif>self.<ruleDescriptor.grammar:delegateName()>.<ruleDescriptor.name><(if(ruleDescriptor.parameterScope)
><ruleDescriptor.parameterScope.attributes:{a|<a.name>}; separator=", "><endif>)

>>

parser(grammar, name, scopes, tokens, tokenNames, rules, numRules, bitsets,
  ASTLabelType="Object", superClass="Parser", labelType="Token",
  members={<actions.parser.members>},
  init={<actions.parser.init>}
) ::= <<
<genericParser(grammar, name, scopes, tokens, tokenNames, rules, numRules,
  bitsets, "TokenStream", superClass,
  labelType, members, "Token",
  false, init, ASTLabelType)>

>>

/** How to generate a tree parser; same as parser except the input
 * stream is a different type.
 */
treeParser(grammar, name, scopes, tokens, tokenNames, globalAction, rules,
  numRules, bitsets, filterMode, labelType={<ASTLabelType>}, ASTLabelType="Object",
superClass={<if(filterMode)><if(buildAST)>TreeRewriter<else>TreeFilter<endif><else>TreeParser<endif>},
members={<actions.treeparser.members>},
init={<actions.treeparser.init>}
) ::= <<
<genericParser(grammar, name, scopes, tokens, tokenNames, rules, numRules,
  bitsets, "TreeNodeStream", superClass,

```

```

        labelType, members, "Node",
        filterMode, init, ASTLabelType)>
>>

/** A simpler version of a rule template that is specific to the imaginary
 * rules created for syntactic predicates. As they never have return values
 * nor parameters etc..., just give simplest possible method. Don't do
 * any of the normal memoization stuff in here either; it's a waste.
 * As predicates cannot be inlined into the invoking rule, they need to
 * be in a rule by themselves.
 */
synpredRule(ruleName, ruleDescriptor, block, description, nakedBlock) ::=
<<
# $ANTLR start "<ruleName>"
def <ruleName>_fragment(self, <ruleDescriptor.parameterScope:parameterScope()>):
    <ruleLabelDefs()>
<if(trace)>
    self.traceIn("<ruleName>_fragment", <ruleDescriptor.index>)
    try:
        <block>

    finally:
        self.traceOut("<ruleName>_fragment", <ruleDescriptor.index>)

<else>
    <block>
<endif>
# $ANTLR end "<ruleName>"

>>

synpred(name) ::= <<
def <name>(self):
    self._state.backtracking += 1
    <@start()>
    start = self.input.mark()
    try:
        self.<name>_fragment()
    except BacktrackingFailed:
        success = False
    else:
        success = True
    self.input.rewind(start)
    <@stop()>
    self._state.backtracking -= 1
    return success

```

```
>>
```

```
lexerSynpred(name) ::= <<
```

```
<synpred(name)>
```

```
>>
```

```
ruleMemoization(name) ::= <<
```

```
<if(memoize)>
```

```
if self._state.backtracking > 0 and self.alreadyParsedRule(self.input, <ruleDescriptor.index>):
```

```
    # for cached failed rules, alreadyParsedRule will raise an exception
```

```
    success = True
```

```
    return <ruleReturnValue()>
```

```
<endif>
```

```
>>
```

```
/** This rule has failed, exit indicating failure during backtrack */
```

```
ruleBacktrackFailure() ::= <<
```

```
<if(backtracking)>
```

```
if self._state.backtracking > 0:
```

```
    raise BacktrackingFailed
```

```
<endif>
```

```
>>
```

```
/** How to generate code for a rule. This includes any return type
```

```
* data aggregates required for multiple return values.
```

```
*/
```

```
rule(ruleName,ruleDescriptor,block,emptyRule,description,exceptions,finally,memoize) ::= <<
```

```
<returnScope(scope=ruleDescriptor.returnScope)>
```

```
# $ANTLR start "<ruleName>"
```

```
# <fileName>:<description>
```

```
<ruleDescriptor.actions.decorate>
```

```
def <ruleName>(self, <ruleDescriptor.parameterScope:parameterScope()>):
```

```
<if(trace)>
```

```
    self.traceIn("<ruleName>", <ruleDescriptor.index>)<\n>
```

```
<endif>
```

```
    <ruleScopeSetUp()>
```

```
    <ruleDeclarations()>
```

```
    <ruleLabelDefs()>
```

```
    <ruleDescriptor.actions.init>
```

```
    <@preamble()>
```

```
    <@body><ruleBody()><@end>
```

```
    <@postamble()>
```

```
    return <ruleReturnValue()>
```

```

# $ANTLR end "<ruleName>"
>>

ruleBody() ::= <<
<if(memoize)>
<if(backtracking)>
success = False<\n>
<endif>
<endif>
try:
  try:
    <ruleMemoization(name=ruleName)>
    <block>
    <ruleCleanUp()>
    <(ruleDescriptor.actions.after):execAction()>

<if(memoize)>
<if(backtracking)>
  success = True<\n>
<endif>
<endif>
<if(exceptions)>
  <exceptions:{ e|<catch(decl=e.decl,action=e.action)><\n> }>
<else>
<if(!emptyRule)>
<if(actions.(actionScope).rulecatch)>
  <actions.(actionScope).rulecatch>
<else>
  except RecognitionException, re:
    self.reportError(re)
    self.recover(self.input, re)
    <@setErrorReturnValue()>

<endif>
<else>
  finally:
    pass

<endif>
<endif>
finally:
<if(trace)>
  self.traceOut("<ruleName>", <ruleDescriptor.index>)<\n>
<endif>
  <memoize()>
  <ruleScopeCleanUp()>
  <finally>
  pass

```

>>

```
catch(decl,action) ::= <<
except <e.decl>:
  <e.action>
```

>>

```
ruleDeclarations() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
retval = self.<ruleDescriptor.name>_return()
retval.start = self.input.LT(1)<\n>
<elseif(ruleDescriptor.returnScope)>
<ruleDescriptor.returnScope.attributes: { a |
<a.name> = <if(a.initValue)><a.initValue><else>None<endif>
}>
<endif>
<if(memoize)>
<ruleDescriptor.name>_startIndex = self.input.index()
<endif>
>>
```

```
ruleScopeSetUp() ::= <<
<ruleDescriptor.useScopes: {it | self.<it>_stack.append(<it>_scope())}; separator="\n">
<ruleDescriptor.ruleScope: {it | self.<it.name>_stack.append(<it.name>_scope())}; separator="\n">
>>
```

```
ruleScopeCleanUp() ::= <<
<ruleDescriptor.useScopes: {it | self.<it>_stack.pop()}; separator="\n">
<ruleDescriptor.ruleScope: {it | self.<it.name>_stack.pop()}; separator="\n">
>>
```

```
ruleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,ruleDescriptor.tokenListLabels,
ruleDescriptor.wildcardTreeLabels,ruleDescriptor.wildcardTreeListLabels]
: {it | <it.label.text> = None }; separator="\n"
>
<[ruleDescriptor.tokenListLabels,ruleDescriptor.ruleListLabels,
ruleDescriptor.wildcardTreeListLabels]
: {it | list_<it.label.text> = None }; separator="\n"
>
<ruleDescriptor.ruleLabels:ruleLabelDef(); separator="\n">
<ruleDescriptor.ruleListLabels: {it | <it.label.text> = None }; separator="\n">
>>
```

```
lexerRuleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,
ruleDescriptor.tokenListLabels,
```

```

ruleDescriptor.ruleLabels]
  :{it | <it.label.text> = None}; separator="\n"
>
<ruleDescriptor.charLabels:{it | <it.label.text> = None}; separator="\n">
<[ruleDescriptor.tokenListLabels,
ruleDescriptor.ruleListLabels]
  :{it | list_<it.label.text> = None}; separator="\n"
>
>>

ruleReturnValue() ::= <%
<if(!ruleDescriptor.isSynPred)>
<if(ruleDescriptor.hasReturnValue)>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnName>
<else>
retval
<endif>
<endif>
<endif>
%>

ruleCleanup() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(!TREE_PARSER)>
retval.stop = self.input.LT(-1)<\n>
<endif>
<endif>
>>

memoize() ::= <<
<if(memoize)>
<if(backtracking)>
if self._state.backtracking > 0:
    self.memoize(self.input, <ruleDescriptor.index>, <ruleDescriptor.name>_startIndex, success)

<endif>
<endif>
>>

/** How to generate a rule in the lexer; naked blocks are used for
 * fragment rules.
 */
lexerRule(ruleName,nakedBlock,ruleDescriptor,block,memoize) ::= <<
# $ANTLR start "<ruleName>"
def m<ruleName>(self, <ruleDescriptor.parameterScope:parameterScope(>>):
<if(trace)>
    self.traceIn("<ruleName>", <ruleDescriptor.index>)<\n>

```

```

<endif>
  <ruleScopeSetUp()>
  <ruleDeclarations()>
<if(memoize)>
<if(backtracking)>
  success = False<\n>
<endif>
<endif>
  try:
<if(nakedBlock)>
  <ruleMemoization(name=ruleName)>
  <lexerRuleLabelDefs()>
  <ruleDescriptor.actions.init>
  <block><\n>
<else>
  _type = <ruleName>
  _channel = DEFAULT_CHANNEL

  <ruleMemoization(name=ruleName)>
  <lexerRuleLabelDefs()>
  <ruleDescriptor.actions.init>
  <block>
  <ruleCleanUp()>
  self._state.type = _type
  self._state.channel = _channel
  <(ruleDescriptor.actions.after):execAction()>
<endif>
<if(memoize)>
<if(backtracking)>
  success = True<\n>
<endif>
<endif>

  finally:
<if(trace)>
  self.traceOut("<ruleName>", <ruleDescriptor.index>)<\n>
<endif>
  <ruleScopeCleanUp()>
  <memoize()>
  pass

# $ANTLR end "<ruleName>"

>>

/** How to generate code for the implicitly-defined lexer grammar rule
 * that chooses between lexer rules.

```

```

*/
tokensRule(ruleName,nakedBlock,args,block,ruleDescriptor) ::= <<
def mTokens(self):
    <block><\n>

>>

// S U B R U L E S

/** A (...) subrule with multiple alternatives */
block(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
# <fileName>:<description>
alt<decisionNumber> = <maxAlt>
<decls>
<@body><blockBody()><@end>
>>

blockBody() ::= <<
<@predecision()>
<@decision><decision><@end>
<@postdecision()>
<@prebranch()>
<alts:{ a | <altSwitchCase(i, a)> }; separator="\nel">
<@postbranch()>
>>

/** A rule block with multiple alternatives */
ruleBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
# <fileName>:<description>
alt<decisionNumber> = <maxAlt>
<decls>
<@predecision()>
<@decision><decision><@end>
<@postdecision()>
<alts:{ a | <altSwitchCase(i, a)> }; separator="\nel">
>>

ruleBlockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
# <fileName>:<description>
<decls>
<@prealt()>
<alts>
<@postalt()>
>>

/** A special case of a (...) subrule with a single alternative */
blockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<

```

```

# <fileName>:<description>
<decls>
<@prealt()>
<alts>
<@postalt()>
>>

/** A (..)+ block with 1 or more alternatives */
positiveClosureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
# <fileName>:<description>
cnt<decisionNumber> = 0
<decls>
<@preloop()>
<@loopBody>
<positiveClosureBlockLoop()>
<@end>
<@postloop()>
>>

positiveClosureBlockLoop() ::= <<
while True: #loop<decisionNumber>
  alt<decisionNumber> = <maxAlt>
  <@predecision()>
  <@decisionBody><decision><@end>
  <@postdecision()>
  <alts:{a | <altSwitchCase(i, a)>}; separator="\nel">
  else:
    if cnt<decisionNumber> >= 1:
      break #loop<decisionNumber>

  <ruleBacktrackFailure()>
  eee = EarlyExitException(<decisionNumber>, self.input)
  <@earlyExitException()>
  raise eee

  cnt<decisionNumber> += 1
>>

positiveClosureBlockSingleAlt ::= positiveClosureBlock

/** A (..)* block with 1 or more alternatives */
closureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::=
<<
# <fileName>:<description>
<decls>
<@preloop()>
<@loopBody>

```

```

<closureBlockLoop()>
<@end>
<@postloop()>
>>

```

```

closureBlockLoop() ::= <<
while True: #loop<decisionNumber>
  alt<decisionNumber> = <maxAlt>
  <@predecision()>
  <@decisionBody><decision><@end>
  <@postdecision()>
  <alts:{a | <altSwitchCase(i, a)>} ; separator="\nel">
  else:
    break #loop<decisionNumber>
>>

```

```

closureBlockSingleAlt ::= closureBlock

```

```

/** Optional blocks (x)? are translated to (x|) by before code generation
 * so we can just use the normal block template
 */

```

```

optionalBlock ::= block

```

```

optionalBlockSingleAlt ::= block

```

```

/** A case in a switch that jumps to an alternative given the alternative
 * number. A DFA predicts the alternative and then a simple switch
 * does the jump to the code that actually matches that alternative.
 */

```

```

altSwitchCase(altNum,alt) ::= <<
if alt<decisionNumber> == <altNum>:
  <@prealt()>
  <alt>
>>

```

```

/** An alternative is just a list of elements; at outermost level */
alt(elements,altNum,description,autoAST,outerAlt,treeLevel,rew) ::= <<
# <fileName>:<description>
pass <! so empty alternatives are a valid block !>
<@declarations()>
<elements:element()>
<rew>
<@cleanup()>
>>

```

```

/** What to emit when there is no rewrite. For auto build
 * mode, does nothing.
 */

```

```

noRewrite(rewriteBlockLevel, treeLevel) ::= ""

// E L E M E N T S

/** Dump the elements one per line */
element(e) ::= <<
<@prematch(>
<e.el><\n>
>>

/** match a token optionally with a label in front */
tokenRef(token,label,elementIndex,terminalOptions={}) ::= <<
<if(label)><label> = <endif>self.match(self.input, <token>,
self.FOLLOW_<token>_in_<ruleName><elementIndex>)
>>

/** ids+=ID */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRef(token,label,elementIndex,terminalOptions)>
<listLabel(label, label)>
>>

listLabel(label, elem) ::= <<
if list_<label> is None:
    list_<label> = []
list_<label>.append(<elem>)<\n>
>>

/** match a character */
charRef(char,label) ::= <<
<if(label)>
<label> = self.input.LA(1)<\n>
<endif>
self.match(<char>)
>>

/** match a character range */
charRangeRef(a,b,label) ::= <<
<if(label)>
<label> = self.input.LA(1)<\n>
<endif>
self.matchRange(<a>, <b>)
>>

/** For now, sets are interval tests and must be tested inline */
matchSet(s,label,elementIndex,postmatchCode="",terminalOptions={}) ::= <<
<if(label)>
<label> = self.input.LT(1)<\n>

```

```

<endif>
if <s>:
    self.input.consume()
    <postmatchCode>
<if(!LEXER)>
    self._state.errorRecovery = False<\n>
<endif>

else:
    <ruleBacktrackFailure()>
    mse = MismatchedSetException(None, self.input)
    <@mismatchedSetException()>
<if(LEXER)>
    self.recover(mse)
    raise mse
<else>
    raise mse
    <! use following code to make it recover inline; remove throw mse;
    self.recoverFromMismatchedSet(
        self.input, mse, self.FOLLOW_set_in_<ruleName><elementIndex>
    )
    !>
<endif>
<\n>
>>

matchRuleBlockSet ::= matchSet

matchSetAndListLabel(s,label,elementIndex,postmatchCode) ::= <<
<matchSet(...)>
<listLabel(label, label)>
>>

/** Match a string literal */
lexerStringRef(string,label,elementIndex="0") ::= <<
<if(label)>
<label>Start = self.getCharIndex()
self.match(<string>)
<label>StartLine<elementIndex> = self.getLine()
<label>StartCharPos<elementIndex> = self.getCharPositionInLine()
<label> = <labelType>(input=self.input, type=INVALID_TOKEN_TYPE, channel=DEFAULT_CHANNEL,
start=<label>Start, stop=self.getCharIndex()-1)
<label>.setLine(<label>StartLine<elementIndex>)
<label>.setCharPositionInLine(<label>StartCharPos<elementIndex>)
<else>
self.match(<string>)
<endif>
>>

```

```

wildcard(token,label,elementIndex,terminalOptions={ }) ::= <<
<if(label)>
<label> = self.input.LT(1)<\n>
<endif>
self.matchAny(self.input)
>>

```

```

wildcardAndListLabel(token,label,elementIndex,terminalOptions={ }) ::= <<
<wildcard(...)>
<listLabel(label,label)>
>>

```

```

/** Match . wildcard in lexer */
wildcardChar(label, elementIndex) ::= <<
<if(label)>
<label> = self.input.LA(1)<\n>
<endif>
self.matchAny()
>>

```

```

wildcardCharListLabel(label, elementIndex) ::= <<
<wildcardChar(label, elementIndex)>
<listLabel(label, label)>
>>

```

```

/** Match a rule reference by invoking it possibly with arguments
 * and a return value or values. The 'rule' argument was the
 * target rule name, but now is type Rule, whose toString is
 * same: the rule name. Now though you can access full rule
 * descriptor stuff.
 */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
self._state.following.append(self.FOLLOW_<rule.name>_in_<ruleName><elementIndex>)
<if(label)><label> = <endif>self.<if(scope)><scope.delegateName()>.<endif><rule.name>(<args; separator="
">)<\n>
self._state.following.pop()
>>

```

```

/** ids+=rule */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(rule,label,elementIndex,args,scope)>
<listLabel(label, label)>
>>

```

```

/** A lexer rule reference
 * The 'rule' argument was the target rule name, but now
 * is type Rule, whose toString is same: the rule name.

```

```

* Now though you can access full rule descriptor stuff.
*/
lexerRuleRef(rule,label,args,elementIndex,scope) ::= <<
<if(label)>
<label>Start<elementIndex> = self.getCharIndex()
self.<if(scope)><scope.delegateName()>.<endif>m<rule.name>(<args; separator=", ">)
<label>StartLine<elementIndex> = self.getLine()
<label>StartCharPos<elementIndex> = self.getCharPositionInLine()
<label> = <labelType>(
  input=self.input,
  type=INVALID_TOKEN_TYPE,
  channel=DEFAULT_CHANNEL,
  start=<label>Start<elementIndex>,
  stop=self.getCharIndex()-1)
<label>.setLine(<label>StartLine<elementIndex>)
<label>.setCharPositionInLine(<label>StartCharPos<elementIndex>)
<else>
self.<if(scope)><scope.delegateName()>.<endif>m<rule.name>(<args; separator=", ">)
<endif>
>>

/** i+=INT in lexer */
lexerRuleRefAndListLabel(rule,label,args,elementIndex,scope) ::= <<
<lexerRuleRef(rule,label,args,elementIndex,scope)>
<listLabel(label, label)>
>>

/** EOF in the lexer */
lexerMatchEOF(label,elementIndex) ::= <<
<if(label)>
<label>Start<elementIndex> = self.getCharIndex()
<label>StartLine<elementIndex> = self.getLine()
<label>StartCharPos<elementIndex> = self.getCharPositionInLine()
self.match(EOF)
<label> = <labelType>(input=self.input, type=EOF, channel=DEFAULT_CHANNEL,
start=<label>Start<elementIndex>, stop=self.getCharIndex()-1)
<label>.setLine(<label>StartLine<elementIndex>)
<label>.setCharPositionInLine(<label>StartCharPos<elementIndex>)
<else>
self.match(EOF)
<endif>
>>

// used for left-recursive rules
recRuleDefArg()          ::= "<recRuleArg()>"
recRuleArg()            ::= "_p"
recRuleAltPredicate(ruleName, opPrec) ::= "<recRuleArg()> \<= <opPrec>"
recRuleSetResultAction() ::= "root_0 = $<ruleName>_primary.tree"

```

```
recRuleSetReturnAction(src, name) ::= "$<name> = $<src>.<name>"
```

```
/** match ^(root children) in tree parser */
```

```
tree(root, actionsAfterRoot, children, nullableChildList,  
    enclosingTreeLevel, treeLevel) ::= <<
```

```
<root:element()>
```

```
<actionsAfterRoot:element()>
```

```
<if(nullableChildList)>
```

```
if self.input.LA(1) == DOWN:
```

```
    self.match(self.input, DOWN, None)
```

```
    <children:element()>
```

```
    self.match(self.input, UP, None)
```

```
<else>
```

```
self.match(self.input, DOWN, None)
```

```
<children:element()>
```

```
self.match(self.input, UP, None)
```

```
<endif>
```

```
>>
```

```
/** Every predicate is used as a validating predicate (even when it is
```

```
* also hoisted into a prediction expression).
```

```
*/
```

```
validateSemanticPredicate(pred,description) ::= <<
```

```
if not (<evalPredicate(pred, description)>):
```

```
    <ruleBacktrackFailure()>
```

```
    raise FailedPredicateException(self.input, "<ruleName>", "<description>")
```

```
>>
```

```
// F i x e d D F A (if-then-else)
```

```
dfaState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
```

```
LA<decisionNumber>_<stateNumber> = self.input.LA(<k>)<\n>
```

```
<edges; separator="\n" >
```

```
else:
```

```
<if(eotPredictsAlt)>
```

```
    alt<decisionNumber> = <eotPredictsAlt>
```

```
<else>
```

```
    <ruleBacktrackFailure()>
```

```
    nvae = NoViableAltException("<description>", <decisionNumber>, <stateNumber>, self.input)<\n>
```

```
    <@noViableAltException()>
```

```
    raise nvae<\n>
```

```
<endif>
```

```
>>
```

```
/** Same as a normal DFA state except that we don't examine lookahead
```

```
* for the bypass alternative. It delays error detection but this
```

```

* is faster, smaller, and more what people expect. For (X)? people
* expect "if ( LA(1)==X ) match(X);" and that's it.
*/
dfaOptionalBlockState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
LA<decisionNumber>_<stateNumber> = self.input.LA(<k>)<\n>
<edges; separator="\n" >
>>

/** A DFA state that is actually the loopback decision of a closure
* loop. If end-of-token (EOT) predicts any of the targets then it
* should act like a default clause (i.e., no error can be generated).
* This is used only in the lexer so that for ('a')* on the end of a rule
* anything other than 'a' predicts exiting.
*/
dfaLoopbackState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
LA<decisionNumber>_<stateNumber> = self.input.LA(<k>)<\n>
<edges; separator="\n" ><\n>
<if(eotPredictsAlt)>
<if(!edges)>
alt<decisionNumber> = <eotPredictsAlt> <! if no edges, don't gen ELSE !>
<else>
else:
  alt<decisionNumber> = <eotPredictsAlt>
<\n>
<endif>
<endif>
>>

/** An accept state indicates a unique alternative has been predicted */
dfaAcceptState(alt) ::= "alt<decisionNumber> = <alt>"

/** A simple edge with an expression. If the expression is satisfied,
* enter to the target state. To handle gated productions, we may
* have to evaluate some predicates for this edge.
*/
dfaEdge(labelExpr, targetState, predicates) ::= <<
if (<labelExpr>) <if(predicates)>and (<predicates>)<endif>:
  <targetState>
>>

// F i x e d D F A (switch case)

/** A DFA state where a SWITCH may be generated. The code generator
* decides if this is possible: CodeGenerator.canGenerateSwitch().
*/
dfaStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
<!
  FIXME: this is one of the few occasion, where I miss a switch statement

```

in Python. ATM this is implemented as a list of if .. elif ..

This may be replaced by faster a dictionary lookup, when I find a solution for the cases when an edge is not a plain dfaAcceptState.

```
!>
LA<decisionNumber> = self.input.LA(<k>)
<edges; separator="\n" >
else:
<if(eotPredictsAlt)>
    alt<decisionNumber> = <eotPredictsAlt>
<else>
    <ruleBacktrackFailure()>
    nvae = NoViableAltException("<description>", <decisionNumber>, <stateNumber>, self.input)<\n>
    <@noViableAltException()>
    raise nvae<\n>
<endif>

>>

dfaOptionalBlockStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
LA<decisionNumber> = self.input.LA(<k>)
<edges; separator="\n" >
>>

dfaLoopbackStateSwitch(k, edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
LA<decisionNumber> = self.input.LA(<k>)
<edges; separator="\n" >
<if(eotPredictsAlt)>
else:
    alt<decisionNumber> = <eotPredictsAlt>
<endif>
>>

dfaEdgeSwitch(labels, targetState) ::= <<
if <labels:{it | LA<decisionNumber> == <it>} ; separator=" or ">:
    <targetState>
>>

// C y c l i c D F A

/** The code to initiate execution of a cyclic DFA; this is used
 * in the rule to predict an alt just like the fixed DFA case.
 * The <name> attribute is inherited via the parser, lexer, ...
 */
dfaDecision(decisionNumber,description) ::= <<
alt<decisionNumber> = self.dfa<decisionNumber>.predict(self.input)
>>

/* Dump DFA tables as run-length-encoded Strings of octal values.
```

- * Can't use hex as compiler translates them before compilation.
- * These strings are split into multiple, concatenated strings.
- * Java puts them back together at compile time thankfully.
- * Java cannot handle large static arrays, so we're stuck with this
- * encode/decode approach. See analysis and runtime DFA for
- * the encoding methods.

*/

cyclicDFA(dfa) ::= <<

lookup tables for DFA #<dfa.decisionNumber>

```
DFA<dfa.decisionNumber>_eot = DFA.unpack(
  u"<dfa.javaCompressedEOT; wrap=""\n  u\>"
)
```

```
DFA<dfa.decisionNumber>_eof = DFA.unpack(
  u"<dfa.javaCompressedEOF; wrap=""\n  u\>"
)
```

```
DFA<dfa.decisionNumber>_min = DFA.unpack(
  u"<dfa.javaCompressedMin; wrap=""\n  u\>"
)
```

```
DFA<dfa.decisionNumber>_max = DFA.unpack(
  u"<dfa.javaCompressedMax; wrap=""\n  u\>"
)
```

```
DFA<dfa.decisionNumber>_accept = DFA.unpack(
  u"<dfa.javaCompressedAccept; wrap=""\n  u\>"
)
```

```
DFA<dfa.decisionNumber>_special = DFA.unpack(
  u"<dfa.javaCompressedSpecial; wrap=""\n  u\>"
)
```

```
DFA<dfa.decisionNumber>_transition = [
  <dfa.javaCompressedTransition:{s|DFA.unpack(u"<s; wrap=""\nu\>"}); separator=",\n">
]
```

class definition for DFA #<dfa.decisionNumber>

```
class DFA<dfa.decisionNumber>(DFA):
```

```
  pass
```

```
  <@errorMethod()>
```

```
  <if(dfa.specialStateSTs)>
```

```
    def specialStateTransition(self_, s, input):
```

```

# convince pylint that my self_ magic is ok ;)
# pylint: disable-msg=E0213

# pretend we are a member of the recognizer
# thus semantic predicates can be evaluated
self = self_.recognizer

_s = s

<dfa.specialStateSTs:{state | if s == <i0>: <! compressed special state numbers 0..n-1 !>
<state>} ; separator="\nел">

<if(backtracking)>
    if self._state.backtracking > 0:
        raise BacktrackingFailed

<endif>
    nvae = NoViableAltException(self_.getDescription(), <dfa.decisionNumber>, _s, input)
    self_.error(nvae)
    raise nvae<\n>
<endif>

>>

cyclicDFAInit(dfa) ::= <<
self.dfa<dfa.decisionNumber> = self.DFA<dfa.decisionNumber>(
    self, <dfa.decisionNumber>,
    eot = self.DFA<dfa.decisionNumber>_eot,
    eof = self.DFA<dfa.decisionNumber>_eof,
    min = self.DFA<dfa.decisionNumber>_min,
    max = self.DFA<dfa.decisionNumber>_max,
    accept = self.DFA<dfa.decisionNumber>_accept,
    special = self.DFA<dfa.decisionNumber>_special,
    transition = self.DFA<dfa.decisionNumber>_transition
)<\n>
>>

/** A state in a cyclic DFA; it's a special state and part of a big switch on
 * state.
 */
cyclicDFAState(decisionNumber,stateNumber,edges,needErrorClause,semPredState) ::= <<
LA<decisionNumber>_<stateNumber> = input.LA(1)<\n>
<if(semPredState)> <! get next lookahead symbol to test edges, then rewind !>
index<decisionNumber>_<stateNumber> = input.index()
input.rewind()<\n>
<endif>
s = -1
<edges; separator="\nел">

```

```

<if(semPredState)> <! return input cursor to state before we rewound !>
input.seek(index<decisionNumber>_<stateNumber>)<\n>
<endif>
if s >= 0:
    return s
>>

/** Just like a fixed DFA edge, test the lookahead and indicate what
 * state to jump to next if successful.
 */
cyclicDFAEdge(labelExpr, targetStateNumber, edgeNumber, predicates) ::= <<
if (<labelExpr>)<if(predicates)> and (<predicates>)<endif>:
    s = <targetStateNumber><\n>
>>

/** An edge pointing at end-of-token; essentially matches any char;
 * always jump to the target.
 */
eotDFAEdge(targetStateNumber,edgeNumber, predicates) ::= <<
se:
    s = <targetStateNumber><\n>
>>

// D F A E X P R E S S I O N S

andPredicates(left,right) ::= "((<left>) and (<right>))"

orPredicates(operands) ::= "(<operands; separator=\" or \">)"

notPredicate(pred) ::= "not (<evalPredicate(pred, { })>)"

evalPredicate(pred,description) ::= "(<pred>)"

evalSynPredicate(pred,description) ::= "self.<pred>()"

lookaheadTest(atom,k,atomAsInt) ::= "LA<decisionNumber>_<stateNumber> == <atom>"

/** Sometimes a lookahead test cannot assume that LA(k) is in a temp variable
 * somewhere. Must ask for the lookahead directly.
 */
isolatedLookaheadTest(atom,k,atomAsInt) ::= "self.input.LA(<k>) == <atom>"

lookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= <%
(<lower> \<= LA<decisionNumber>_<stateNumber> \<= <upper>)
%>

isolatedLookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= "(<lower> \<=

```

```
self.input.LA(<k>) \<= <upper>)"
```

```
setTest(ranges) ::= "<ranges; separator=\\\" or \\\">"
```

```
// A T T R I B U T E S
```

```
globalAttributeScopeClass(scope) ::= <<
```

```
<if(scope)>
```

```
<if(scope.attributes)>
```

```
class <scope.name>_scope(object):
```

```
    def __init__(self):
```

```
        <scope.attributes:{it | self.<it.decl> = None}; separator=\\\"n\">
```

```
<endif>
```

```
<endif>
```

```
>>
```

```
globalAttributeScopeStack(scope) ::= <<
```

```
<if(scope)>
```

```
<if(scope.attributes)>
```

```
self.<scope.name>_stack = []<\n>
```

```
<endif>
```

```
<endif>
```

```
>>
```

```
ruleAttributeScopeClass(scope) ::= <<
```

```
<if(scope)>
```

```
<if(scope.attributes)>
```

```
class <scope.name>_scope(object):
```

```
    def __init__(self):
```

```
        <scope.attributes:{it | self.<it.decl> = None}; separator=\\\"n\">
```

```
<endif>
```

```
<endif>
```

```
>>
```

```
ruleAttributeScopeStack(scope) ::= <<
```

```
<if(scope)>
```

```
<if(scope.attributes)>
```

```
self.<scope.name>_stack = []<\n>
```

```
<endif>
```

```
<endif>
```

```
>>
```

```
delegateName(d) ::= <<
```

```
<if(d.label)><d.label><else>g<d.name><endif>
```

```
>>
```

```

/** Define a rule label including default value */
ruleLabelDef(label) ::= <<
<label.label.text> = None
>>

returnStructName(r) ::= "<r.name>_return"

/** Define a return struct for a rule if the code needs to access its
 * start/stop tokens, tree stuff, attributes, ... Leave a hole for
 * subgroups to stick in members.
 */
returnScope(scope) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
class <ruleDescriptor:returnStructName()>( <if(TREE_PARSER)>Tree<else>Parser<endif>RuleReturnScope):
  def __init__(self):
    super(<grammar.recognizerName>.<ruleDescriptor:returnStructName()>, self).__init__()

    <if(scope)><scope.attributes: {it | self.<it.decl> = None}; separator="\n"><endif>
    <@ruleReturnInit()>

    <@ruleReturnMembers()>

<endif>
>>

parameterScope(scope) ::= <<
<if(scope)><scope.attributes: {it | <it.decl>}; separator=", "><endif>
>>

parameterAttributeRef(attr) ::= "<attr.name>"
parameterSetAttributeRef(attr,expr) ::= "<attr.name> = <expr>"

scopeAttributeRef(scope,attr,index,negIndex) ::= <%
<if(negIndex)>
self.<scope>_stack[-<negIndex>].<attr.name>
<else>
<if(index)>
self.<scope>_stack[<index>].<attr.name>
<else>
self.<scope>_stack[-1].<attr.name>
<endif>
<endif>
%>

/* not applying patch because of bug in action parser!

<if(negIndex)>

```

```

((len(self.<scope>_stack) - <negIndex> - 1) >= 0 and [self.<scope>_stack[-<negIndex>].<attr.name>] or [None])[0]
<else>
<if(index)>
((<index> \< len(self.<scope>_stack)) and [self.<scope>_stack[<index>].<attr.name>] or [None])[0]
<else>
((len(self.<scope>_stack) > 0) and [self.<scope>_stack[-1].<attr.name>] or [None])[0]
<endif>
<endif>

*/

scopeSetAttributeRef(scope,attr,expr,index,negIndex) ::= <%
<if(negIndex)>
<!--FIXME: this seems not to be used by ActionTranslator...!-->
self.<scope>_stack[-<negIndex>].<attr.name> = <expr>
<else>
<if(index)>
<!--FIXME: this seems not to be used by ActionTranslator...!-->
self.<scope>_stack[<index>].<attr.name> = <expr>
<else>
self.<scope>_stack[-1].<attr.name> = <expr>
<endif>
<endif>
%>

/** $x is either global scope or x is rule with dynamic scope; refers
 * to stack itself not top of stack. This is useful for predicates
 * like {$function.size()>0 && $function::name.equals("foo")}?
 */
isolatedDynamicScopeRef(scope) ::= "self.<scope>_stack"

/** reference an attribute of rule; might only have single return value */
ruleLabelRef(referencedRule,scope,attr) ::= <%
<if(referencedRule.hasMultipleReturnValues)>
((<scope> is not None) and [<scope>.<attr.name>] or [None])[0]
<else>
<scope>
<endif>
%>

returnAttributeRef(ruleDescriptor,attr) ::= <%
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name>
<else>
<attr.name>
<endif>
%>

```

```

returnSetAttributeRef(ruleDescriptor,attr,expr) ::= <%
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name> = <expr>
<else>
<attr.name> = <expr>
<endif>
%>

/** How to translate $tokenLabel */
tokenLabelRef(label) ::= "<label>"

/** ids+=ID {$ids} or e+=expr {$e} */
listLabelRef(label) ::= "list_<label>"

// not sure the next are the right approach; and they are evaluated early;
// they cannot see TREE_PARSER or PARSER attributes for example. :(

tokenLabelPropertyRef_text(scope,attr) ::= "<scope>.text"
tokenLabelPropertyRef_type(scope,attr) ::= "<scope>.type"
tokenLabelPropertyRef_line(scope,attr) ::= "<scope>.line"
tokenLabelPropertyRef_pos(scope,attr) ::= "<scope>.charPositionInLine"
tokenLabelPropertyRef_channel(scope,attr) ::= "<scope>.channel"
tokenLabelPropertyRef_index(scope,attr) ::= "<scope>.index"
tokenLabelPropertyRef_tree(scope,attr) ::= "<scope>_tree"

ruleLabelPropertyRef_start(scope,attr) ::= "<scope>.start"
ruleLabelPropertyRef_stop(scope,attr) ::= "<scope>.stop"
ruleLabelPropertyRef_tree(scope,attr) ::= "<scope>.tree"
ruleLabelPropertyRef_text(scope,attr) ::= <%
<if(TREE_PARSER)>
((<scope> is not None) and [self.input.getTokenStream().toString(
    self.input.getTreeAdaptor().getTokenStartIndex(<scope>.start),
    self.input.getTreeAdaptor().getTokenStopIndex(<scope>.start)
)] or [None])[0]
<else>
((<scope> is not None) and [self.input.toString(<scope>.start,<scope>.stop)] or [None])[0]
<endif>
%>
ruleLabelPropertyRef_st(scope,attr) ::= "((<scope> is not None) and [<scope>.st] or [None])[0]"

/** Isolated $RULE ref ok in lexer as it's a Token */
lexerRuleLabel(label) ::= "<label>"

lexerRuleLabelPropertyRef_type(scope,attr) ::= "((<scope> is not None) and [<scope>.type] or [0])[0]"
lexerRuleLabelPropertyRef_line(scope,attr) ::= "((<scope> is not None) and [<scope>.line] or [0])[0]"
lexerRuleLabelPropertyRef_pos(scope,attr) ::= "((<scope> is not None) and [<scope>.charPositionInLine] or [0])[0]"

```

```

lexerRuleLabelPropertyRef_channel(scope,attr) ::= "((<scope> is not None) and [<scope>.channel] or [0])[0]"
lexerRuleLabelPropertyRef_index(scope,attr) ::= "((<scope> is not None) and [<scope>.index] or [0])[0]"
lexerRuleLabelPropertyRef_text(scope,attr) ::= "((<scope> is not None) and [<scope>.text] or [None])[0]"
lexerRuleLabelPropertyRef_int(scope,attr) ::= "((<scope> is not None) and [int(<scope>.text)] or [0])[0]"

```

```

// Somebody may ref $template or $tree or $stop within a rule:

```

```

rulePropertyRef_start(scope,attr) ::= "retval.start"
rulePropertyRef_stop(scope,attr) ::= "retval.stop" //mmm... or input.LT(-1)??
rulePropertyRef_tree(scope,attr) ::= "retval.tree"
rulePropertyRef_text(scope,attr) ::= "self.input.toString(retval.start, self.input.LT(-1))"
rulePropertyRef_st(scope,attr) ::= "retval.st"

```

```

lexerRulePropertyRef_text(scope,attr) ::= "self.text"
lexerRulePropertyRef_type(scope,attr) ::= "_type"
lexerRulePropertyRef_line(scope,attr) ::= "self._state.tokenStartLine"
lexerRulePropertyRef_pos(scope,attr) ::= "self._state.tokenStartCharPositionInLine"
lexerRulePropertyRef_index(scope,attr) ::= "-1" // undefined token index in lexer
lexerRulePropertyRef_channel(scope,attr) ::= "_channel"
lexerRulePropertyRef_start(scope,attr) ::= "self._state.tokenStartCharIndex"
lexerRulePropertyRef_stop(scope,attr) ::= "(self.getCharIndex()-1)"
lexerRulePropertyRef_int(scope,attr) ::= "int(<scope>.text)"

```

```

// setting $st and $tree is allowed in local rule. everything else
// is flagged as error

```

```

ruleSetPropertyRef_tree(scope,attr,expr) ::= "retval.tree =<expr>"
ruleSetPropertyRef_st(scope,attr,expr) ::= "retval.st =<expr>"

```

```

/** How to execute an action (only when not backtracking) */

```

```

execAction(action) ::= <<
<if(backtracking)>
<if(actions.(actionScope).synpredgate)>
if <actions.(actionScope).synpredgate>:
    pass
    <action>

<else>
if <actions.(actionScope).synpredgate>:
    pass
    <action>

<endif>
<else>
#action start
<action>
#action end
<endif>
>>

```

```

/** How to always execute an action even when backtracking */
execForcedAction(action) ::= "<action>"

// M I S C (properties, etc...)

codeFileExtension() ::= ".py"

true_value() ::= "True"
false_value() ::= "False"

Found in path(s):
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-
jar/org/antlr/codegen/templates/Python/Python.stg
No license file was found, but licenses were detected in source scan.

/*
[The "BSD license"]
Copyright (c) 2005-2012 Terence Parr
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

/** Template overrides to add debugging to AST stuff.  Dynamic inheritance
* hierarchy is set up as ASTDbg : AST : Dbg : Python by code generator.
*/

```

```

group ASTDbg;

astAccessor() ::= <<
def setTreeAdaptor(self, adaptor):
<if(grammar.grammarIsRoot)>
    self._adaptor = DebugTreeAdaptor(self.dbg, adaptor)
<else>
    self._adaptor = adaptor # delegator sends dbg adaptor
<endif>
    <grammar.directDelegates:{g|<g.delegateName()>.setTreeAdaptor(self._adaptor)}>

def getTreeAdaptor(self):
    return self._adaptor

adaptor = property(getTreeAdaptor, setTreeAdaptor)<\n>
>>

createListenerAndHandshake() ::= <<
proxy = DebugEventSocketProxy(self,
adaptor=<if(TREE_PARSER)>self.input.getTreeAdaptor()<else>self._adaptor<endif>,
                debug=debug_socket, port=port)
self.setDebugListener(proxy)
self.adaptor.setDebugListener(proxy)
self.input.setDebugListener(proxy)
#self.set<inputStreamType>(Debug<inputStreamType>(self.input, proxy))
proxy.handshake()
>>

@rewriteElement.pregen() ::= "self._dbg.location(<e.line>, <e.pos>)"

```

Found in path(s):

```
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-
jar/org/antlr/codegen/templates/Python3/ASTDbg.stg
```

No license file was found, but licenses were detected in source scan.

```
/* [The "BSD license"]
```

```
Copyright (c) 2008 Erik van Bilsen
Copyright (c) 2007-2008 Johannes Luber
Copyright (c) 2005-2007 Kunle Odutola
Copyright (c) 2005-2006 Terence Parr
All rights reserved.
```

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright

notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

group Delphi;

```
csharpTypeInitMap ::= [  
  "int":"0",  
  "uint":"0",  
  "long":"0",  
  "ulong":"0",  
  "float":"0.0",  
  "double":"0.0",  
  "bool":"False",  
  "byte":"0",  
  "sbyte":"0",  
  "short":"0",  
  "ushort":"0",  
  "char":"#0",  
  "string":"",  
  "String":"",  
  default:"nil" // anything other than an atomic type  
]  
  
/** The overall file structure of a recognizer; stores methods for rules  
 * and cyclic DFAs plus support code.  
 * LEXER (Boolean): should we generate lexer code?  
 * PARSER (Boolean): should we generate parser code?  
 * TREE_PARSER (Boolean): should we generate tree parser code?  
 * actionScope (String): 'lexer', 'parser', 'tree_parser' or custom scope  
 * actions (HashMap):  
 * docComment (String): document comment  
 * recognizer (Object): recognizer class generator  
 * name (String): name of grammar  
 * tokens (HashMap<name: String, type: Integer>):  
 * tokenNames:
```

```

* rules:
* cyclicDFAs:
* bitsets:
* buildTemplate (Boolean): should we generate a string template?
* buildAST (Boolean): should we generate an AST?
* rewriteMode (Boolean): are we rewriteing nodes?
* profile (Boolean):
* backtracking (Boolean): backtracking mode?
* synpreds (): syntactic predicates
* memoize (Boolean): should we memoize?
* numRules (Integer): number of rules
* fileName (String): fully qualified name of original .g file
* ANTLRVersion (String): ANTLR version in Major.Minor.Build format
* generatedTimestamp (String): date/time when the file is generated
* trace (Boolean): should we trace input/output?
* scopes:
* superClass (String): name of base class, or empty string
* literals:
*/
outputFile(LEXER,PARSER,TREE_PARSER, actionScope, actions,
           docComment, recognizer,
           name, tokens, tokenNames, rules, cyclicDFAs,
           bitsets, buildTemplate, buildAST, rewriteMode, profile,
           backtracking, synpreds, memoize, numRules,
           fileName, ANTLRVersion, generatedTimestamp, trace,
           scopes, superClass, literals) ::=
<<
unit <name>;

{$HINTS OFF}

// $ANTLR <ANTLRVersion> <fileName> <generatedTimestamp>

<actions.(actionScope).header>

interface

<@imports>
uses<\n>
<@end>
<actions.(actionScope).usesInterface>
<if(TREE_PARSER)>
  Antlr.Runtime.Tree,<\n>
<endif>
  Antlr.Runtime,
  Antlr.Runtime.Collections,
  Antlr.Runtime.Tools;

```

```

<docComment>
<recognizer>
>>

/** Generates source code for the lexer class
 * grammar (Grammar object)
 */
lexer(grammar, name, tokens, scopes, rules, numRules, labelType="Token",
    filterMode, superClass="Lexer") ::= <<
type
I<grammar.recognizerName> = interface(I<@superClassName><superClass><@end>)
end;

T<grammar.recognizerName> = class(T<@superClassName><superClass><@end>,
I<grammar.recognizerName>)
strict private
    FCnt: array [0..<grammar.numberOfDecisions>] of Byte;
    FLA: array [0..<grammar.numberOfDecisions>, 0..255] of Integer;
    FException: ERecognitionException;
    procedure InitializeCyclicDFAs;
<cyclicDFAs:cyclicDFADeclaration()>
public
    const
        <tokens:{<it.name> = <it.type>;} separator="\n">
        <scopes:{<if(it.isDynamicGlobalScope)><globalAttributeScope(scope=it)><endif>}>
strict private
    <actions.(actionScope).memberDeclarations>
public
    // delegates
    <grammar.delegates: {g|<g:delegateName(): I<superClass>; {<g.recognizerName>}}; separator="\n">
public
    // delegators
    <grammar.delegators: {g|<g:delegateName(): Pointer; {<g.recognizerName>}}; separator="\n">
    <last(grammar.delegators):{g|gParent: Pointer; {<g.recognizerName>}}>
protected
    { IBaseRecognizer }
    function GetGrammarFileName: String; override;
<if(filterMode)>
    function AlreadyParsedRule(const Input: IIntStream;
        const RuleIndex: Integer): Boolean; override;
    procedure Memoize(const Input: IIntStream; const RuleIndex,
        RuleStartIndex: Integer); override;
protected
    { ILexer }
    function NextToken: IToken; override;<\n>
<endif>
protected
    { ILexer }

```

```

    procedure DoTokens; override;
public
    constructor Create; overload;
    constructor Create(const AInput: ICharStream<grammar.delegators:{g}; const A<g:delegateName():
IBaseRecognizer{<g.recognizerName>}>); overload;
    constructor Create(const AInput: ICharStream; const AState: IRecognizerSharedState<grammar.delegators:{g};
const A<g:delegateName(): IBaseRecognizer{<g.recognizerName>}>); overload;

    <rules: {r | <if(!r.ruleDescriptor.isSynPred)><lexerRuleDeclaration(r)><endif>>
    <synpreds: {p | <lexerSynpredDeclaration(p)>}; separator="\n">
end;

implementation

uses
    <grammar.delegates: {g|<g.recognizerName>,}; separator="\n">
    <grammar.delegators: {g|<g.recognizerName>,}; separator="\n">
    <actions.(actionScope).usesImplementation>
    SysUtils,
    StrUtils,
    Math;

{ T<grammar.recognizerName> }

constructor T<grammar.recognizerName>.Create;
begin
    InitializeCyclicDFAs;
end;

constructor T<grammar.recognizerName>.Create(const AInput: ICharStream<grammar.delegators:{g}; const
A<g:delegateName(): IBaseRecognizer{<g.recognizerName>}>);
begin
    Create(AInput, nil<grammar.delegators:{g}, A<g:delegateName()>>);
end;

constructor T<grammar.recognizerName>.Create(const AInput: ICharStream; const AState:
IRecognizerSharedState<grammar.delegators:{g}; const A<g:delegateName():
IBaseRecognizer{<g.recognizerName>}>);
begin
    inherited Create(AInput, AState);
    InitializeCyclicDFAs; { TODO: Necessary in Delphi??? Not removed yet. }
    <if(memoize)>
    <if(grammar.grammarIsRoot)>
    State.RuleMemoCount := <numRules>+1;<\n> <! index from 1..n !>
    <endif>
    <endif>
    <grammar.directDelegates:
    {g|<g:delegateName()> := T<g.recognizerName>.Create(AInput, State<trunc(g.delegators):{p|,

```

```

<p:delegateName(>>, Self);}; separator="\n">
<grammar.delegators:
  {g<g:delegateName(>> := Pointer(A<g:delegateName(>>);}; separator="\n">
<last(grammar.delegators):{g|gParent := Pointer(A<g:delegateName(>>);}>
<actions.(actionScope).memberInitializations>
end;
<actions.(actionScope).memberImplementations>
function T<grammar.recognizerName>.GetGrammarFileName: String;
begin
  Result := '<fileName>';
end;

<if(filterMode)>
<filteringNextToken()>
<endif>

<rules; separator="\n\n">
<synpreds:{p | <lexerSynpred(p)>}>

procedure T<grammar.recognizerName>.InitializeCyclicDFAs;
begin
  <cyclicDFAs:{ dfa | FDFA<dfa.decisionNumber> :=
TDFA<dfa.decisionNumber>.Create(Self<@debugAddition(>>);}; separator="\n">
  <cyclicDFAs:{ dfa | <if(dfa.specialStateSTs)>FDFA<dfa.decisionNumber>.SpecialStateTransitionHandler :=
DFA<dfa.decisionNumber>_SpecialStateTransition;<endif>}; separator="\n">
end;

<cyclicDFAs:cyclicDFA(>> <! dump tables for all DFA !>
end.>>

lexerRuleDeclaration(rule) ::= <<
procedure m<rule.ruleName>(<rule.ruleDescriptor.parameterScope:parameterScope(scope=rule)>);<\n>
>>

/** A override of Lexer.nextToken() that backtracks over mTokens() looking
 * for matches. No error can be generated upon error; just rewind, consume
 * a token and then try again. backtracking needs to be set as well.
 *
 * Make rule memoization happen only at levels above 1 as we start mTokens
 * at backtracking==1.
 */
filteringNextToken() ::= <<
function T<grammar.recognizerName>.NextToken: IToken;
var
  M: Integer;
begin
  while (True) do
  begin

```

```

if (Input.LA(1) = Integer(cscEOF)) then
  Exit(TToken.EOF_TOKEN);

State.Token := nil;
State.Channel := TToken.DEFAULT_CHANNEL;
State.TokenStartCharIndex := Input.Index;
State.TokenStartCharPositionInLine := Input.CharPositionInLine;
State.TokenStartLine := Input.Line;
State.Text := "";
try
  M := Input.Mark();
  State.Backtracking := 1; <! means we won't throw slow exception !>
  State.Failed := False;
  mTokens();
  State.Backtracking := 0;
<!
  mTokens backtracks with synpred at backtracking==2
  and we set the synpredgate to allow actions at level 1.
!>
  if (State.Failed) then
    begin
      Input.Rewind(M);
      Input.Consume; <! // advance one char and try again !>
    end
  else
    begin
      Emit;
      Exit(State.Token);
    end;
  except
    on RE: ERecognitionException do
      begin
        // shouldn't happen in backtracking mode, but...
        ReportError(RE);
        Recover(RE);
      end;
    end;
  end;
end;

function T<grammar.recognizerName>.AlreadyParsedRule(const Input: IIntStream;
const RuleIndex: Integer): Boolean;
begin
  if (State.Backtracking > 1) then
    Result := inherited AlreadyParsedRule(Input, RuleIndex)
  else
    Result := False;
  end;
end;

```

```

procedure T<grammar.recognizerName>.Memoize(const Input: IIntStream; const RuleIndex,
RuleStartIndex: Integer);
begin
if (State.Backtracking > 1) then
inherited Memoize(Input, RuleIndex, RuleStartIndex);
end;

>>

filteringActionGate() ::= "(State.Backtracking = 1)"

/** How to generate a parser */
genericParser(grammar, name, scopes, tokens, tokenNames, rules, numRules,
bitsets, inputStreamType, superClass, filterMode,
ASTLabelType="ANTLRInterface", labelType, members, rewriteElementType) ::= <<
type
<rules: {r | <genericParserRuleReturnType(rule=r, ruleDescriptor=r.ruleDescriptor)>>>
I<grammar.recognizerName> = interface(I<@superClassName><superClass><@end>)
<rules: {r | <genericParserRuleInterface(rule=r, ruleDescriptor=r.ruleDescriptor)>>>
end;

T<grammar.recognizerName> = class(T<@superClassName><superClass><@end>,
I<grammar.recognizerName>)
<if(grammar.grammarIsRoot)>
public
const
TOKEN_NAMES: array [0..

```

```

public
  constructor Create(const AInput: <inputStreamType><grammar.delegators:{g|; const A<g:delegateName()>:
IBaseRecognizer{<g.recognizerName>}>); overload;
  constructor Create(const AInput: <inputStreamType>; const AState:
IRecognizerSharedState<grammar.delegators:{g|; const A<g:delegateName()>:
IBaseRecognizer{<g.recognizerName>}>); overload;
<@end>
protected
  { IBaseRecognizer }
  function GetTokenNames: TStringArray; override;
  function GetGrammarFileName: String; override;
strict private
  <actions.(actionScope).memberDeclarations>
  <rules: {r | <genericParserRuleDeclaration(rule=r, ruleDescriptor=r.ruleDescriptor)>}>

<! generate rule/method definitions for imported rules so they
appear to be defined in this recognizer. !>
  // Delegated rules
  <grammar.delegatedRules:{ruleDescriptor| <delegatedRuleDeclaration(ruleDescriptor)>}>

  <synpreds:{p | <synpredDeclaration(p)>}; separator="\n">
  <cyclicDFAs:cyclicDFADeclaration()>
strict private
  FException: ERecognitionException;
  FLA: array [0..<grammar.numberofDecisions>, 0..255] of Integer;
  FCnt: array [0..<grammar.numberofDecisions>] of Byte;
  procedure InitializeCyclicDFAs;
<if(bitsets)>
public
  class var
    <bitsets.bitsetDecl(name={ FOLLOW_<it.name>_in_<it.inName><it.tokenIndex>})>
public
  class procedure InitializeBitsets; static;<\n>
<endif>
end;

implementation

uses
  <grammar.delegates: {g|<g.recognizerName>,}; separator="\n">
  <grammar.delegators: {g|<g.recognizerName>,}; separator="\n">
  <actions.(actionScope).usesImplementation>
  SysUtils,
  StrUtils,
  Math;

  { T<grammar.recognizerName> }

```

```

constructor T<grammar.recognizerName>.Create(const AInput: <inputStreamType><grammar.delegators:{g}; const
A<g:delegateName(): IBaseRecognizer{<g.recognizerName>}>);
begin
  Create(AInput, TRecognizerSharedState.Create<grammar.delegators:{g}, A<g:delegateName()>>);
end;

constructor T<grammar.recognizerName>.Create(const AInput: <inputStreamType>;
const AState: IRecognizerSharedState<grammar.delegators:{g}; const A<g:delegateName():
IBaseRecognizer{<g.recognizerName>}>);
begin
  inherited Create(AInput, AState);
  <@membersConstructor>
  <@end>
  <parserCtorBody()>
  <grammar.directDelegates:{g|<g:delegateName()> := T<g.recognizerName>.Create(Input,
State<trunc(g.delegators):{p}, <p:delegateName()>>, Self);}; separator="\n">
  <grammar.indirectDelegates:{g | <g:delegateName()> := <g.delegator:delegateName()>.<g:delegateName()>};
separator="\n">
  <last(grammar.delegators):{g|gParent := Pointer(A<g:delegateName()>);}>
  <rules: {r | <ruleAttributeScopeInit(scope=r.ruleDescriptor.ruleScope)>>}>
  <scopes:{<if(it.isDynamicGlobalScope)><globalAttributeScope(scope=it)><endif>}>
  <actions.(actionScope).memberInitializations>
end;
<actions.(actionScope).memberImplementations>

<grammar.delegatedRules:{ruleDescriptor| <delegatedRuleImplementation(ruleDescriptor)>}; separator="\n">
procedure T<grammar.recognizerName>.InitializeCyclicDFAs;
begin
  <cyclicDFAs:{dfa | FDFA<dfa.decisionNumber> := TDFA<dfa.decisionNumber>.Create(Self);}; separator="\n">
  <cyclicDFAs:{dfa | <if(dfa.specialStateSTs)>FDFA<dfa.decisionNumber>.SpecialStateTransitionHandler :=
DFA<dfa.decisionNumber>_SpecialStateTransition;<endif>}; separator="\n">
end;

<if(bitsets)>
class procedure T<grammar.recognizerName>.InitializeBitsets;
begin
  <bitsets:bitset(name={FOLLOW_<it.name>_in_<it.inName><it.tokenIndex>}, words64=it.bits)>
end;
<endif>

<@membersImplementation>
<@end>

function T<grammar.recognizerName>.GetTokenNames: TStringArray;
var
  I: Integer;
begin
  SetLength(Result,Length(T<grammar.composite.rootGrammar.recognizerName>.TOKEN_NAMES));

```

```

for I := 0 to Length(T<grammar.composite.rootGrammar.recognizerName>.TOKEN_NAMES) - 1 do
  Result[I] := T<grammar.composite.rootGrammar.recognizerName>.TOKEN_NAMES[I];
end;

function T<grammar.recognizerName>.GetGrammarFileName: String;
begin
  Result := '<fileName>';
end;

<rules; separator="\n\n">
<synpreds: {p | <synpred(p)>}>

<cyclicDFAs:cyclicDFA()> <! dump tables for all DFA !>
<if(bitsets)>
initialization
  T<grammar.recognizerName>.InitializeBitsets;<\n>
<endif>
end.>>

delegatedRuleDeclaration(ruleDescriptor) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
function <ruleDescriptor.name>(<ruleDescriptor.parameterScope:parameterScope(scope=it)>):
I<returnType()>;<\n>
<else>
<if(ruleDescriptor.hasSingleReturnValue)>
function <ruleDescriptor.name>(<ruleDescriptor.parameterScope:parameterScope(scope=it)>):
<returnType()>;<\n>
<else>
procedure <ruleDescriptor.name>(<ruleDescriptor.parameterScope:parameterScope(scope=it)>);<\n>
<endif>
<endif>
>>

delegatedRuleImplementation(ruleDescriptor) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
function
  T<grammar.recognizerName>.<ruleDescriptor.name>(<ruleDescriptor.parameterScope:parameterScope(scope=it)>
): I<returnType()>;<\n>
<else>
<if(ruleDescriptor.hasSingleReturnValue)>
function
  T<grammar.recognizerName>.<ruleDescriptor.name>(<ruleDescriptor.parameterScope:parameterScope(scope=it)>
): <returnType()>;<\n>
<else>
procedure
  T<grammar.recognizerName>.<ruleDescriptor.name>(<ruleDescriptor.parameterScope:parameterScope(scope=it)>
);<\n>
<endif>

```

```

<endif>
begin
  <if(ruleDescriptor.hasReturnValue)>Result :=<endif>
  T<ruleDescriptor.grammar.recognizerName>(<ruleDescriptor.grammar.delegateName()>.Implementor).<ruleDescriptor.name>(<ruleDescriptor.parameterScope.attributes:{a|<a.name>}; separator=", ">);
end;

>>

parserCtorBody() ::= <<
InitializeCyclicDFAs;
<if(memoize)>
<if(grammar.grammarIsRoot)>
State.RuleMemoCount := <length(grammar.allImportedRules)>+1;<\n> <! index from 1..n !>
<endif>
<endif>
<grammar.delegators: {g|<g.delegateName()> := Pointer(A<g.delegateName()>); separator="\n">
>>

parser(grammar, name, scopes, tokens, tokenNames, rules, numRules, bitsets, ASTLabelType, superClass="Parser",
labelType="Token", members={<actions.parser.members>}) ::= <<
<genericParser(inputStreamType="ITokenStream", rewriteElementType="Token", ...)>
>>

/** How to generate a tree parser; same as parser except the input
 * stream is a different type.
 */
treeParser(grammar, name, scopes, tokens, tokenNames, globalAction, rules, numRules, bitsets,
labelType={<ASTLabelType>}, ASTLabelType="object", superClass="TreeParser",
members={<actions.treeparser.members>}, filterMode) ::= <<
<genericParser(inputStreamType="ITreeNodeStream", rewriteElementType="Node", ...)>
>>

/** A simpler version of a rule template that is specific to the imaginary
 * rules created for syntactic predicates. As they never have return values
 * nor parameters etc..., just give simplest possible method. Don't do
 * any of the normal memoization stuff in here either; it's a waste.
 * As predicates cannot be inlined into the invoking rule, they need to
 * be in a rule by themselves.
 */
synpredRule(ruleName, ruleDescriptor, block, description, nakedBlock) ::=
<<
// $ANTLR start "<ruleName>"
procedure
T<grammar.recognizerName>.<ruleName>_fragment(<ruleDescriptor.parameterScope:parameterScope(scope=it)>)
;
var
Alt: array [0..<grammar.numberOfDecisions>] of Integer;

```

```

<ruleLabelDefVars()>
begin
  <ruleLabelDefs()>
  <if(trace)>
    TraceIn('<ruleName>_fragment', <ruleDescriptor.index>);
  try
    <block>
  finally
    TraceOut('<ruleName>_fragment', <ruleDescriptor.index>);
  end;
  <else>
    <block>
  <endif>
end;
// $ANTLR end "<ruleName>"
>>

```

```

synpredDecls(name) ::= <<
SynPredPointer <name>;<\n>
>>

```

```

synpred(name) ::= <<

```

```

function T<grammar.recognizerName>.<name>: Boolean;
var
  Start: Integer;
  Success: Boolean;
begin
  State.Backtracking := State.Backtracking + 1;
  <@start()>
  Start := Input.Mark;
  try
    <name>_fragment(); // can never throw exception
  except
    on RE: ERecognitionException do
      WriteLn('Impossible: ' + RE.ToString);
    end;
  Success := not State.Failed;
  Input.Rewind(Start);
  <@stop()>
  State.Backtracking := State.Backtracking - 1;
  State.Failed := False;
  Result := Success;
end;<\n>
>>

```

```

lexerSynpred(name) ::= <<
<synpred(name)>

```

```

>>

lexerSynpredDeclaration(name) ::= <<
function <name>: Boolean;
procedure <name>_fragment;
>>

synpredDeclaration(name) ::= <<
function <name>: Boolean;
procedure <name>_fragment;
>>

ruleMemoization(name) ::= <<
<if(memoize)>
if ((State.Backtracking > 0) and AlreadyParsedRule(Input, <ruleDescriptor.index>)) then
  Exit(<ruleReturnValue(>);
<endif>
>>

/** How to test for failure and return from rule */
checkRuleBacktrackFailure() ::= <<
<if(backtracking)><\n>if (State.Failed) then Exit(<ruleReturnValue(>);<\n><endif>
>>

/** This rule has failed, exit indicating failure during backtrack */
ruleBacktrackFailure() ::= <<
<if(backtracking)>if (State.Backtracking > 0) then
begin
  State.Failed := True;
  Exit(<ruleReturnValue(>);
end;<endif>
>>

genericParserRuleDeclaration(rule, ruleDescriptor) ::= <<
<if(ruleDescriptor.isSynPred)>
<else>
<ruleAttributeScopeDeclaration(scope=ruleDescriptor.ruleScope)>
<returnScopeDeclaration(scope=ruleDescriptor.returnScope)>
public
<if(ruleDescriptor.hasMultipleReturnValues)>
function <rule.ruleName>: I<returnType(>);<\n>
<else>
<if(ruleDescriptor.hasSingleReturnValue)>
function <rule.ruleName>: <returnType(>);<\n>
<else>
procedure <rule.ruleName>;<\n>
<endif>
<endif>

```

```

<endif>
>>

genericParserRuleInterface(rule, ruleDescriptor) ::= <<
<if(ruleDescriptor.isSynPred)>
<else>
<if(ruleDescriptor.hasMultipleReturnValues)>
function <rule.ruleName>: I<returnType()>;<\n>
<else>
<if(ruleDescriptor.hasSingleReturnValue)>
function <rule.ruleName>: <returnType()>;<\n>
<else>
procedure <rule.ruleName>;<\n>
<endif>
<endif>
<endif>
>>

genericParserRuleReturnType(rule, ruleDescriptor) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(ruleDescriptor.isSynPred)>
<else>
I<returnType()> = interface(I<if(TREE_PARSER)>Tree<else>Parser<endif>RuleReturnScope)
end;<\n>
<endif>
<endif>
>>

/** How to generate code for a rule. This includes any return type
 * data aggregates required for multiple return values.
 */
rule(ruleName,ruleDescriptor,block,emptyRule,description,exceptions,finally,memoize) ::= <<
<ruleAttributeScope(scope=ruleDescriptor.ruleScope)>
<returnScope(scope=ruleDescriptor.returnScope)>

// $ANTLR start "<ruleName>"
(* <fileName>:<description> *)
<if(ruleDescriptor.hasMultipleReturnValues)>
function T<grammar.recognizerName>.<ruleName>(<ruleDescriptor.parameterScope:parameterScope(scope=it)>):
I<returnType()>;
<else>
<if(ruleDescriptor.hasSingleReturnValue)>
function T<grammar.recognizerName>.<ruleName>(<ruleDescriptor.parameterScope:parameterScope(scope=it)>):
<returnType()>;
<else>
procedure
T<grammar.recognizerName>.<ruleName>(<ruleDescriptor.parameterScope:parameterScope(scope=it)>);
<endif>

```

```

<endif>

var
<ruleDescriptor.actions.vars>
  Locals: TLocalStorage;
<if(ruleDescriptor.hasMultipleReturnValues)>
 RetVal: I<returnType()>;<\n>
<else>
<if(ruleDescriptor.hasSingleReturnValue)>
 RetVal: <returnType()>;<\n>
<else>
<endif>
<endif>
Alt: array [0..<grammar.numberOfDecisions>] of Integer;
<ruleDeclarationVars()>
<ruleLabelDefVars()>
begin
  Locals.Initialize;
  try
    <if(trace)>TraceIn('<ruleName>', <ruleDescriptor.index>);<endif>
    <ruleScopeSetUp()>
    <ruleDeclarations()>
    <ruleLabelDefs()>
    <ruleDescriptor.actions.init>
    <@preamble()>
    try
      try
        <ruleMemoization(name=ruleName)>
        <block>
        <ruleCleanUp()>
        <(ruleDescriptor.actions.after):execAction()>
      <if(exceptions)>
        <exceptions: { e|<catch(decl=e.decl,action=e.action)><\n}> }>
      <else>
      <if(!emptyRule)>
      <if(actions.(actionScope).rulecatch)>
        <actions.(actionScope).rulecatch>
      <else>
        except
          on RE: ERecognitionException do
            begin
              ReportError(RE);
              Recover(Input,RE);
              <@setErrorReturnValue()>
            end;<\n>
          <endif>
        <endif>
      <endif>
    <endif>
  <endif>

```

```

    end;
finally
    <if(trace)>TraceOut("<ruleName>", <ruleDescriptor.index>);<endif>
    <memoize()>
    <ruleScopeCleanUp()>
    <finally>
end;
<@postamble()>
finally
    Locals.Finalize;
end;
Exit(<ruleReturnValue()>);
end;
// $ANTLR end "<ruleName>"
>>

catch(decl,action) ::= <<
catch (<e.decl>)
{
    <e.action>
}
>>

ruleDeclarations() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
RetVal := T<returnType()>.Create;
RetVal.Start := Input.LT(1);<\n>
<else>
<ruleDescriptor.returnScope.attributes: { a |
<a.name> := <if(a.initValue)><a.initValue><else><initValue(a.type)><endif>;
}>
<endif>
<if(memoize)>
<ruleDescriptor.name>_startIndex := Input.Index();
<endif>
>>

ruleDeclarationVars() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<else>
<ruleDescriptor.returnScope.attributes: { a |
<a.name>: <a.type>;
}>
<endif>
<if(memoize)>
<ruleDescriptor.name>_startIndex: Integer;
<endif>
>>

```

```

ruleScopeSetUp() ::= <<
<ruleDescriptor.useScopes: { <it>Stack.Push(T<it>Scope.Create); }; separator="\n">
<ruleDescriptor.ruleScope: { <it.name>Stack.Push(T<it.name>Scope.Create); }; separator="\n">
>>

ruleScopeCleanUp() ::= <<
<ruleDescriptor.useScopes: { <it>Stack.Pop(); }; separator="\n">
<ruleDescriptor.ruleScope: { <it.name>Stack.Pop(); }; separator="\n">
>>

ruleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,ruleDescriptor.tokenListLabels]: { <it.label.text> := nil; }; separator="\n">
<[ruleDescriptor.tokenListLabels,ruleDescriptor.ruleListLabels]: { list_<it.label.text> := nil; }; separator="\n">
<ruleDescriptor.ruleLabels:ruleLabelDef(label=it); separator="\n">
<ruleDescriptor.ruleListLabels: { ll<ll.label.text> := nil; }; separator="\n">
>>

ruleLabelDefVars() ::= <<
<[ruleDescriptor.tokenLabels,ruleDescriptor.tokenListLabels]: { <it.label.text>: I<labelType>; }; separator="\n">
<[ruleDescriptor.tokenListLabels,ruleDescriptor.ruleListLabels]: { list_<it.label.text>: IList<IANTLRInterface>; };
separator="\n">
<ruleDescriptor.ruleLabels:ruleLabelDefVar(label=it); separator="\n">
<ruleDescriptor.ruleListLabels: { ll<ll.label.text>: <ruleLabelType(referencedRule=ll.referencedRule)>; };
separator="\n">
>>

lexerRuleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,
ruleDescriptor.tokenListLabels,
ruleDescriptor.ruleLabels]
: { <it.label.text> := nil; }; separator="\n"
>
<ruleDescriptor.charLabels: { int <it.label.text>; }; separator="\n">
<[ruleDescriptor.tokenListLabels,
ruleDescriptor.ruleListLabels,
ruleDescriptor.ruleListLabels]
: { List_<it.label.text> := nil; }; separator="\n"
>
>>

lexerRuleLabelDefDeclarations() ::= <<
<[ruleDescriptor.tokenLabels,
ruleDescriptor.tokenListLabels,
ruleDescriptor.ruleLabels]
: { <it.label.text>: I<labelType>; }; separator="\n"
>
<ruleDescriptor.charLabels: { int <it.label.text>; }; separator="\n">

```

```

<[ruleDescriptor.tokenListLabels,
ruleDescriptor.ruleListLabels,
ruleDescriptor.ruleListLabels]
  :{List_<it.label.text>: IList;}; separator="\n"
>
>>

```

```

ruleReturnValue() ::= <<
<if(!ruleDescriptor.isSynPred)>
<if(ruleDescriptor.hasReturnValue)>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnName>
<else>
RetVal
<endif>
<else>
<! nil !>
<endif>
<endif>
>>

```

```

ruleCleanUp() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(!TREE_PARSER)>
RetVal.Stop := Input.LT(-1);
<endif>
<endif>
>>

```

```

memoize() ::= <<
<if(memoize)>
<if(backtracking)>
if (State.Backtracking > 0) then
Memoize(Input, <ruleDescriptor.index>, <ruleDescriptor.name>_StartIndex);
<endif>
<endif>
>>

```

```

/** How to generate a rule in the lexer; naked blocks are used for
 * fragment rules.
 */

```

```

lexerRule(ruleName,nakedBlock,ruleDescriptor,block,memoize) ::= <<
// $ANTLR start "<ruleName>"
<ruleDescriptor.parameterScope>
procedure
T<grammar.recognizerName>.m<ruleName>(<ruleDescriptor.parameterScope:parameterScope(scope=it)>);
var
<ruleDescriptor.actions.vars>

```

```

Locals: TLocalStorage;
TokenType, Channel: Integer;
Alt: array [0..] of Integer;
<lexerRuleLabelDefDeclarations()>
begin
  Locals.Initialize;
  try
    <ruleAttributeScope(scope=ruleDescriptor.ruleScope)>
    <if(trace)>TraceIn("<ruleName>", <ruleDescriptor.index>);<endif>
    <ruleScopeSetUp()>
    <ruleDeclarations()>
    try
<if(nakedBlock)>
      <ruleMemoization(name=ruleName)>
      <lexerRuleLabelDefs()>
      <ruleDescriptor.actions.init>
      <block><\n>
<else>
      TokenType := <ruleName>;
      Channel := DEFAULT_TOKEN_CHANNEL;
      <ruleMemoization(name=ruleName)>
      <lexerRuleLabelDefs()>
      <ruleDescriptor.actions.init>
      <block>
      <ruleCleanUp()>
      State.TokenType := TokenType;
      State.Channel := Channel;
      <(ruleDescriptor.actions.after):execAction()>
<endif>
    finally
      <if(trace)>TraceOut("<ruleName>", <ruleDescriptor.index>);<endif>
      <ruleScopeCleanUp()>
      <memoize()>
    end;
  finally
    Locals.Finalize;
  end;
end;
// $ANTLR end "<ruleName>"
>>

/** How to generate code for the implicitly-defined lexer grammar rule
 * that chooses between lexer rules.
 */
tokensRule(ruleName,nakedBlock,args,block,ruleDescriptor) ::= <<
procedure T<grammar.recognizerName>.mTokens;
var
  Alt: array [0..] of Integer;

```

```

begin
  <block>
end;

procedure T<grammar.recognizerName>.DoTokens;
begin
  mTokens;
end;
>>

// S U B R U L E S

/** A (...) subrule with multiple alternatives */
block(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
(* <fileName>:<description> *)
Alt[<decisionNumber>] := <maxAlt>;
<decls>
<@predecision()>
<decision>
<@postdecision()>
<@prebranch()>
case Alt[<decisionNumber>] of
  <alts:altSwitchCase()>
end;
<@postbranch()>
>>

/** A rule block with multiple alternatives */
ruleBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
(* <fileName>:<description> *)
Alt[<decisionNumber>] := <maxAlt>;
<decls>
<@predecision()>
<decision>
<@postdecision()>
case Alt[<decisionNumber>] of
  <alts:altSwitchCase()>
end;
>>

ruleBlockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
(* <fileName>:<description> *)
<decls>
<@prealt()>
<alts>
<@postalt()>
>>

```

```

/** A special case of a (...) subrule with a single alternative */
blockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
(* <fileName>:<description> *)
<decls>
<@prealt()>
<alts>
<@postalt()>
>>

/** A (..)+ block with 1 or more alternatives */
positiveClosureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
(* <fileName>:<description> *)
FCnt[<decisionNumber>] := 0;
<decls>
<@preloop()>
while (True) do
begin
Alt[<decisionNumber>] := <maxAlt>;
<@predecision()>
<decision>
<@postdecision()>
case Alt[<decisionNumber>] of
<alts:altSwitchCase()>
else
begin
if (FCnt[<decisionNumber>] >= 1) then
Break;
<ruleBacktrackFailure()>
raise EEarlyExitException.Create(<decisionNumber>, Input);
<@earlyExitException()>
end;
end;
Inc(FCnt[<decisionNumber>]);
end;
<@postloop()>
>>

```

```

positiveClosureBlockSingleAlt ::= positiveClosureBlock

```

```

/** A (..)* block with 1 or more alternatives */
closureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::=
<<
(* <fileName>:<description> *)
<decls>
<@preloop()>
while (True) do
begin

```

```

Alt[<decisionNumber>] := <maxAlt>;
<@predecision()>
<decision>
<@postdecision()>
case Alt[<decisionNumber>] of
  <alts:altSwitchCase()>
else
  Break;
end;
end;
<@postloop()>
>>

closureBlockSingleAlt ::= closureBlock

/** Optional blocks (x)? are translated to (x|) by before code generation
 * so we can just use the normal block template
 */
optionalBlock ::= block

optionalBlockSingleAlt ::= block

/** A case in a switch that jumps to an alternative given the alternative
 * number. A DFA predicts the alternative and then a simple switch
 * does the jump to the code that actually matches that alternative.
 */
altSwitchCase() ::= <<
<i>:
<@prealt()>
<it><\n>
>>

/** An alternative is just a list of elements; at outermost level */
alt(elements,altNum,description,autoAST,outerAlt,treeLevel,rew) ::= <<
(* <fileName>:<description> *)
begin
<@declarations()>
<elements:element()>
<rew>
<@cleanup()>
end;
>>

/** What to emit when there is no rewrite. For auto build
 * mode, does nothing.
 */
noRewrite(rewriteBlockLevel, treeLevel) ::= ""

```

```
// ELEMENTS
```

```
/** Dump the elements one per line */
```

```
element() ::= <<  
<@prematch(>  
<it.el>  
>>
```

```
/** match a token optionally with a label in front */
```

```
tokenRef(token,label,elementIndex,terminalOptions) ::= <<  
<if(label)><label> := <endif>Match(Input, <token>,  
FOLLOW_<token>_in_<ruleName><elementIndex><if(label)> as  
I<labelType><endif>;<\n><checkRuleBacktrackFailure(>  
>>
```

```
/** ids+=ID */
```

```
tokenRefAndListLabel(token,label,elementIndex,terminalOptions) ::= <<  
<tokenRef(...)>  
<listLabel(elem=label,...)>  
>>
```

```
listLabel(label,elem) ::= <<
```

```
if (list_<label> = nil) then list_<label> := TList<IANTLRInterface>.Create;  
list_<label>.Add(<elem>);<\n>  
>>
```

```
/** match a character */
```

```
charRef(char,label) ::= <<  
<if(label)>  
<label> := Input.LA(1);<\n>  
<endif>  
Match(<char>); <checkRuleBacktrackFailure(>  
>>
```

```
/** match a character range */
```

```
charRangeRef(a,b,label) ::= <<  
<if(label)>  
<label> := Input.LA(1);<\n>  
<endif>  
MatchRange(<a>, <b>); <checkRuleBacktrackFailure(>  
>>
```

```
/** For now, sets are interval tests and must be tested inline */
```

```
matchSet(s,label,elementIndex,postmatchCode="") ::= <<  
<if(label)>  
<if(LEXER)>  
<label> := Input.LA(1);<\n>  
<else>
```

```

<label> := Input.LT(1) as I<labelType>;<\n>
<endif>
<endif>
if (<s>) then
begin
Input.Consume;
<postmatchCode>
<if(!LEXER)>
State.ErrorRecovery := False;<endif>
<if(backtracking)>State.Failed := False;<endif>
end
else
begin
<ruleBacktrackFailure()>
FException := EMismatchedSetException.Create(nil, Input);
<@mismatchedSetException()>
<if(LEXER)>
Recover(FException);
raise FException;<\n>
<else>
raise FException;
<! use following code to make it recover inline; remove throw mse;
RecoverFromMismatchedSet(input,mse,FOLLOW_set_in_<ruleName><elementIndex>);
!>
<endif>
end;<\n>
>>

```

```

matchRuleBlockSet ::= matchSet

```

```

matchSetAndListLabel(s,label,elementIndex,postmatchCode) ::= <<
<matchSet(...)>
<listLabel(elem=label,...)>
>>

```

```

/** Match a string literal */

```

```

lexerStringRef(string,label,elementIndex) ::= <<
<if(label)>
Locals.AsInteger['<label>Start'] := CharIndex;
Match(<string>); <checkRuleBacktrackFailure()>
<label> := TCommonToken.Create(Input, TToken.INVALID_TOKEN_TYPE, TToken.DEFAULT_CHANNEL,
Locals.AsInteger['<label>Start'], CharIndex-1);
<else>
Match(<string>); <checkRuleBacktrackFailure()>
<endif>
>>

```

```

wildcard(label,elementIndex) ::= <<

```

```

<if(label)>
<label> := Input.LT(1) as I<labelType>;<\n>
<endif>
MatchAny(input); <checkRuleBacktrackFailure()>
>>

wildcardAndListLabel(label,elementIndex) ::= <<
<wildcard(...)>
<listLabel(elem=label,...)>
>>

/** Match . wildcard in lexer */
wildcardChar(label, elementIndex) ::= <<
<if(label)>
<label> := Input.LA(1);<\n>
<endif>
MatchAny(); <checkRuleBacktrackFailure()>
>>

wildcardCharListLabel(label, elementIndex) ::= <<
<wildcardChar(...)>
<listLabel(elem=label,...)>
>>

/** Match a rule reference by invoking it possibly with arguments
 * and a return value or values. The 'rule' argument was the
 * target rule name, but now is type Rule, whose toString is
 * same: the rule name. Now though you can access full rule
 * descriptor stuff.
 */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
PushFollow(FOLLOW_<rule.name>_in_<ruleName><elementIndex>);
<if(label)>
<label> := <if(scope)><scope:delegateName()>.<endif><rule.name><args; separator=", ">;<\n>
<else>
<if(scope)>T<scope.recognizerName>(IANTLRObject(<scope:delegateName()>).Implementor).<endif><rule.name>
><args; separator=", ">;<\n>
<endif>
State.FollowingStackPointer := State.FollowingStackPointer - 1;
<checkRuleBacktrackFailure()>
>>

/** ids+=1 */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(elem=label,...)>
>>

```

```

/** A lexer rule reference.
 *
 * The 'rule' argument was the target rule name, but now
 * is type Rule, whose toString is same: the rule name.
 * Now though you can access full rule descriptor stuff.
 */
lexerRuleRef(rule,label,args,elementIndex,scope) ::= <<
<if(label)>
Locals.AsInteger['<label>Start<elementIndex>'] := CharIndex;
<if(scope)><scope:delegateName().<endif>m<rule.name>(<args; separator=", ">);
<checkRuleBacktrackFailure()>
<label> := TCommonToken.Create(Input, TToken.INVALID_TOKEN_TYPE, TToken.DEFAULT_CHANNEL,
Locals.AsInteger['<label>Start<elementIndex>'], CharIndex - 1);
<else>
<if(scope)><scope:delegateName().Implementor as T<scope.recognizerName>.<endif>m<rule.name>(<args;
separator=", ">); <checkRuleBacktrackFailure()>
<endif>
>>

/** i+=INT in lexer */
lexerRuleRefAndListLabel(rule,label,args,elementIndex,scope) ::= <<
<lexerRuleRef(...)>
<listLabel(elem=label,...)>
>>

/** EOF in the lexer */
lexerMatchEOF(label,elementIndex) ::= <<
<if(label)>
Locals.AsInteger['<label>Start<elementIndex>'] := CharIndex;
Match(EOF); <checkRuleBacktrackFailure()>
Locals['<label>'] := TCommonToken.Create(Input, EOF, TToken.DEFAULT_CHANNEL,
Locals.AsInteger['<label>Start<elementIndex>'], CharIndex-1);
<else>
Match(EOF); <checkRuleBacktrackFailure()>
<endif>
>>

/** match ^(root children) in tree parser */
tree(root, actionsAfterRoot, children, nullableChildList,
enclosingTreeLevel, treeLevel) ::= <<
<root:element()>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if (Input.LA(1) = TToken.DOWN) then
begin
Match(Input, TToken.DOWN, nil); <checkRuleBacktrackFailure()>
<children:element()>
Match(Input, TToken.UP, nil); <checkRuleBacktrackFailure()>

```

```

end;
<else>
Match(Input, TToken.DOWN, nil); <checkRuleBacktrackFailure()>
<children:element()>
Match(Input, TToken.UP, nil);<\n><checkRuleBacktrackFailure()>
<endif>
>>

/** Every predicate is used as a validating predicate (even when it is
 * also hoisted into a prediction expression).
 */
validateSemanticPredicate(pred,description) ::= <<
if (not (<evalPredicate(...)>)) then
begin
<ruleBacktrackFailure()>
raise EFailedPredicateException.Create(Input, '<ruleName>', '<description>');
end;<\n>
>>

// F i x e d D F A (if-then-else)

dfaState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
FLA[<decisionNumber>,<stateNumber>] := Input.LA(<k>);<\n>
<edges; separator="\nelse ">
else
begin
<if(eotPredictsAlt)>
Alt[<decisionNumber>] := <eotPredictsAlt>;<\n>
<else>
<ruleBacktrackFailure()>
raise ENoViableAltException.Create('<description>', <decisionNumber>, <stateNumber>, Input);<\n>
<endif>
end;
>>

/** Same as a normal DFA state except that we don't examine lookahead
 * for the bypass alternative. It delays error detection but this
 * is faster, smaller, and more what people expect. For (X)? people
 * expect "if ( LA(1)==X ) match(X);" and that's it.
 */
dfaOptionalBlockState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
FLA[<decisionNumber>,<stateNumber>] := Input.LA(<k>);<\n>
<edges; separator="\nelse ">;
>>

/** A DFA state that is actually the loopback decision of a closure
 * loop. If end-of-token (EOT) predicts any of the targets then it
 * should act like a default clause (i.e., no error can be generated).

```

```

* This is used only in the lexer so that for ('a')* on the end of a rule
* anything other than 'a' predicts exiting.
*/
dfaLoopbackState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
FLA[<decisionNumber>,<stateNumber>] := Input.LA(<k>);
<edges; separator="\nelse ">;<\n>
<if(eotPredictsAlt)>
<if(!edges)>
Alt[<decisionNumber>] := <eotPredictsAlt>; <! if no edges, don't gen ELSE !>
<else>
else
begin
  Alt[<decisionNumber>] := <eotPredictsAlt>;
end;<\n>
<endif>
<endif>
>>

/** An accept state indicates a unique alternative has been predicted */
dfaAcceptState(alt) ::= "Alt[<decisionNumber>] := <alt>;"

/** A simple edge with an expression. If the expression is satisfied,
* enter to the target state. To handle gated productions, we may
* have to evaluate some predicates for this edge.
*/
dfaEdge(labelExpr, targetState, predicates) ::= <<
if ((<labelExpr>)<if(predicates)> and (<predicates>)<endif>) then
begin
  <targetState>
end <! no ; here !>
>>

// F i x e d D F A (switch case)

/** A DFA state where a SWITCH may be generated. The code generator
* decides if this is possible: CodeGenerator.canGenerateSwitch().
*/
dfaStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
case Input.LA(<k>) of
  <edges; separator="\n">
else
begin
  <if(eotPredictsAlt)>
    Alt[<decisionNumber>] := <eotPredictsAlt>;
  <else>
    <ruleBacktrackFailure()>
    <@noViableAltException()>
    raise ENoViableAltException.Create('<description>', <decisionNumber>, <stateNumber>, Input);<\n>
  </if>
end
>>

```

```

<endif>
end;
end;<\n>
>>

dfaOptionalBlockStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
case Input.LA(<k>) of
  <edges; separator="\n">
end;<\n>
>>

dfaLoopbackStateSwitch(k, edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
case Input.LA(<k>) of
  <edges; separator="\n"><\n>
  <if(eotPredictsAlt)>
else
  Alt[<decisionNumber>] := <eotPredictsAlt>;<\n>
<endif>
end;<\n>
>>

dfaEdgeSwitch(labels, targetState) ::= <<
<labels:{<it>}; separator=",\n">:
begin
  <targetState>
end;
>>

// C y c l i c D F A

/** The code to initiate execution of a cyclic DFA; this is used
 * in the rule to predict an alt just like the fixed DFA case.
 * The <name> attribute is inherited via the parser, lexer, ...
 */
dfaDecision(decisionNumber,description) ::= <<
Alt[<decisionNumber>] := FDFA<decisionNumber>.Predict(Input);
>>

/* Dump DFA tables.
 */
cyclicDFADeclaration(dfa) ::= <<
strict protected
type
  TDFA<dfa.decisionNumber> = class(TDFA)
protected
  { IDFA }
  function Description: String; override;
public

```

```

    constructor Create(const ARecognizer: IBaseRecognizer);
end;
var
    FDFA<dfa.decisionNumber>: IDFA;
<if(dfa.specialStateSTs)>
strict protected
function DFA<dfa.decisionNumber>_SpecialStateTransition(const DFA: IDFA; S: Integer;
    const AInput: IIntStream): Integer;<endif>
>>

cyclicDFA(dfa) ::= <<
{ T<grammar.recognizerName>.T DFA<dfa.decisionNumber> }

constructor T<grammar.recognizerName>.T DFA<dfa.decisionNumber>.Create(const ARecognizer:
IBaseRecognizer);
const
    DFA<dfa.decisionNumber>_EOT = '<dfa.javaCompressedEOT; wrap=""+\n  "'>;
    DFA<dfa.decisionNumber>_EOF = '<dfa.javaCompressedEOF; wrap=""+\n  "'>;
    DFA<dfa.decisionNumber>_MIN = '<dfa.javaCompressedMin; wrap=""+\n  "'>;
    DFA<dfa.decisionNumber>_MAX = '<dfa.javaCompressedMax; wrap=""+\n  "'>;
    DFA<dfa.decisionNumber>_ACCEPT = '<dfa.javaCompressedAccept; wrap=""+\n  "'>;
    DFA<dfa.decisionNumber>_SPECIAL = '<dfa.javaCompressedSpecial; wrap=""+\n  "'>;
    DFA<dfa.decisionNumber>_TRANSITION: array [0..<length(dfa.javaCompressedTransition)>-1] of String = (
    <dfa.javaCompressedTransition: {s|<s; wrap=""+\n"'>}; separator=",\n">);
begin
inherited Create;
Recognizer := ARecognizer;
DecisionNumber := <dfa.decisionNumber>;
EOT := TDFA.UnpackEncodedString(DFA<dfa.decisionNumber>_EOT);
EOF := TDFA.UnpackEncodedString(DFA<dfa.decisionNumber>_EOF);
Min := TDFA.UnpackEncodedStringToUnsignedChars(DFA<dfa.decisionNumber>_MIN);
Max := TDFA.UnpackEncodedStringToUnsignedChars(DFA<dfa.decisionNumber>_MAX);
Accept := TDFA.UnpackEncodedString(DFA<dfa.decisionNumber>_ACCEPT);
Special := TDFA.UnpackEncodedString(DFA<dfa.decisionNumber>_SPECIAL);
Transition := TDFA.UnpackEncodedStringArray(DFA<dfa.decisionNumber>_TRANSITION);
end;

function T<grammar.recognizerName>.T DFA<dfa.decisionNumber>.Description: String;
begin
    Result := '<dfa.description>';
end;<\n>
<if(dfa.specialStateSTs)>
function T<grammar.recognizerName>.DFA<dfa.decisionNumber>_SpecialStateTransition(const DFA: IDFA; S:
Integer;
const AInput: IIntStream): Integer;
var
    Locals: TLocalStorage;
<if(LEXER)>

```

```

Input: IIntStream;
<endif>
<if(PARSER)>
Input: ITokenStream;
<endif>
<if(TREE_PARSER)>
Input: ITreeNodeStream;
<endif>
_S: Integer;
NVAE: ENoViableAltException;
begin
Result := -1;
Locals.Initialize;
try
  <if(LEXER)>
    Input := AInput;
  <endif>
  <if(PARSER)>
    Input := AInput as ITokenStream;
  <endif>
  <if(TREE_PARSER)>
    Input := AInput as ITreeNodeStream;
  <endif>
  _S := S;
  case S of
    <dfa.specialStateSTs:{state | <i0>: begin<! compressed special state numbers 0..n-1 !>
    <state> <\n> end;}; separator="\n">
    end;
  <if(backtracking)>
    if (State.Backtracking > 0) then
      begin
        State.Failed := True;
        Exit(-1);
      end;<\n>
    <endif>
    NVAE := ENoViableAltException.Create(DFA.Description, <dfa.decisionNumber>, _S, Input);
    DFA.Error(NVAE);
    raise NVAE;
  finally
    Locals.Finalize;
  end;
end;<\n>
<endif>
>>

/** A state in a cyclic DFA; it's a special state and part of a big switch on
 * state.
 */

```

```

cyclicDFAState(decisionNumber,stateNumber,edges,needErrorClause,semPredState) ::= <<
FLA[<decisionNumber>,<stateNumber>] := Input.LA(1);<\n>
<if(semPredState)> <! get next lookahead symbol to test edges, then rewind !>
Locals.AsInteger['index<decisionNumber>_<stateNumber>'] := Input.Index;
Input.Rewind;<\n>
<endif>
S := -1;
<edges; separator="\nelse ">;
<if(semPredState)> <! return input cursor to state before we rewound !>
Input.Seek(Locals.AsInteger['index<decisionNumber>_<stateNumber>']);<\n>
<endif>
if (S >= 0) then
  Exit(S);
>>

/** Just like a fixed DFA edge, test the lookahead and indicate what
 * state to jump to next if successful.
 */
cyclicDFAEdge(labelExpr, targetStateNumber, edgeNumber, predicates) ::= <<
if ((<labelExpr>)<if(predicates)> and (<predicates>)<endif>) then
  S := <targetStateNumber>
>>

/** An edge pointing at end-of-token; essentially matches any char;
 * always jump to the target.
 */
eotDFAEdge(targetStateNumber,edgeNumber, predicates) ::= <<
S := <targetStateNumber>;<\n>
>>

// D F A E X P R E S S I O N S

andPredicates(left,right) ::= "((<left>) and (<right>))"

orPredicates(operands) ::= "(<operands:{o|(<o>)}; separator=\" or \")"

notPredicate(pred) ::= "!(<evalPredicate(...)>)"

evalPredicate(pred,description) ::= "(<pred>)"

evalSynPredicate(pred,description) ::= "<pred>()"

lookaheadTest(atom,k,atomAsInt) ::= "FLA[<decisionNumber>,<stateNumber>] = <atomAsInt>"

/** Sometimes a lookahead test cannot assume that LA(k) is in a temp variable
 * somewhere. Must ask for the lookahead directly.
 */

```

```

isolatedLookaheadTest(atom,k,atomAsInt) ::= "Input.LA(<k>) = <atomAsInt>"

lookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= <<
((FLA[<decisionNumber>,<stateNumber>] \>= <lowerAsInt>) and (FLA[<decisionNumber>,<stateNumber>] \<=
<upperAsInt>))
>>

isolatedLookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= "(Input.LA(<k>) \>=
<lowerAsInt>) and (Input.LA(<k>) \<= <upperAsInt>)"

setTest(ranges) ::= "<ranges; separator=\\\" or (\\\">"

// A T T R I B U T E S

globalAttributeScope(scope) ::= <<
<scope.name>Stack := TStackList\<I<scope.name>Scope\>.Create;<n>
<endif>
>>

globalAttributeScopeDeclaration(scope) ::= <<
<if(scope.attributes)>
strict protected
type
  I<scope.name>Scope = interface(IANTLRObject)
  end;
  T<scope.name>Scope = class(TANTLRObject, I<scope.name>Scope)
  protected
    <scope.attributes:{<it.name>: <it.type>;}; separator="\\n">
  end;
strict protected
  <scope.name>Stack: IStackList\<I<scope.name>Scope\>;
<endif>
>>

ruleAttributeScopeDeclaration(scope) ::= <<
<if(scope.attributes)>
strict protected
type
  I<scope.name>Scope = interface(IANTLRObject)
  end;
  T<scope.name>Scope = class(TANTLRObject, I<scope.name>Scope)
  protected
    <scope.attributes:{<it.name>: <it.type>;}; separator="\\n">
  end;
strict protected
  <scope.name>Stack: IStackList\<I<scope.name>Scope\>;
<endif>
>>

```

```

ruleAttributeScope(scope) ::= <<
<! protected Stack <scope.name>Stack = new Stack();<\n !>
>>

ruleAttributeScopeInit(scope) ::= <<
<if(scope)>
<scope.name>Stack := TStackList<I<scope.name>Scope>.Create;<\n>
<endif>
>>

returnStructName() ::= "<it.name>_return"

returnType() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<ruleDescriptor:returnStructName()>
< ! I<if(TREE_PARSER)>Tree<else>Parser<endif>RuleReturnScope !>
<else>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnType>
<else>
<! Pointer/void !>
<endif>
<endif>
>>

/** Generate the C# type associated with a single or multiple return
* values.
*/
ruleLabelType(referencedRule) ::= <<
<if(referencedRule.hasMultipleReturnValues)>
I<referencedRule.name>_return
<else>
<if(referencedRule.hasSingleReturnValue)>
<referencedRule.singleValueReturnType>
<else>
void
<endif>
<endif>
>>

delegateName() ::= <<
<if(it.label)><it.label><else>g<it.name><endif>
>>

/** Using a type to init value map, try to init a type; if not in table
* must be an object, default value is "null".
*/

```

```

initValue(typeName) ::= <<
<csharpTypeInitMap.(typeName)>
>>

/** Define a rule label including default value */
ruleLabelDef(label) ::= <<
<label.label.text> := <initValue(typeName=ruleLabelType(referencedRule=label.referencedRule))>;<\n>
>>

ruleLabelDefVar(label) ::= <<
<label.label.text>: <ruleLabelType(referencedRule=label.referencedRule)>;
>>

/** Define a return struct for a rule if the code needs to access its
 * start/stop tokens, tree stuff, attributes, ... Leave a hole for
 * subgroups to stick in members.
 */
returnScope(scope) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
{ T<ruleDescriptor:returnStructName()> }

<scope.attributes:{public <it.decl>;}; separator="\n">
<@ruleReturnMembers()>
<endif>
>>

returnScopeDeclaration(scope) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
public
type
T<ruleDescriptor:returnStructName()> =
class(T<if(TREE_PARSER)>Tree<else>Parser<endif>RuleReturnScope, I<ruleDescriptor:returnStructName()>)
<scope.attributes:{public <it.decl>;}; separator="\n">
<@ruleReturnMembers()>
end;
<endif>
>>

parameterScope(scope) ::= <<
<scope.attributes:{<it.decl>;}; separator=", ">
>>

parameterAttributeRef(attr) ::= "<attr.name>"
parameterSetAttributeRef(attr,expr) ::= "<attr.name> := <expr>";

scopeAttributeRef(scope,attr,index,negIndex) ::= <<
<if(negIndex)>
(<scope>Stack[<scope>Stack.Count-<negIndex>-1] as T<scope>Scope).<attr.name>

```

```

<else>
<if(index)>
(<scope>Stack[<index>] as T<scope>Scope).<attr.name>
((<scope>_scope)<scope>_stack[<index>]).<attr.name>
<else>
(<scope>Stack.Peek.Implementor as T<scope>Scope).<attr.name>
<endif>
<endif>
>>

scopeSetAttributeRef(scope,attr,expr,index,negIndex) ::= <<
<if(negIndex)>
(<scope>Stack[<scope>Stack.Count-<negIndex>-1] as T<scope>Scope).<attr.name> := <expr>;<\n>
<else>
<if(index)>
(<scope>Stack[<index>] as T<scope>Scope).<attr.name> := <expr>;<\n>
<else>
(<scope>Stack.Peek.Implementor as T<scope>Scope).<attr.name> := <expr>;<\n>
<endif>
<endif>
>>

/** $x is either global scope or x is rule with dynamic scope; refers
 * to stack itself not top of stack. This is useful for predicates
 * like {$function.size()>0 && $function::name.equals("foo")}?
 */
isolatedDynamicScopeRef(scope) ::= "<scope>Stack"

/** reference an attribute of rule; might only have single return value */
ruleLabelRef(referencedRule,scope,attr) ::= <<
<if(referencedRule.hasMultipleReturnValues)>
(IfThen(Assigned(<scope>),Def(<scope>).<attr.name>,<initValue(attr.type)>))
<else>
<scope>
<endif>
>>

returnAttributeRef(ruleDescriptor,attr) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
RetVal.<attr.name>
<else>
<attr.name>
<endif>
>>

returnSetAttributeRef(ruleDescriptor,attr,expr) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
RetVal.<attr.name> := <expr>;

```

```

<else>
<attr.name> := <expr>;
<endif>
>>

/** How to translate $tokenLabel */
tokenLabelRef(label) ::= "<label>"

/** ids+=ID {$ids} or e+=expr {$e} */
listLabelRef(label) ::= "list_<label>"

// not sure the next are the right approach

tokenLabelPropertyRef_text(scope,attr) ::= "(Def(<scope>).Text)"
tokenLabelPropertyRef_type(scope,attr) ::= "(Def(<scope>).TokenType)"
tokenLabelPropertyRef_line(scope,attr) ::= "(Def(<scope>).Line)"
tokenLabelPropertyRef_pos(scope,attr) ::= "(Def(<scope>).CharPositionInLine)"
tokenLabelPropertyRef_channel(scope,attr) ::= "(Def(<scope>).Channel)"
tokenLabelPropertyRef_index(scope,attr) ::= "(Def(<scope>).TokenIndex)"
tokenLabelPropertyRef_tree(scope,attr) ::= "<scope>_tree"
tokenLabelPropertyRef_int(scope,attr) ::= "(StrToIntDef(Def(<scope>).Text,0))"

ruleLabelPropertyRef_start(scope,attr) ::= "(IfThen(Assigned(<scope>), Def(<scope>).Start, nil) as I<labelType>)"
ruleLabelPropertyRef_stop(scope,attr) ::= "(Def(<scope>).Stop as I<labelType>)"
ruleLabelPropertyRef_tree(scope,attr) ::= "(Def(Def(<scope>).Tree as I<ASTLabelType>))"
ruleLabelPropertyRef_text(scope,attr) ::= <<
<if(TREE_PARSER)>
IfThen(Assigned(<scope>), Input.TokenStream.ToString(
  Input.TreeAdaptor.GetTokenStartIndex(Def(<scope>).Start),
  Input.TreeAdaptor.GetTokenStopIndex(Def(<scope>).Start)), ")
<else>
IfThen(Assigned(<scope>), Input.ToString(
  (Def(<scope>).Start) as IToken,(Def(<scope>).Stop) as IToken), ")
<endif>
>>
ruleLabelPropertyRef_st(scope,attr) ::= "((<scope> != null) ? <scope>.ST : null)"

/** Isolated $RULE ref ok in lexer as it's a Token */
lexerRuleLabel(label) ::= "<label>"

lexerRuleLabelPropertyRef_type(scope,attr) ::= "(Def(<scope>).TokenType)"
lexerRuleLabelPropertyRef_line(scope,attr) ::= "(Def(<scope>).Line)"
lexerRuleLabelPropertyRef_pos(scope,attr) ::= "(IfThen(Assigned(<scope>),Def(<scope>).CharPositionInLine,-1))"
lexerRuleLabelPropertyRef_channel(scope,attr) ::= "(Def(<scope>).Channel)"
lexerRuleLabelPropertyRef_index(scope,attr) ::= "(Def(<scope>).TokenIndex)"
lexerRuleLabelPropertyRef_text(scope,attr) ::= "(Def(<scope>).Text)"
lexerRuleLabelPropertyRef_int(scope,attr) ::= "(StrToIntDef(Def(<scope>).Text,0))"

```

```

// Somebody may ref $template or $tree or $stop within a rule:
rulePropertyRef_start(scope,attr) ::= "(RetVal.Start as I<labelType>)"
rulePropertyRef_stop(scope,attr) ::= "(RetVal.Stop as I<labelType>)"
rulePropertyRef_tree(scope,attr) ::= "(RetVal.Tree as I<ASTLabelType>)"
rulePropertyRef_text(scope,attr) ::= <<
<if(TREE_PARSER)>
Input.TokenStream.ToString(
  Input.TreeAdaptor.GetTokenStartIndex(RetVal.Start),
  Input.TreeAdaptor.GetTokenStopIndex(RetVal.Start))
<else>
Input.ToString(RetVal.Start as IToken,Input.LT(-1))
<endif>
>>
rulePropertyRef_st(scope,attr) ::= "RetVal.ST"

lexerRulePropertyRef_text(scope,attr) ::= "Text"
lexerRulePropertyRef_type(scope,attr) ::= "TokenType"
lexerRulePropertyRef_line(scope,attr) ::= "State.TokenStartLine"
lexerRulePropertyRef_pos(scope,attr) ::= "State.TokenStartCharPositionInLine"
lexerRulePropertyRef_index(scope,attr) ::= "-1" // undefined token index in lexer
lexerRulePropertyRef_channel(scope,attr) ::= "Channel"
lexerRulePropertyRef_start(scope,attr) ::= "State.TokenStartCharIndex"
lexerRulePropertyRef_stop(scope,attr) ::= "(CharIndex-1)"
lexerRulePropertyRef_int(scope,attr) ::= "StrToInt(<scope>.Text)"

// setting $st and $tree is allowed in local rule. everything else
// is flagged as error
ruleSetPropertyRef_tree(scope,attr,expr) ::= "RetVal.Tree := <expr>;"
ruleSetPropertyRef_st(scope,attr,expr) ::= "RetVal.ST := <expr>;"

/** How to execute an action (only when not backtracking) */
execAction(action) ::= <<
<if(backtracking)>
<if(actions.(actionScope).synpredgate)>
if (<actions.(actionScope).synpredgate>) then
begin
  <action>
end;
<else>
if (State.Backtracking = 0) then
begin
  <action>
end;<\n>
<endif>
<else>
<action>

```

```
<endif>
```

```
>>
```

```
/** How to always execute an action even when backtracking */  
execForcedAction(action) ::= "<action>"
```

```
// M I S C (properties, etc...)
```

```
bitset(name, words64) ::= <<  
<name> := TBitSet.Create([<words64: {<it>}separator=",">]);<n>  
>>
```

```
bitsetDecl(name) ::= <<  
<name>: IBitSet;<n>  
>>
```

```
codeFileExtension() ::= ".pas"
```

```
true() ::= "True"  
false() ::= "False"
```

```
Found in path(s):
```

```
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-  
jar/org/antlr/codegen/templates/Delphi/Delphi.stg
```

```
No license file was found, but licenses were detected in source scan.
```

```
/*
```

```
* [The "BSD license"]
```

```
* Copyright (c) 2005-2008 Terence Parr
```

```
* All rights reserved.
```

```
*
```

```
* Conversion to C#:
```

```
* Copyright (c) 2008-2009 Sam Harwell, Pixel Mine, Inc.
```

```
* All rights reserved.
```

```
*
```

```
* Redistribution and use in source and binary forms, with or without
```

```
* modification, are permitted provided that the following conditions
```

```
* are met:
```

```
* 1. Redistributions of source code must retain the above copyright
```

```
* notice, this list of conditions and the following disclaimer.
```

```
* 2. Redistributions in binary form must reproduce the above copyright
```

```
* notice, this list of conditions and the following disclaimer in the
```

```
* documentation and/or other materials provided with the distribution.
```

```
* 3. The name of the author may not be used to endorse or promote products
```

```
* derived from this software without specific prior written permission.
```

```
*
```

```
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
```

```
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/
```

```
/** Template overrides to add debugging to AST stuff. Dynamic inheritance
* hierarchy is set up as ASTDbg : AST : Dbg : Java by code generator.
*/
```

```
parserMembers() ::= <<
// Implement this function in your helper file to use a custom tree adaptor
partial void InitializeTreeAdaptor();
protected DebugTreeAdaptor adaptor;

public ITreeAdaptor TreeAdaptor
{
    get
    {
        return adaptor;
    }
    set
    {
<if(grammar.grammarIsRoot)>
        this.adaptor = new DebugTreeAdaptor(dbg,adaptor);
<else>
        this.adaptor = (DebugTreeAdaptor)adaptor; // delegator sends dbg adaptor
<endif><\n>
        <grammar.directDelegates:{g|<g.delegateName()>.TreeAdaptor = this.adaptor;}>
    }
}<\n>
>>
```

```
parserCtorBody() ::= <<
<super.parserCtorBody()>
>>
```

```
createListenerAndHandshake() ::= <<
DebugEventSocketProxy proxy = new DebugEventSocketProxy( this, port,
<if(TREE_PARSER)>input.TreeAdaptor<else>adaptor<endif> );
DebugListener = proxy;
<inputStreamType> = new Debug<inputStreamType>( input, proxy );
try
```

```

{
    proxy.Handshake();
}
catch ( IOException ioe )
{
    ReportError( ioe );
}
>>

```

```

@ctorForRootGrammar.finally() ::= <<
ITreeAdaptor adap = new CommonTreeAdaptor();
TreeAdaptor = adap;
proxy.TreeAdaptor = adap;
>>

```

```

@ctorForProfilingRootGrammar.finally() ::= <<
ITreeAdaptor adap = new CommonTreeAdaptor();
TreeAdaptor = adap;
>>

```

```

@ctorForPredefinedListener.superClassRef() ::= " : base( input, dbg )"

```

```

@ctorForPredefinedListener.finally() ::= <<
<if(grammar.grammarIsRoot)><<! don't create new adaptor for delegates !>
ITreeAdaptor adap = new CommonTreeAdaptor();
TreeAdaptor = adap;<\n>
<endif>
>>

```

```

//@rewriteElement.pregen() ::= "dbg.Location( <e.line>, <e.pos> );"

```

Found in path(s):

```

* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-
jar/org/antlr/codegen/templates/CSharp3/ASTDbg.stg

```

No license file was found, but licenses were detected in source scan.

/*

* [The "BSD license"]

* Copyright (c) 2007-2008 Johannes Luber

* Copyright (c) 2005-2007 Kunle Odutola

* Copyright (c) 2011 Sam Harwell

* Copyright (c) 2011 Terence Parr

* All rights reserved.

*

* Redistribution and use in source and binary forms, with or without

* modification, are permitted provided that the following conditions

* are met:

* 1. Redistributions of source code must retain the above copyright

```

* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products
* derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

```

```

/** Templates for building ASTs during normal parsing.

```

```

*
* Deal with many combinations. Dimensions are:
* Auto build or rewrite
* no label, label, list label (label/no-label handled together)
* child, root
* token, set, rule, wildcard
*
* The situation is not too bad as rewrite (->) usage makes ^ and !
* invalid. There is no huge explosion of combinations.
*/

```

```

@rule.setErrorReturnValue() ::= <<
retval.Tree = (<ASTLabelType>)adaptor.ErrorNode(input, retval.Start, input.LT(-1), re);
<! System.out.WriteLine("<ruleName> returns "+((CommonTree)retval.tree).toStringTree()); !>
>>

```

```

// TOKEN AST STUFF

```

```

/** ID and output=AST */
tokenRef(token,label,elementIndex,terminalOptions={}) ::= <<
<super.tokenRef(...)>
<if(backtracking)>if (state.backtracking == 0) {<endif>
<label>_tree = <createNodeFromToken(...)>;
adaptor.AddChild(root_0, <label>_tree);
<if(backtracking)>}<endif>
>>

```

```

/** ID! and output=AST (same as plain tokenRef) */

```

```

tokenRefBang(token,label,elementIndex) ::= "<super.tokenRef(...)>"

/** ID^ and output=AST */
tokenRefRuleRoot(token,label,elementIndex,terminalOptions={}) ::= <<
<super.tokenRef(...)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) {<endif>
<label>_tree = <createNodeFromToken(...)>;
root_0 = (<ASTLabelType>)adaptor.BecomeRoot(<label>_tree, root_0);
<if(backtracking)>}<endif>
>>

/** ids+=ID! and output=AST */
tokenRefBangAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRefBang(...)>
<listLabelElem(elem=label,elemType=labelType,...)>
>>

/** label+=TOKEN when output=AST but not rewrite alt */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRef(...)>
<listLabelElem(elem=label,elemType=labelType,...)>
>>

/** Match label+=TOKEN^ when output=AST but not rewrite alt */
tokenRefRuleRootAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRefRuleRoot(...)>
<listLabelElem(elem=label,elemType=labelType,...)>
>>

// SET AST

// the match set stuff is interesting in that it uses an argument list
// to pass code to the default matchSet; another possible way to alter
// inherited code. I don't use the region stuff because I need to pass
// different chunks depending on the operator. I don't like making
// the template name have the operator as the number of templates gets
// large but this is the most flexible--this is as opposed to having
// the code generator call matchSet then add root code or ruleroot code
// plus list label plus ... The combinations might require complicated
// rather than just added on code. Investigate that refactoring when
// I have more time.

matchSet(s,label,elementIndex,postmatchCode,terminalOptions={}) ::= <<
<super.matchSet(postmatchCode={<if(backtracking)>if (<actions.(actionScope).synpredgate>
<endif>adaptor.AddChild(root_0, <createNodeFromToken(...)>);}, ...)>
>>

matchRuleBlockSet(s,label,elementIndex,postmatchCode,treeLevel="0",terminalOptions={}) ::= <<

```

```

<matchSet(...)>
>>

matchSetBang(s,label,elementIndex,postmatchCode,terminalOptions={}) ::= "<super.matchSet(...)>"

// note there is no matchSetTrack because -> rewrites force sets to be
// plain old blocks of alts: (A|B|...|C)

matchSetRuleRoot(s,label,elementIndex,debug,terminalOptions={}) ::= <<
<if(label)>
<label>=(<labelType>)input.LT(1);
<endif>
<super.matchSet(postmatchCode={<if(backtracking)>if (<actions.(actionScope).synpredgate>) <endif>root_0 =
(<ASTLabelType>)adaptor.BecomeRoot(<createNodeFromToken(...)>, root_0);}, ...)>
>>

// RULE REF AST

/** rule when output=AST */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) <endif>adaptor.AddChild(root_0, <label>.Tree);
>>

/** rule! is same as normal rule ref */
ruleRefBang(rule,label,elementIndex,args,scope) ::= "<super.ruleRef(...)>"

/** rule^ */
ruleRefRuleRoot(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) <endif>root_0 =
(<ASTLabelType>)adaptor.BecomeRoot(<label>.Tree, root_0);
>>

/** x+=rule when output=AST */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabelElem(elem={<label>.Tree},elemType=ASTLabelType,...)>
>>

/** x+=rule! when output=AST is a rule ref with list addition */
ruleRefBangAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefBang(...)>
<listLabelElem(elem={<label>.Tree},elemType=ASTLabelType,...)>
>>

/** x+=rule^ */
ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope) ::= <<

```

```

<ruleRefRuleRoot(...)>
<listLabelElem(elem={<label>.Tree},elemType=ASTLabelType,...)>
>>

// WILDCARD AST

wildcard(token,label,elementIndex,terminalOptions={}) ::= <<
<super.wildcard(...)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) {<endif>
<label>_tree = (<ASTLabelType>)adaptor.Create(<label>);
adaptor.AddChild(root_0, <label>_tree);
<if(backtracking)>}<endif>
>>

wildcardBang(label,elementIndex) ::= "<super.wildcard(token=[],...)>"

wildcardRuleRoot(token,label,elementIndex,terminalOptions={}) ::= <<
<super.wildcard(...)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) {<endif>
<label>_tree = (<ASTLabelType>)adaptor.Create(<label>);
root_0 = (<ASTLabelType>)adaptor.BecomeRoot(<label>_tree, root_0);
<if(backtracking)>}<endif>
>>

createNodeFromToken(label,terminalOptions={}) ::= <%
<if(terminalOptions.node)>
new
<terminalOptions.node>(<if(terminalOptions.type)><terminalOptions.type>,<endif><label><if(terminalOptions.text)>,<terminalOptions.text; format="string"><endif>)
<else>
(<ASTLabelType>)adaptor.Create(<if(terminalOptions.type)><terminalOptions.type>,<endif><label><if(terminalOptions.text)>,<terminalOptions.text; format="string"><endif>)
<endif>
%>

ruleCleanUp() ::= <<
<super.ruleCleanUp()>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) {<endif>
retval.Tree = (<ASTLabelType>)adaptor.RulePostProcessing(root_0);
adaptor.SetTokenBoundaries(retval.Tree, retval.Start, retval.Stop);
<if(backtracking)>}<endif>
>>

```

Found in path(s):

* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/CSharp2/ASTParser.stg

No license file was found, but licenses were detected in source scan.

/*

[The "BSD license"]

Copyright (c) 2005-2006 Terence Parr

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

/** Templates for building ASTs during normal parsing.

*

* Deal with many combinations. Dimensions are:

* Auto build or rewrite

* no label, label, list label (label/no-label handled together)

* child, root

* token, set, rule, wildcard

*

* The situation is not too bad as rewrite (->) usage makes ^ and !

* invalid. There is no huge explosion of combinations.

*/

```
finishedBacktracking(block) ::= <<
```

```
<if(!ruleDescriptor.isSynPred)>
```

```
<if(backtracking)>
```

```
if <actions.(actionScope).synpredgate>:
```

```
  <block>
```

```
<else>
```

```
<block>
```

```
<endif>
```

```

<endif>
>>

@ruleBody.setErrorReturnValue() ::= <<
retval.tree = self._adaptor.errorNode(self.input, retval.start, self.input.LT(-1), re)
>>

// TOKEN AST STUFF

/** ID and output=AST */
tokenRef(token,label,elementIndex,terminalOptions={ }) ::= <<
<super.tokenRef(...)>
<finishedBacktracking({
<label>_tree = <createNodeFromToken(...)>
self._adaptor.addChild(root_0, <label>_tree)
})>
>>

/** ID! and output=AST (same as plain tokenRef) */
tokenRefBang(token,label,elementIndex,terminalOptions={ }) ::= "<super.tokenRef(...)>"

/** ID^ and output=AST */
tokenRefRuleRoot(token,label,elementIndex,terminalOptions={ }) ::= <<
<super.tokenRef(...)>
<finishedBacktracking({
<label>_tree = <createNodeFromToken(...)>
root_0 = self._adaptor.becomeRoot(<label>_tree, root_0)
})>
>>

/** ids+=ID! and output=AST */
tokenRefBangAndListLabel(token,label,elementIndex,terminalOptions={ }) ::= <<
<tokenRefBang(...)>
<listLabel(elem=label,...)>
>>

/** label+=TOKEN when output=AST but not rewrite alt */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions={ }) ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

/** Match label+=TOKEN^ when output=AST but not rewrite alt */
tokenRefRuleRootAndListLabel(token,label,elementIndex,terminalOptions={ }) ::= <<
<tokenRefRuleRoot(...)>
<listLabel(elem=label,...)>
>>

```

```

// SET AST

// the match set stuff is interesting in that it uses an argument list
// to pass code to the default matchSet; another possible way to alter
// inherited code. I don't use the region stuff because I need to pass
// different chunks depending on the operator. I don't like making
// the template name have the operator as the number of templates gets
// large but this is the most flexible--this is as opposed to having
// the code generator call matchSet then add root code or ruleroot code
// plus list label plus ... The combinations might require complicated
// rather than just added on code. Investigate that refactoring when
// I have more time.

matchSet(s,label,elementIndex,postmatchCode,terminalOptions={}) ::= <%
<super.matchSet(postmatchCode={<finishedBacktracking({self._adaptor.addChild(root_0,
<createNodeFromToken(...)>})>}, ...)>
%>

matchRuleBlockSet(s,label,elementIndex,postmatchCode,treeLevel="0",terminalOptions={}) ::= <<
<matchSet(...)>
>>

matchSetBang(s,label,elementIndex,postmatchCode,terminalOptions={}) ::= "<super.matchSet(...)>"

// note there is no matchSetTrack because -> rewrites force sets to be
// plain old blocks of alts: (A|B|...|C)

matchSetRuleRoot(s,label,elementIndex,debug,terminalOptions={}) ::= <<
<if(label)>
<label> = self.input.LT(1)<\n>
<endif>
<super.matchSet(postmatchCode={<finishedBacktracking({root_0 =
self._adaptor.becomeRoot(<createNodeFromToken(...)>, root_0)})>}, ...)>
>>

// RULE REF AST

/** rule when output=AST */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<finishedBacktracking({self._adaptor.addChild(root_0, <label>.tree)})>
>>

/** rule! is same as normal rule ref */
ruleRefBang(rule,label,elementIndex,args,scope) ::= "<super.ruleRef(...)>"

/** rule^ */
ruleRefRuleRoot(rule,label,elementIndex,args,scope) ::= <<

```

```

<super.ruleRef(...)>
<finishedBacktracking({root_0 = self._adaptor.becomeRoot(<label>.tree, root_0)})>
>>

/** x+=rule when output=AST */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(label, {<label>.tree})>
>>

/** x+=rule! when output=AST is a rule ref with list addition */
ruleRefBangAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefBang(...)>
<listLabel(label, {<label>.tree})>
>>

/** x+=rule^ */
ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<listLabel(label, {<label>.tree})>
>>

// WILDCARD AST

wildcard(token,label,elementIndex,terminalOptions={ }) ::= <<
<super.wildcard(...)>
<finishedBacktracking({
<label>_tree = self._adaptor.createWithPayload(<label>)
self._adaptor.addChild(root_0, <label>_tree)
})>
>>

wildcardBang(label,elementIndex) ::= "<super.wildcard(...)>"

wildcardRuleRoot(token,label,elementIndex,terminalOptions={ }) ::= <<
<super.wildcard(...)>
<finishedBacktracking({
<label>_tree = self._adaptor.createWithPayload(<label>)
root_0 = self._adaptor.becomeRoot(<label>_tree, root_0)
})>
>>

createNodeFromToken(label,terminalOptions={ }) ::= <%
<if(terminalOptions.node)>
<terminalOptions.node>(<label>) <! new MethodNode(IDLabel) !>
<else>
self._adaptor.createWithPayload(<label>)
<endif>

```

%>

```
ruleCleanUp() ::= <<
<super.ruleCleanUp()>
<finishedBacktracking({
retval.tree = self._adaptor.rulePostProcessing(root_0)
self._adaptor.setTokenBoundaries(retval.tree, retval.start, retval.stop)
})>
>>
```

Found in path(s):

* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/Python/ASTParser.stg

No license file was found, but licenses were detected in source scan.

/*

[The "BSD license"]

Copyright (c) 2008 Erik van Bilzen

Copyright (c) 2007-2008 Johannes Luber

Copyright (c) 2005-2007 Kunle Odutola

Copyright (c) 2005 Terence Parr

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

group AST;

@outputFile.imports() ::= <<

```

<@super.imports(><if(!TREE_PARSER)><! tree parser would already have imported !>
Antlr.Runtime.Tree,<\n><endif>
>>

@genericParser.members() ::= <<
<@super.members(>
<parserMembers(>
>>

@genericParser.membersConstructor() ::= <<
<@super.membersConstructor(>
<parserMembersConstructor(>
>>

@genericParser.membersImplementation() ::= <<
<@super.membersImplementation(>
<parserMembersImplementation(>
>>

/** Add an adaptor property that knows how to build trees */
parserMembers() ::= <<
strict protected
  FAdaptor: ITreeAdaptor;
  procedure SetAdaptor(const Value: ITreeAdaptor);
  property Adaptor: ITreeAdaptor read FAdaptor;
public
  property TreeAdaptor: ITreeAdaptor read FAdaptor write SetAdaptor;

>>

parserMembersConstructor() ::= <<
FAdaptor := TCommonTreeAdaptor.Create;
>>

parserMembersImplementation() ::= <<
procedure T<grammar.recognizerName>.SetAdaptor(const Value: ITreeAdaptor);
begin
  FAdaptor := Value;
  <grammar.directDelegates:{g|<g:delegateName(>.TreeAdaptor := FAdaptor;}>
end;
>>

@returnScope.ruleReturnMembers() ::= <<
function T<grammar.recognizerName>.T<ruleDescriptor:returnStructName(>.GetTree: IANTLRInterface;
begin
  Result := FTree;
end;

```

```

procedure T<grammar.recognizerName>.T<ruleDescriptor:returnStructName()>.SetTree(const Value:
IANTLRInterface);
begin
  FTree := Value as I<ASTLabelType>;
end;
>>

@returnScopeDeclaration.ruleReturnMembers() ::= <<
strict private
  FTree: I<ASTLabelType>;
protected
  { IRuleReturnScope }
  function GetTree: IANTLRInterface; override;
  procedure SetTree(const Value: IANTLRInterface); override;
>>

/** Add a variable to track rule's return AST */
ruleDeclarations() ::= <<
<super.ruleDeclarations()>
Root[0] := nil;<\n>
>>

ruleDeclarationVars() ::= <<
<super.ruleDeclarationVars()>
Root: array [0..63] of I<ASTLabelType>;
>>

ruleLabelDefs() ::= <<
<super.ruleLabelDefs()>
<ruleDescriptor.tokenLabels: {<it.label.text>_tree := nil;}; separator="\n">
<ruleDescriptor.tokenListLabels: {<it.label.text>_tree := nil;}; separator="\n">
<ruleDescriptor.allTokenRefsInAltsWithRewrites: {Locals['Stream_<it>'] :=
TRewriteRule<rewriteElementType>Stream.Create(Adaptor,'token <it>');}; separator="\n">
<ruleDescriptor.allRuleRefsInAltsWithRewrites: {Locals['Stream_<it>'] :=
TRewriteRuleSubtreeStream.Create(Adaptor,'rule <it>');}; separator="\n">
>>

ruleLabelDefVars() ::= <<
<super.ruleLabelDefVars()>
<ruleDescriptor.tokenLabels: {<it.label.text>_tree: I<ASTLabelType>;}; separator="\n">
<ruleDescriptor.tokenListLabels: {<it.label.text>_tree: I<ASTLabelType>;}; separator="\n">
>>
/** When doing auto AST construction, we must define some variables;
 * These should be turned off if doing rewrites. This must be a "mode"
 * as a rule could have both rewrite and AST within the same alternative
 * block.
 */
@alt.declarations() ::= <<

```

```

<if(autoAST)>
<if(outerAlt)>
<if(!rewriteMode)>
Root[0] := Adaptor.GetNilNode as I<ASTLabelType>;
<endif>
<endif>
<endif>
>>

// Tracking Rule Elements

/** ID and track it for use in a rewrite rule */
tokenRefTrack(token,label,elementIndex,terminalOptions) ::= <<
<tokenRefBang(...)> <! Track implies no auto AST construction!>
<if(backtracking)>if (State.Backtracking = 0) then <endif>(Locals['Stream_<token>'] as
IRewriteRuleElementStream).Add(<label>);<\n>
>>

/** ids+=ID and track it for use in a rewrite rule; adds to ids *and*
 * to the tracking list stream_ID for use in the rewrite.
 */
tokenRefTrackAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<tokenRefTrack(...)>
<listLabel(elem=label,...)>
>>

/** ^(ID ...) track for rewrite */
tokenRefRuleRootTrack(token,label,elementIndex,terminalOptions) ::= <<
<tokenRefBang(...)>
<if(backtracking)>if (State.Backtracking = 0) then <endif>(Locals['Stream_<token>'] as
IRewriteRuleElementStream).Add(<label>);<\n>
>>

/** Match ^(label+=TOKEN ...) track for rewrite */
tokenRefRuleRootTrackAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<tokenRefRuleRootTrack(...)>
<listLabel(elem=label,...)>
>>

wildcardTrack(label,elementIndex) ::= <<
<super.wildcard(...)>
>>

/** rule when output=AST and tracking for rewrite */
ruleRefTrack(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<if(backtracking)>if (State.Backtracking = 0) then <endif>(Locals['Stream_<rule.name>'] as
IRewriteRuleElementStream).Add(<label>.Tree);<\n>

```

```

>>

/** x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefTrack(...)>
<listLabel(elem=label+".Tree",...)>
>>

/** ^(rule ...) rewrite */
ruleRefRuleRootTrack(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<if(backtracking)>if (State.Backtracking = 0) then <endif>(Locals['Stream_<rule>'] as
IRewriteRuleElementStream).Add(<label>.Tree);
>>

/** ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRootTrack(...)>
<listLabel(elem=label+".Tree",...)>
>>

// R e w r i t e

rewriteCode(
alts, description,
referencedElementsDeep, // ALL referenced elements to right of ->
referencedTokenLabels,
referencedTokenListLabels,
referencedRuleLabels,
referencedRuleListLabels,
referencedWildcardLabels,
referencedWildcardListLabels,
rewriteBlockLevel, enclosingTreeLevel, treeLevel) ::=
<<

// AST REWRITE
// elements: <referencedElementsDeep; separator=", ">
// token labels: <referencedTokenLabels; separator=", ">
// rule labels: <referencedRuleLabels; separator=", ">
// token list labels: <referencedTokenListLabels; separator=", ">
// rule list labels: <referencedRuleListLabels; separator=", ">
<if(backtracking)>
if (State.Backtracking = 0) then
begin<\n>
<endif>
<prevRuleRootRef().Tree := Root[0];
<rewriteCodeLabels()>
Root[0] := Adaptor.GetNilNode as I<ASTLabelType>;

```

```

<alts:rewriteAlt(); separator="else ">
<! if tree parser and rewrite=true !>
<if(TREE_PARSER)>
<if(rewriteMode)>
<prevRuleRootRef().Tree = (<ASTLabelType>)adaptor.rulePostProcessing(root[0]);
input.ReplaceChildren(adaptor.GetParent(retval.Start),
    adaptor.GetChildIndex(retval.Start),
    adaptor.GetChildIndex(_last),
    retval.Tree);
<endif>
<endif>
<! if parser or rewrite!=true, we need to set result !>
<if(!TREE_PARSER)>
<prevRuleRootRef().Tree := Root[0];<\n>
<endif>
<if(!rewriteMode)>
<prevRuleRootRef().Tree := Root[0];<\n>
<endif>
<if(backtracking)>
end;
<endif>
>>

rewriteCodeLabels() ::= <<
<referencedTokenLabels
: {Locals['Stream_<it>'] := TRewriteRule<rewriteElementType>Stream.Create(Adaptor, 'token <it>', <it>);};
separator="\n"
>
<referencedTokenListLabels
: {Locals['Stream_<it>'] := TRewriteRule<rewriteElementType>Stream.Create(Adaptor, 'token <it>', list_<it>);};
separator="\n"
>
<referencedRuleLabels: {
if Assigned(<it>) then
Locals['Stream_<it>'] := TRewriteRuleSubtreeStream.Create(Adaptor, 'token <it>', <it>.Tree)
else
Locals['Stream_<it>'] := TRewriteRuleSubtreeStream.Create(Adaptor, 'token <it>', nil);}; separator="\n">
<referencedRuleListLabels
: {Locals['Stream_<it>'] := TRewriteRuleSubtreeStream.Create(Adaptor, 'token <it>', list_<it>);};
separator="\n"
>
>>

/** Generate code for an optional rewrite block; note it uses the deep ref'd element
* list rather shallow like other blocks.
*/
rewriteOptionalBlock(
alt,rewriteBlockLevel,

```

```

referencedElementsDeep, // all nested refs
referencedElements, // elements in immediately block; no nested blocks
description) ::=
<<
(* <fileName>:<description> *)
if (<referencedElementsDeep:{el | (Locals['Stream_<el>'] as IRewriteRuleElementStream).HasNext}; separator="
or ">) then
begin
<alt>
end;
<referencedElementsDeep:{el | (Locals['Stream_<el>'] as IRewriteRuleElementStream).Reset;<\n>}>
>>

rewriteClosureBlock(
alt,rewriteBlockLevel,
referencedElementsDeep, // all nested refs
referencedElements, // elements in immediately block; no nested blocks
description) ::=
<<
(* <fileName>:<description> *)
while (<referencedElements:{el | (Locals['Stream_<el>'] as IRewriteRuleElementStream).HasNext}; separator=" or
">) do
begin
<alt>
end;
<referencedElements:{el | (Locals['Stream_<el>'] as IRewriteRuleElementStream).Reset();<\n>}>
>>

rewritePositiveClosureBlock(
alt,rewriteBlockLevel,
referencedElementsDeep, // all nested refs
referencedElements, // elements in immediately block; no nested blocks
description) ::=
<<
if (not (<referencedElements:{el | (Locals['Stream_<el>'] as IRewriteRuleElementStream).HasNext}; separator=" or
">)) then
raise ERewriteEarlyExitException.Create("");

while (<referencedElements:{el | (Locals['Stream_<el>'] as IRewriteRuleElementStream).HasNext}; separator=" or
">) do
begin
<alt>
end;
<referencedElements:{el | (Locals['Stream_<el>'] as IRewriteRuleElementStream).Reset();<\n>}>
>>

rewriteAlt(a) ::= <<
(* <a.description> *)

```

```

<if(a.pred)>
if (<a.pred>) then
begin
  <a.alt>
end<\n>
<else>
begin
  <a.alt>
end;<\n>
<endif>
>>

/** For empty rewrites: "r : ... -> ;" */
rewriteEmptyAlt() ::= "Root[0] = null;"

rewriteTree(root,children,description,enclosingTreeLevel,treeLevel) ::= <<
(* <fileName>:<description> *)
begin
  Root[<treeLevel>] := Adaptor.GetNilNode as I<ASTLabelType>;
  <root:rewriteElement()>
  <children:rewriteElement()>
  Adaptor.AddChild(Root[<enclosingTreeLevel>], Root[<treeLevel>]);
end;<\n>
>>

rewriteElementList(elements) ::= "<elements:rewriteElement()>"

rewriteElement(e) ::= <<
<@pregen()>
<e.el>
>>

/** Gen ID or ID[args] */
rewriteTokenRef(token,elementIndex,terminalOptions,args) ::= <<
Adaptor.AddChild(Root[<treeLevel>], <createRewriteNodeFromElement(...)>);<\n>
>>

/** Gen $label ... where defined via label=ID */
rewriteTokenLabelRef(label,elementIndex) ::= <<
Adaptor.AddChild(Root[<treeLevel>], (Locals['Stream_<label>'] as
IRewriteRuleElementStream).NextNode());<\n>
>>

/** Gen $label ... where defined via label+=ID */
rewriteTokenListLabelRef(label,elementIndex) ::= <<
Adaptor.AddChild(Root[<treeLevel>], (Locals['Stream_<label>'] as IRewriteRuleElementStream).NextNode());<\n>
>>

```

```

/** Gen ^($label ...) */
rewriteTokenLabelRefRoot(label,elementIndex) ::= <<
Root[<treeLevel>] := Adaptor.BecomeRoot((Locals['Stream_<label>'] as
IRewriteRuleElementStream).NextNode(), Root[<treeLevel>]) as I<ASTLabelType>;<\n>
>>

/** Gen ^($label ...) where label+=... */
rewriteTokenListLabelRefRoot ::= rewriteTokenLabelRefRoot

/** Gen ^(ID ...) or ^(ID[args] ...) */
rewriteTokenRefRoot(token,elementIndex,terminalOptions,args) ::= <<
Root[<treeLevel>] := Adaptor.BecomeRoot(<createRewriteNodeFromElement(...)>, Root[<treeLevel>]) as
I<ASTLabelType>;<\n>
>>

rewriteImaginaryTokenRef(args,token,terminalOptions,elementIndex) ::= <<
Adaptor.AddChild(Root[<treeLevel>], <createImaginaryNode(tokenType=token, ...)>;<\n>
>>

rewriteImaginaryTokenRefRoot(args,token,terminalOptions,elementIndex) ::= <<
Root[<treeLevel>] := Adaptor.BecomeRoot(<createImaginaryNode(tokenType=token, ...)>, Root[<treeLevel>]) as
I<ASTLabelType>;<\n>
>>

/** plain -> {foo} action */
rewriteAction(action) ::= <<
Root[0] = <action>;<\n>
>>

/** What is the name of the previous value of this rule's root tree? This
* let's us refer to $rule to mean previous value. I am reusing the
* variable 'tree' sitting in retval struct to hold the value of Root[0] right
* before I set it during rewrites. The assign will be to retval.Tree.
*/
prevRuleRootRef() ::= "RetVal"

rewriteRuleRef(rule) ::= <<
Adaptor.AddChild(Root[<treeLevel>], (Locals['Stream_<rule>'] as IRewriteRuleElementStream).NextTree());<\n>
>>

rewriteRuleRefRoot(rule) ::= <<
Root[<treeLevel>] := Adaptor.BecomeRoot((Locals['Stream_<rule>'] as IRewriteRuleElementStream).NextNode,
Root[<treeLevel>]) as I<ASTLabelType>;<\n>
>>

rewriteNodeAction(action) ::= <<
Adaptor.AddChild(Root[<treeLevel>], <action>;<\n>
>>

```

```

rewriteNodeActionRoot(action) ::= <<
Root[<treeLevel>] := Adaptor.BecomeRoot(<action>, Root[<treeLevel>]) as I<ASTLabelType>;<\n>
>>

/** Gen $ruleLabel ... where defined via ruleLabel=rule */
rewriteRuleLabelRef(label) ::= <<
Adaptor.AddChild(Root[<treeLevel>], (Locals['Stream_<label>'] as IRewriteRuleElementStream).NextTree());<\n>
>>

/** Gen $ruleLabel ... where defined via ruleLabel+=rule */
rewriteRuleListLabelRef(label) ::= <<
Adaptor.AddChild(Root[<treeLevel>], (Locals['Stream_<label>'] as IRewriteRuleElementStream).NextTree());<\n>
>>

/** Gen ^($ruleLabel ...) where ruleLabel=rule */
rewriteRuleLabelRefRoot(label) ::= <<
Root[<treeLevel>] := Adaptor.BecomeRoot((Locals['Stream_<label>'] as IRewriteRuleElementStream).NextNode,
Root[<treeLevel>]) as I<ASTLabelType>;<\n>
>>

/** Gen ^($ruleLabel ...) where ruleLabel+=rule */
rewriteRuleListLabelRefRoot(label) ::= <<
Root[<treeLevel>] := Adaptor.BecomeRoot((Locals['Stream_<label>'] as IRewriteRuleElementStream).NextNode,
Root[<treeLevel>]) as I<ASTLabelType>;<\n>
>>

createImaginaryNode(tokenType,terminalOptions,args) ::= <<
<if(terminalOptions.node)>
<! new MethodNode(IDLabel, args) !>
T<terminalOptions.node>.Create(<tokenType><if(args)>, <args; separator=", "><endif>)
<else>
Adaptor.CreateNode(<tokenType>, <args; separator=", "><if(!args)>'<tokenType>'<endif>) as I<ASTLabelType>
<endif>
>>

createRewriteNodeFromElement(token,terminalOptions,args) ::= <<
<if(terminalOptions.node)>
T<terminalOptions.node>.Create((Locals['Stream_<token>'] as IRewriteRuleElementStream).NextToken<if(args)>,
<args; separator=", "><endif>)
<else>
<if(args)> <! must create new node from old !>
Adaptor.Create(<token>, <args; separator=", ">)
<else>
(Locals['Stream_<token>'] as IRewriteRuleElementStream).NextNode
<endif>
<endif>
>>

```

Found in path(s):

```
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-  
jar/org/antlr/codegen/templates/Delphi/AST.stg
```

No license file was found, but licenses were detected in source scan.

```
/*
```

```
[The "BSD license"]
```

```
Copyright (c) 2005-2006 Terence Parr
```

```
All rights reserved.
```

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```
*/
```

```
/** Template overrides to add debugging to normal Java output;
```

```
* If ASTs are built, then you'll also get ASTDbg.stg loaded.
```

```
*/
```

```
@outputFile.imports() ::= <<
```

```
<@super.imports(>
```

```
import org.antlr.runtime.debug.*;
```

```
import java.io.IOException;
```

```
>>
```

```
@genericParser.members() ::= <<
```

```
<if(grammar.grammarIsRoot)>
```

```
public static final String[] ruleNames = new String[] {
```

```
"invalidRule", <grammar.allImportedRules: {rST | "<rST.name>"}; wrap="\n\t", separator=", ">
```

```
};<\n>
```

```
<endif>
```

```

public static final boolean[] decisionCanBacktrack = new boolean[] {
    false, // invalid decision
    <grammar.decisions:{d | <d.dfa.hasSynPred; null="false">}; wrap="\n  ", separator=" ">
};<\n>
<if(grammar.grammarIsRoot)> <! grammar imports other grammar(s) !>
public int ruleLevel = 0;
public int getRuleLevel() { return ruleLevel; }
public void incRuleLevel() { ruleLevel++; }
public void decRuleLevel() { ruleLevel--; }
<if(profile)>
    <ctorForProfilingRootGrammar()>
<else>
    <ctorForRootGrammar()>
<endif>
    <ctorForPredefinedListener()>
<else><! imported grammar !>
public int getRuleLevel() { return <grammar.delegators:{g| <g.delegateName()>}>.getRuleLevel(); }
public void incRuleLevel() { <grammar.delegators:{g| <g.delegateName()>}>.incRuleLevel(); }
public void decRuleLevel() { <grammar.delegators:{g| <g.delegateName()>}>.decRuleLevel(); }
    <ctorForDelegateGrammar()>
<endif>
<if(profile)>
public boolean alreadyParsedRule(IntStream input, int ruleIndex) {
    int stopIndex = getRuleMemoization(ruleIndex, input.index());
    ((Profiler)dbg).examineRuleMemoization(input, ruleIndex, stopIndex,
    <grammar.composite.rootGrammar.recognizerName>.ruleNames[ruleIndex]);
    return super.alreadyParsedRule(input, ruleIndex);
}

@Override
public void memoize(IntStream input,
    int ruleIndex,
    int ruleStartIndex)
{
    ((Profiler)dbg).memoize(input, ruleIndex, ruleStartIndex,
    <grammar.composite.rootGrammar.recognizerName>.ruleNames[ruleIndex]);
    super.memoize(input, ruleIndex, ruleStartIndex);
}<\n>
<endif>
protected boolean evalPredicate(boolean result, String predicate) {
    dbg.semanticPredicate(result, predicate);
    return result;
}<\n>
>>

ctorForRootGrammar() ::= <<
<! bug: can't use <@super.members()> cut-n-paste instead !>
<! Same except we add port number and profile stuff if root grammar !>

```

```

public <name>(<inputStreamType> input) {
    this(input, DebugEventSocketProxy.DEFAULT_DEBUGGER_PORT, new RecognizerSharedState());
}
public <name>(<inputStreamType> input, int port, RecognizerSharedState state) {
    super(input, state);
    <parserCtorBody()>
    <createListenerAndHandshake()>
    <grammar.directDelegates:{g|<g.delegateName()> = new <g.recognizerName>(input, dbg, this.state,
this<grammar.delegates:{g|,<g.delegateName()>}>);}; separator="\n">
    <@finally()>
}<\n>
>>

```

```

ctorForProfilingRootGrammar() ::= <<
<! bug: can't use <@super.members()> cut-n-paste instead !>
public <name>(<inputStreamType> input) {
    this(input, new Profiler(null), new RecognizerSharedState());
}
public <name>(<inputStreamType> input, DebugEventListener dbg, RecognizerSharedState state) {
    super(input, dbg, state);
    Profiler p = (Profiler)dbg;
    p.setParser(this);
    <parserCtorBody()>
    <grammar.directDelegates:
    {g|<g.delegateName()> = new <g.recognizerName>(input, dbg, this.state, this<grammar.delegates:{g|,
<g.delegateName()>}>);}; separator="\n">
    <@finally()>
}
<\n>
>>

```

```

/** Basically we don't want to set any dbg listeners are root will have it. */
ctorForDelegateGrammar() ::= <<
public <name>(<inputStreamType> input, DebugEventListener dbg, RecognizerSharedState
state<grammar.delegates:{g|,<g.recognizerName> <g.delegateName()>}>) {
    super(input, dbg, state);
    <parserCtorBody()>
    <grammar.directDelegates:
    {g|<g.delegateName()> = new <g.recognizerName>(input, this, this.state<grammar.delegates:{g|,
<g.delegateName()>}>);}; separator="\n">
}<\n>
>>

```

```

ctorForPredefinedListener() ::= <<
public <name>(<inputStreamType> input, DebugEventListener dbg) {
    <@superClassRef>super(input, dbg, new RecognizerSharedState());<@end>
    <if(profile)>
        Profiler p = (Profiler)dbg;

```

```

    p.setParser(this);
<endif>
<parserCtorBody()>
<grammar.directDelegates:{g|<g:delegateName()> = new <g.recognizerName>(input, dbg, this.state,
this<grammar.delegates:{g|, <g:delegateName()>}>);}; separator="\n">
<@finally()>
}<\n>
>>

```

```

createListenerAndHandshake() ::= <<
<if(TREE_PARSER)>
DebugEventSocketProxy proxy =
new DebugEventSocketProxy(this, port, input.getTreeAdaptor());<\n>
<else>
DebugEventSocketProxy proxy =
new DebugEventSocketProxy(this, port, null);<\n>
<endif>
setDebugListener(proxy);
try {
    proxy.handshake();
}
catch (IOException ioe) {
    reportError(ioe);
}
>>

```

```

@genericParser.superClassName() ::= "Debug<@super.superClassName()>"

```

```

@rule.preamble() ::= <<
try { dbg.enterRule(getGrammarFileName(), "<ruleName>");
if ( getRuleLevel()==0 ) {dbg.commence();}
incRuleLevel();
dbg.location(<ruleDescriptor.tree.line>, <ruleDescriptor.tree.charPositionInLine>);<\n>
>>

```

```

@rule.postamble() ::= <<
dbg.location(<ruleDescriptor.EORNode.line>, <ruleDescriptor.EORNode.charPositionInLine>);<\n>
}
finally {
    dbg.exitRule(getGrammarFileName(), "<ruleName>");
    decRuleLevel();
    if ( getRuleLevel()==0 ) {dbg.terminate();}
}<\n>
>>

```

```

@synpred.start() ::= "dbg.beginBacktrack(state.backtracking);"

```

```

@synpred.stop() ::= "dbg.endBacktrack(state.backtracking, success);"

```

```

// Common debug event triggers used by region overrides below

enterSubRule() ::=
"try { dbg.enterSubRule(<decisionNumber>);<\n>"

exitSubRule() ::=
"} finally {dbg.exitSubRule(<decisionNumber>);}<\n>"

enterDecision() ::=
"try { dbg.enterDecision(<decisionNumber>, decisionCanBacktrack[<decisionNumber>]);<\n>"

exitDecision() ::=
"} finally {dbg.exitDecision(<decisionNumber>);}<\n>"

enterAlt(n) ::= "dbg.enterAlt(<n>);<\n>"

// Region overrides that tell various constructs to add debugging triggers

@block.predecision() ::= "<enterSubRule()><enterDecision()>"

@block.postdecision() ::= "<exitDecision()>"

@block.postbranch() ::= "<exitSubRule()>"

@ruleBlock.predecision() ::= "<enterDecision()>"

@ruleBlock.postdecision() ::= "<exitDecision()>"

@ruleBlockSingleAlt.preal() ::= "<enterAlt(n=\"1\")>"

@blockSingleAlt.preal() ::= "<enterAlt(n=\"1\")>"

@positiveClosureBlock.preloop() ::= "<enterSubRule()>"

@positiveClosureBlock.postloop() ::= "<exitSubRule()>"

@positiveClosureBlock.predecision() ::= "<enterDecision()>"

@positiveClosureBlock.postdecision() ::= "<exitDecision()>"

@positiveClosureBlock.earlyExitException() ::=
"dbg.recognitionException(eee);<\n>"

@closureBlock.preloop() ::= "<enterSubRule()>"

@closureBlock.postloop() ::= "<exitSubRule()>"

```

```

@closureBlock.predecision() ::= "<enterDecision()"

@closureBlock.postdecision() ::= "<exitDecision()"

@altSwitchCase.preal() ::= "<enterAlt(altNum)" // altNum is arg of altSwitchCase

@element.prematch() ::=
"dbg.location(<e.line>,<e.pos>);" // e is arg of element

@matchSet.mismatchedSetException() ::=
"dbg.recognitionException(mse);"

@dfaState.noViableAltException() ::= "dbg.recognitionException(nvae);"

@dfaStateSwitch.noViableAltException() ::= "dbg.recognitionException(nvae);"

dfaDecision(decisionNumber,description) ::= <<
try {
  isCyclicDecision = true;
  <super.dfaDecision(...)>
}
catch (NoViableAltException nvae) {
  dbg.recognitionException(nvae);
  throw nvae;
}
>>

@cyclicDFA.errorMethod() ::= <<
public void error(NoViableAltException nvae) {
  dbg.recognitionException(nvae);
}
>>

/** Force predicate validation to trigger an event */
evalPredicate(pred,description) ::= <<
evalPredicate(<pred>,<description>")
>>

```

Found in path(s):

```

* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/Java/Dbg.stg
No license file was found, but licenses were detected in source scan.

```

```

/*

```

```

[The "BSD license"]

```

```

Copyright (c) 2008 Erik van Bilzen

```

```

Copyright (c) 2007-2008 Johannes Luber

```

```

Copyright (c) 2005-2007 Kunle Odutola

```

```

Copyright (c) 2005-2006 Terence Parr

```

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

/** Templates for building ASTs during normal parsing.

*

* Deal with many combinations. Dimensions are:

* Auto build or rewrite

* no label, label, list label (label/no-label handled together)

* child, root

* token, set, rule, wildcard

*

* The situation is not too bad as rewrite (->) usage makes ^ and !

* invalid. There is no huge explosion of combinations.

*/

group ASTParser;

@rule.setErrorReturnValue() ::= <<

RetVal.Tree := Adaptor.ErrorNode(Input, RetVal.Start as IToken,
Input.LT(-1), RE) as I<ASTLabelType>;

>>

// TOKEN AST STUFF

/** ID and output=AST */

tokenRef(token,label,elementIndex,terminalOptions) ::= <<

<super.tokenRef(...)>

```

<if(backtracking)>
if (State.Backtracking = 0) then
begin<\n>
<endif>
<label>_tree := <createNodeFromToken(...)>;
Adaptor.AddChild(Root[0], <label>_tree);
<if(backtracking)>
end;
<endif>
>>

/** ID! and output=AST (same as plain tokenRef) */
tokenRefBang(token,label,elementIndex) ::= "<super.tokenRef(...)>"

/** ID^ and output=AST */
tokenRefRuleRoot(token,label,elementIndex,terminalOptions) ::= <<
<super.tokenRef(...)>
<if(backtracking)>
if (State.Backtracking = 0) then
begin
<endif>
<label>_tree := <createNodeFromToken(...)>;
Root[0] := Adaptor.BecomeRoot(<label>_tree, Root[0]) as I<ASTLabelType>;
<if(backtracking)>
end;
<endif>
>>

/** ids+=ID! and output=AST */
tokenRefBangAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<tokenRefBang(...)>
<listLabel(elem=label,...)>
>>

/** label+=TOKEN when output=AST but not rewrite alt */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

/** Match label+=TOKEN^ when output=AST but not rewrite alt */
tokenRefRuleRootAndListLabel(token,label,terminalOptions,elementIndex) ::= <<
<tokenRefRuleRoot(...)>
<listLabel(elem=label,...)>
>>

// SET AST

```

```

// the match set stuff is interesting in that it uses an argument list
// to pass code to the default matchSet; another possible way to alter
// inherited code. I don't use the region stuff because I need to pass
// different chunks depending on the operator. I don't like making
// the template name have the operator as the number of templates gets
// large but this is the most flexible--this is as opposed to having
// the code generator call matchSet then add root code or ruleroot code
// plus list label plus ... The combinations might require complicated
// rather than just added on code. Investigate that refactoring when
// I have more time.

matchSet(s,label,terminalOptions,elementIndex,postmatchCode) ::= <<
<super.matchSet(..., postmatchCode={<if(backtracking)>if (State.Backtracking = 0) then
<endif>Adaptor.AddChild(Root[0], <createNodeFromToken(...)>);}>
>>

matchRuleBlockSet(s,label,terminalOptions,elementIndex,postmatchCode,treeLevel="0") ::= <<
<matchSet(...)>
>>

matchSetBang(s,label,elementIndex,postmatchCode) ::= "<super.matchSet(...)>"

// note there is no matchSetTrack because -> rewrites force sets to be
// plain old blocks of alts: (A|B|...|C)

matchSetRuleRoot(s,label,terminalOptions,elementIndex,debug) ::= <<
<if(label)>
<label> := Input.LT(1) as I<labelType>;<\n>
<endif>
<super.matchSet(..., postmatchCode={<if(backtracking)>if (State.Backtracking = 0) then <endif>Root[0] :=
Adaptor.BecomeRoot(<createNodeFromToken(...)>, Root[0] as I<ASTLabelType>);}>
>>

// RULE REF AST

/** rule when output=AST */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<if(backtracking)>if (State.Backtracking = 0) then <endif>Adaptor.AddChild(Root[0], <label>.Tree);
>>

/** rule! is same as normal rule ref */
ruleRefBang(rule,label,elementIndex,args,scope) ::= "<super.ruleRef(...)>"

/** rule^ */
ruleRefRuleRoot(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<if(backtracking)>if (State.Backtracking = 0) then <endif>Root[0] := Adaptor.BecomeRoot(<label>.Tree, Root[0])

```

```

as I<ASTLabelType>;
>>

/** x+=rule when output=AST */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(elem=label+".Tree",...)>
>>

/** x+=rule! when output=AST is a rule ref with list addition */
ruleRefBangAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefBang(...)>
<listLabel(elem=label+".Tree",...)>
>>

/** x+=rule^ */
ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<listLabel(elem=label+".Tree",...)>
>>

// WILDCARD AST

wildcard(label,elementIndex) ::= <<
<super.wildcard(...)>
<if(backtracking)>
if (State.Backtracking = 0) then
begin
<endif>
<label>_tree := Adaptor.CreateNode(<label>) as I<ASTLabelType>;
Adaptor.AddChild(Root[0], <label>_tree);
<if(backtracking)>
end;
<endif>
>>

wildcardBang(label,elementIndex) ::= "<super.wildcard(...)>"

wildcardRuleRoot(label,elementIndex) ::= <<
<super.wildcard(...)>
<if(backtracking)>
if (State.Backtracking = 0) then
begin
<endif>
<label>_tree := Adaptor.CreateNode(<label>) as I<ASTLabelType>;
Root[0] := Adaptor.BecomeRoot(<label>_tree, Root[0]) as I<ASTLabelType>;
<if(backtracking)>
end;

```

```

<endif>
>>

createNodeFromToken(label,terminalOptions) ::= <<
<if(terminalOptions.node)>
T<terminalOptions.node>.Create(<label>) <! new MethodNode(IDLabel) !>
<else>
Adaptor.CreateNode(<label>) as I<ASTLabelType>
<endif>
>>

ruleCleanUp() ::= <<
<super.ruleCleanUp()>
<if(backtracking)>
if (State.Backtracking = 0) then
begin<\n>
<endif>
RetVal.Tree := Adaptor.RulePostProcessing(Root[0]) as I<ASTLabelType>;
<if(!TREE_PARSER)>
Adaptor.SetTokenBoundaries(RetVal.Tree, RetVal.Start as IToken, RetVal.Stop as IToken);
<endif>
<if(backtracking)>
<\n>end;
<endif>
>>

```

Found in path(s):

```
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-
jar/org/antlr/codegen/templates/Delphi/ASTParser.stg
```

No license file was found, but licenses were detected in source scan.

```
/*
```

[The "BSD license"]

Copyright (c) 2005-2009 Jim Idle, Temporal Wave LLC

<http://www.temporal-wave.com>

<http://www.linkedin.com/in/jimidle>

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products

derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

/** Template overrides to add debugging to normal C output;

* If ASTs are built, then you'll also get ASTDbg.stg loaded.

*/

```
@genericParser.members() ::= <<
```

```
<if(grammar.grammarIsRoot)>
```

```
const char *
```

```
ruleNames[] =
```

```
{
```

```
"invalidRule", <grammar.allImportedRules:{rST | "<rST.name>"}; wrap="\n ", separator=", ">
```

```
};<\n>
```

```
<endif>
```

```
<if(grammar.grammarIsRoot)> <! grammar imports other grammar(s) !>
```

```
static ANTLR3_UINT32 ruleLevel = 0;
```

```
static ANTLR3_UINT32 getRuleLevel()
```

```
{
```

```
return ruleLevel;
```

```
}
```

```
static void incRuleLevel()
```

```
{
```

```
ruleLevel++;
```

```
}
```

```
static void decRuleLevel()
```

```
{
```

```
ruleLevel--;
```

```
}
```

```
<else> <! imported grammar !>
```

```
static ANTLR3_UINT32
```

```
getRuleLevel()
```

```
{
```

```
return <grammar.delegators:{g| <g:delegateName()>}>->getRuleLevel();
```

```
}
```

```
static void incRuleLevel()
```

```
{
```

```
<grammar.delegators:{g| <g:delegateName()>}>->incRuleLevel();
```

```
}
```

```

static void
decRuleLevel()
{
    <grammar.delegators:{ g| <g.delegateName()>}>.decRuleLevel();
}
<endif>
<if(profile)>
// Profiling not yet implemented for C target
//
<endif>
<if(grammar.grammarIsRoot)>
<ctorForPredefinedListener()>
<else>
<ctorForDelegateGrammar()>
<endif>

static ANTLR3_BOOLEAN
evalPredicate(p<name> ctx, ANTLR3_BOOLEAN result, const char * predicate)
{
    DBG->semanticPredicate(DBG, result, predicate);
    return result;
}<\n>
>>

@genericParser.debugStuff() ::= <<
<if(grammar.grammarIsRoot)>
<createListenerAndHandshake()>
<endif>
>>

ctorForProfilingRootGrammar() ::= <<
>>

/** Basically we don't want to set any dbg listeners as root will have it. */
ctorForDelegateGrammar() ::= <<
>>

ctorForPredefinedListener() ::= <<
>>

createListenerAndHandshake() ::= <<
{
    // DEBUG MODE code
    //
    pANTLR3_DEBUG_EVENT_LISTENER proxy;
    proxy = antlr3DebugListenerNew();

```

```

proxy->grammarFileName = INPUT->tokenSource->strFactory->newStr8(INPUT->tokenSource->strFactory,
(pANTLR3_UINT8)ctx->getGrammarFileName());

<if(TREE_PARSER)>
proxy->adaptor = ADAPTOR;
<endif>
PARSER->setDebugListener(PARSER, proxy);

// Try to connect to the debugger (waits forever for a connection)
//
proxy->handshake(proxy);

// End DEBUG MODE code
//
}
>>

@rule.preamble() ::= <<
if ( getRuleLevel()==0 )
{
DBG->commence(DBG);
}
DBG->enterRule(DBG, getGrammarFileName(), (const char *)<ruleName>");
incRuleLevel();
DBG->location(DBG, <ruleDescriptor.tree.line>, <ruleDescriptor.tree.column>);<\n>
>>

@rule.postamble() ::= <<
DBG->location(DBG, <ruleDescriptor.EORNode.line>, <ruleDescriptor.EORNode.column>);<\n>
DBG->exitRule(DBG, getGrammarFileName(), (const char *)<ruleName>");
decRuleLevel();
if ( getRuleLevel()==0 )
{
DBG->terminate(DBG);
}
<\n>
>>

@checkRuleBacktrackFailure.debugClean() ::= <<
DBG->exitRule(DBG, getGrammarFileName(), (const char *)<ruleName>");
decRuleLevel();
>>

@synpred.start() ::= "DBG->beginBacktrack(DBG, BACKTRACKING);"

@synpred.stop() ::= "DBG->endBacktrack(DBG, BACKTRACKING, success);"

```

```

// Common debug event triggers used by region overrides below

enterSubRule() ::=
  "DBG->enterSubRule(DBG, <decisionNumber>);<\n>"

exitSubRule() ::=
  "DBG->exitSubRule(DBG, <decisionNumber>);<\n>"

enterDecision() ::=
  "DBG->enterDecision(DBG, <decisionNumber>);<\n>"

exitDecision() ::=
  "DBG->exitDecision(DBG, <decisionNumber>);<\n>"

enterAlt(n) ::= "DBG->enterAlt(DBG, <n>);<\n>"

// Region overrides that tell various constructs to add debugging triggers

@block.predecision() ::= "<enterSubRule()><enterDecision()>"

@block.postdecision() ::= "<exitDecision()>"

@block.postbranch() ::= "<exitSubRule()>"

@ruleBlock.predecision() ::= "<enterDecision()>"

@ruleBlock.postdecision() ::= "<exitDecision()>"

@ruleBlockSingleAlt.prealt() ::= "<enterAlt(n=\"1\")>"

@blockSingleAlt.prealt() ::= "<enterAlt(n=\"1\")>"

@positiveClosureBlock.preloop() ::= "<enterSubRule()>"

@positiveClosureBlock.postloop() ::= "<exitSubRule()>"

@positiveClosureBlock.predecision() ::= "<enterDecision()>"

@positiveClosureBlock.postdecision() ::= "<exitDecision()>"

@positiveClosureBlock.earlyExitException() ::=
  "DBG->recognitionException(DBG, EXCEPTION);<\n>"

@closureBlock.preloop() ::= "<enterSubRule()>"

@closureBlock.postloop() ::= "<exitSubRule()>"

@closureBlock.predecision() ::= "<enterDecision()>"

```

```

@closureBlock.postdecision() ::= "<exitDecision()>"

@altSwitchCase.preal() ::= "<enterAlt(altNum)>"

@element.prematch() ::=
  "DBG->location(DBG, <e.line>, <e.pos>);" // e is arg of element

@matchSet.mismatchedSetException() ::=
  "DBG->recognitionException(DBG, EXCEPTION);"

@newNVException.noViableAltException() ::= "DBG->recognitionException(DBG, EXCEPTION);"

dfaDecision(decisionNumber,description) ::= <<
alt<decisionNumber> = cdfa<decisionNumber>.predict(ctx, RECOGNIZER, ISTREAM,
&cdfa<decisionNumber>);
if (HASEXCEPTION())
{
  DBG->recognitionException(DBG, EXCEPTION);
  goto rule<ruleDescriptor.name>Ex;
}
<checkRuleBacktrackFailure()>
>>

@cyclicDFA.errorMethod() ::= <<
//static void
//dfaError(p<name> ctx)
//{
//  DBG->recognitionException(DBG, EXCEPTION);
//}
>>

/** Force predicate validation to trigger an event */
evalPredicate(pred,description) ::= <<
evalPredicate(ctx, <pred>, (const char *)"<description>")
>>

```

Found in path(s):

```

* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/C/Dbg.stg
No license file was found, but licenses were detected in source scan.

```

```

/*
[The "BSD license"]
Copyright (c) 2007 Kay Roepke 2010 Alan Condit
All rights reserved.

```

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions

are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

/** Templates for building ASTs during tree parsing.

*

* Deal with many combinations. Dimensions are:

* Auto build or rewrite

* no label, label, list label (label/no-label handled together)

* child, root

* token, set, rule, wildcard

*

* Each combination has its own template except that label/no label

* is combined into tokenRef, ruleRef, ...

*/

/* addition memVars for returnscopes */

@returnScopeInterface.memVars() ::= <<

/* ASTTreeParser returnScopeInterface.memVars */

<recognizer.ASTLabelType; null="CommonTree"> *tree;

>>

/** the interface of returnScope methodsDecl */

@returnScopeInterface.methodsDecl() ::= <<

/* ASTTreeParser returnScopeInterface.methodsDecl */

- (<recognizer.ASTLabelType; null="CommonTree"> *)getTree;

- (void) setTree:(<recognizer.ASTLabelType; null="CommonTree"> *)aTree;<\n>

>>

/** the implementation of returnScope methods */

@returnScope.methods() ::= <<

```

/* ASTTreeParser returnScope.methods */
- (<ASTLabelType> *)getTree
{
    return tree;
}

- (void) setTree:(<ASTLabelType> *)aTree
{
    if (tree != aTree) {
        if (tree != nil) [tree release];
        if (aTree != nil) [aTree retain];
        tree = aTree;
    }
}

- (void) dealloc
{
    [self setTree:nil];
    [super dealloc];
}

@synthesize tree;
>>

@returnScopeProperties() ::= <<
@property (retain) <recognizer.ASTLabelType; null="CommonTree"> *tree;
>>

/** Add a variable to track last element matched */
ruleDeclarations() ::= <<
/* ASTTreeParser ruleDeclarations */
<super.ruleDeclarations()>
<ASTLabelType> *_first_0 = nil;
<ASTLabelType> *_last = nil;<\n>
>>

/** What to emit when there is no rewrite rule. For auto build
 * mode, does nothing.
 */
noRewrite(rewriteBlockLevel, treeLevel) ::= <<
/* ASTTreeParser noRewrite */
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<if(rewriteMode)>
retval.tree = (<ASTLabelType> *)_first_0;
if ( [treeAdaptor getParent:retval.tree] != nil && [treeAdaptor isNil:[treeAdaptor getParent:retval.tree]] ) )
    retval.tree = (<ASTLabelType> *)[treeAdaptor getParent:retval.tree];
<endif>
<if(backtracking)>}<endif>

```

```

>>

/** match ^(root children) in tree parser; override here to
 * add tree construction actions.
 */
tree(root, actionsAfterRoot, children, nullableChildList,
     enclosingTreeLevel, treeLevel) ::= <<
/* ASTTreeParser tree */
_last = (<ASTLabelType> *)[input LT:1];
{
<ASTLabelType> *_save_last_<treeLevel> = _last;
<ASTLabelType> *_first_<treeLevel> = nil;
<if(!rewriteMode)>
<ASTLabelType> *root_<treeLevel> = [[[treeAdaptor class] newEmptyTree] retain];
<endif>
<root:element()>
<if(rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> )<endif>
<if(root.el.rule)>
if ( _first_<enclosingTreeLevel>==nil ) _first_<enclosingTreeLevel> = <root.el.label>.tree;
<else>
if ( _first_<enclosingTreeLevel>==nil ) _first_<enclosingTreeLevel> = <root.el.label>;
<endif>
<endif>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if ( [input LA:1] == TokenTypeDOWN ) {
[self match:input TokenType:TokenTypeDOWN Follow:nil]; <checkRuleBacktrackFailure()>
<children:element()>
[self match:input TokenType:TokenTypeUP Follow:nil]; <checkRuleBacktrackFailure()>
}
<else>
[self match:input TokenType:TokenTypeDOWN Follow:nil]; <checkRuleBacktrackFailure()>
<children:element()>
[self match:input TokenType:TokenTypeUP Follow:nil]; <checkRuleBacktrackFailure()>
<endif>
<if(!rewriteMode)>
[treeAdaptor addChild:root_<treeLevel> toTree:root_<enclosingTreeLevel>];
<endif>
_last = _save_last_<treeLevel>;
}<\n>
>>

// TOKEN AST STUFF

/** ID! and output=AST (same as plain tokenRef) 'cept add
 * setting of _last
 */

```

```

tokenRefBang(token,label,elementIndex,terminalOptions) ::= <<
/* ASTTreeParser tokenRefBang */
_last = (<ASTLabelType> *)[input LT:1];
<super.tokenRef(...)>
>>

/** ID auto construct */
tokenRef(token,label,elementIndex,terminalOptions) ::= <<
/* ASTTreeParser tokenRef */
_last = (<ASTLabelType> *)[input LT:1];
<super.tokenRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<if(terminalOptions.node)>
  <label>_tree = [<terminalOptions.node> new<terminalOptions.node>:<label>];
<else>
  <label>_tree = (<ASTLabelType> *)[treeAdaptor dupNode:<label>];
<endif><\n>
  [treeAdaptor addChild:<label>_tree toTree:root_<treeLevel>];
<if(backtracking)>}<endif>
<else><! rewrite mode !>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> )<endif>
if ( _first_<treeLevel>==nil ) _first_<treeLevel> = <label>;
<endif>
>>

/** label+=TOKEN auto construct */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
/* ASTTreeParser tokenRefAndListLabel */
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

/** ^(ID ...) auto construct */
tokenRefRuleRoot(token,label,elementIndex) ::= <<
/* ASTTreeParser tokenRefRuleRoot */
_last = (<ASTLabelType> *)[input LT:1];
<super.tokenRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<if(terminalOptions.node)>
<label>_tree = [<terminalOptions.node> new<terminalOptions.node>:<label>];
<else>
<label>_tree = (<ASTLabelType> *)[treeAdaptor dupNode:<label>];
<endif><\n>
root_<treeLevel> = (<ASTLabelType> *)[treeAdaptor becomeRoot:<label>_tree old:root_<treeLevel>];
<if(backtracking)>}<endif>
<endif>

```

```

>>

/** Match ^(label+=TOKEN ...) auto construct */
tokenRefRuleRootAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
/* ASTTreeParser tokenRefRuleRootAndListLabel */
<tokenRefRuleRoot(...)>
<listLabel(elem=label,...)>
>>

/** Match . wildcard and auto dup the node/subtree */
wildcard(token,label,elementIndex,terminalOptions) ::= <<
/* ASTTreeParser wildcard */
_last = (<ASTLabelType> *)[input LT:1];
<super.wildcard(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<label>_tree = (<ASTLabelType> *)[adaptor dupTree:<label>];
[adaptor addChild:<label>_tree toTree:root_<treeLevel>];
<if(backtracking)>}<endif>
<else> <! rewrite mode !>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> )<endif>
if ( _first_<treeLevel> == nil ) _first_<treeLevel> = <label>;
<endif>
>>

// SET AST

matchSet(s,label,terminalOptions,elementIndex,postmatchCode) ::= <<
/* ASTTreeParser matchSet */
_last = (<ASTLabelType> *)[input LT:1];
<super.matchSet(postmatchCode={
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<if(terminalOptions.node)>
<label>_tree = [<terminalOptions.node> new<terminalOptions.node>:<label>];
<else>
<label>_tree = (<ASTLabelType> *)[adaptor dupNode:<label>];
<endif><\n>
[adaptor addChild:<label>_tree toTree:root_<treeLevel>];
<if(backtracking)>}<endif>
<endif>
}, ...
)>
>>

matchRuleBlockSet(s,label,terminalOptions,elementIndex,postmatchCode,treeLevel="0") ::= <<
/* ASTTreeParser matchRuleBlockSet */
<matchSet(...)>

```

```

<noRewrite(...)> <! set return tree !>
>>

matchSetBang(s,label,terminalOptions,elementIndex,postmatchCode) ::= <<
/* ASTTreeParser matchSetBang */
_last = (<ASTLabelType> *)[input LT:1];
<super.matchSet(...)>
>>

matchSetRuleRoot(s,label,terminalOptions,elementIndex,debug) ::= <<
/* ASTTreeParser matchSetRuleRoot */
<super.matchSet(postmatchCode={
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<if(terminalOptions.node)>
<label>_tree = [<terminalOptions.node> new<terminalOptions.node>:<label>];
<else>
<label>_tree = (<ASTLabelType> *)[adaptor dupNode:<label>];
<endif><\n>
root_<treeLevel> = (<ASTLabelType> *)[adaptor becomeRoot:<label>_tree old:root_<treeLevel>];
<if(backtracking)>}\}<endif>
<endif>
}, ...
)>
>>

// RULE REF AST

/** rule auto construct */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
/* ASTTreeParser ruleRef */
_last = (<ASTLabelType> *)[input LT:1];
<super.ruleRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>
<if(!rewriteMode)>
  [treeAdaptor addChild:<label>.tree toTree:root_<treeLevel>];
<else> <! rewrite mode !>
if ( _first_<treeLevel> == nil ) _first_<treeLevel> = <label>.tree;
<endif>
>>

/** x+=rule auto construct */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
/* ASTTreeParser ruleRefAndListLabel */
<ruleRef(...)>
<! <listLabel(elem = "["+label+" getTree]",...)> !>
<listLabel(elem = {[<label> getTree]},...)>
>>

```

```

/** ^(rule ...) auto construct */
ruleRefRuleRoot(rule,label,elementIndex,args,scope) ::= <<
/* ASTTreeParser ruleRefRuleRoot */
_last = (<ASTLabelType> *)[input LT:1];
<super.ruleRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( state.backtracking == 0 ) <endif>
root_<treeLevel> = (<ASTLabelType> *)[treeAdaptor becomeRoot:<label>.tree old:root_<treeLevel>];
<endif>
>>

/** ^(x+=rule ...) auto construct */
ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope) ::= <<
/* ASTTreeParser ruleRefRuleRootAndListLabel */
<ruleRefRuleRoot(...)>
<listLabel(elem = {[<label> getTree]},...)>
>>

/** rule when output=AST and tracking for rewrite */
ruleRefTrack(rule,label,elementIndex,args,scope) ::= <<
/* ASTTreeParser ruleRefTrack */
_last = (<ASTLabelType> *)[input LT:1];
<super.ruleRefTrack(...)>
>>

/** x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
/* ASTTreeParser ruleRefTrackAndListLabel */
_last = (<ASTLabelType> *)[input LT:1];
<super.ruleRefTrackAndListLabel(...)>
>>

/** ^(rule ...) rewrite */
ruleRefRuleRootTrack(rule,label,elementIndex,args,scope) ::= <<
/* ASTTreeParser ruleRefRuleRootTrack */
_last = (<ASTLabelType> *)[input LT:1];
<super.ruleRefRuleRootTrack(...)>
>>

/** ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
/* ASTTreeParser ruleRefRuleRootTrackAndListLabel */
_last = (<ASTLabelType> *)[input LT:1];
<super.ruleRefRuleRootTrackAndListLabel(...)>
>>

/** Streams for token refs are tree nodes now; override to

```

```

* change nextToken to nextNode.
*/
createRewriteNodeFromElement(token,terminalOptions,scope) ::= <<
/* ASTTreeParser createRewriteNodeFromElement */
<if(terminalOptions.node)>
<! new <terminalOptions.node>(stream_<token>.nextNode()) !>
[[[<terminalOptions.node>(stream_<token> alloc) init] nextNode];
<else>
<! stream_<token>.nextNode() !>
[stream_<token> nextNode]
<endif>
>>

ruleCleanUp() ::= <<
/* ASTTreeParser ruleCleanUp */
<super.ruleCleanUp()>
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<\n<endif>
retval.tree = (<ASTLabelType> *)[treeAdaptor rulePostProcessing:root_0];
<if(backtracking)>}<endif>
<endif>
>>

```

Found in path(s):

```

* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-
jar/org/antlr/codegen/templates/ObjC/ASTTreeParser.stg

```

No license file was found, but licenses were detected in source scan.

```

/*
* [The "BSD license"]
* Copyright (c) 2007-2008 Johannes Luber
* Copyright (c) 2005-2007 Kunle Odutola
* Copyright (c) 2005 Terence Parr
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products
* derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES

```

```

* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

```

```

/** Template overrides to add debugging to AST stuff. Dynamic inheritance
* hierarchy is set up as ASTDbg : AST : Dbg : Java by code generator.
*/

```

```

parserMembers() ::= <<
protected DebugTreeAdaptor adaptor;

public ITreeAdaptor TreeAdaptor
{
get
{
return adaptor;
}
set
{
<if(grammar.grammarIsRoot)>
this.adaptor = new DebugTreeAdaptor(dbg,adaptor);
<else>
this.adaptor = (DebugTreeAdaptor)adaptor; // delegator sends dbg adaptor
<endif><\n>
<grammar.directDelegates:{g|<g.delegateName()>.TreeAdaptor = this.adaptor;}>
}
}<\n>
>>

```

```

parserCtorBody() ::= <<
<super.parserCtorBody()>
>>

```

```

createListenerAndHandshake() ::= <<
DebugEventSocketProxy proxy = new DebugEventSocketProxy( this, port,
<if(TREE_PARSER)>input.TreeAdaptor<else>adaptor<endif> );
DebugListener = proxy;
<inputStreamType> = new Debug<inputStreamType>( input, proxy );
try
{
proxy.Handshake();
}

```

```
catch ( IOException ioe )
{
  ReportError( ioe );
}
>>
```

```
@ctorForRootGrammar.finally() ::= <<
ITreeAdaptor adap = new CommonTreeAdaptor();
TreeAdaptor = adap;
proxy.TreeAdaptor = adap;
>>
```

```
@ctorForProfilingRootGrammar.finally() ::= <<
ITreeAdaptor adap = new CommonTreeAdaptor();
TreeAdaptor = adap;
>>
```

```
@ctorForPredefinedListener.superClassRef() ::= ": base( input, dbg )"

```

```
@ctorForPredefinedListener.finally() ::= <<
<if(grammar.grammarIsRoot)><! don't create new adaptor for delegates !>
ITreeAdaptor adap = new CommonTreeAdaptor();
TreeAdaptor = adap;<\n>
<endif>
>>
```

```
//@rewriteElement.pregen() ::= "dbg.Location( <e.line>, <e.pos> );"
```

Found in path(s):

* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/CSharp2/ASTDbg.stg

No license file was found, but licenses were detected in source scan.

/*

* [The "BSD license"]

* Copyright (c) 2005-2008 Terence Parr

* All rights reserved.

*

* Conversion to C#:

* Copyright (c) 2008-2010 Sam Harwell, Pixel Mine, Inc.

* All rights reserved.

*

* Redistribution and use in source and binary forms, with or without

* modification, are permitted provided that the following conditions

* are met:

* 1. Redistributions of source code must retain the above copyright

* notice, this list of conditions and the following disclaimer.

* 2. Redistributions in binary form must reproduce the above copyright

* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products
* derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):

* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/codegen/CSharp3Target.java

No license file was found, but licenses were detected in source scan.

/*

[The "BSD license"]

Copyright (c) 2005-2009 Gokulakannan Somasundaram,

All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```

*/

/*
* This code generating template and the associated Cpp runtime was produced by:
* Gokulakannan Somasundaram ( heavy lifting from C Run-time by Jim Idle )
*/
cTypeInitMap ::= [
    "int"   : "0",      // Integers  start out being 0
    "long"  : "0",      // Longs    start out being 0
    "float" : "0.0",    // Floats   start out being 0
    "double": "0.0",    // Doubles  start out being 0
    "bool"  : "false", // Booleans start out being Antlr C for false
    "byte"  : "0",      // Bytes    start out being 0
    "short" : "0",      // Shorts   start out being 0
    "char"  : "0"       // Chars    start out being 0
]

leadIn(type) ::=
<<
/** \file
* This <type> file was generated by $ANTLR version <ANTLRVersion>
*
* - From the grammar source file : <fileName>
* - On : <generatedTimestamp>
<if(LEXER)>
* - for the lexer : <name>Lexer
<endif>
<if(PARSER)>
* - for the parser : <name>Parser
<endif>
<if(TREE_PARSER)>
* - for the tree parser : <name>TreeParser
<endif>
*
* Editing it, at least manually, is not wise.
*
* C++ language generator and runtime by Gokulakannan Somasundaram ( heavy lifting from C Run-time by Jim
Idle )
*
*
>>

/** The overall file structure of a recognizer; stores methods for rules
* and cyclic DFAs plus support code.
*/
outputFile( LEXER,
            PARSER,
            TREE_PARSER,

```

```

    actionScope,
    actions,
    docComment,
    recognizer,
    name,
    tokens,
    tokenNames,
    rules,
    cyclicDFAs,
    bitsets,
    buildTemplate,
    buildAST,
    rewriteMode,
    profile,
    backtracking,
    synpreds,
    memoize,
    numRules,
    fileName,
    ANTLRVersion,
    generatedTimestamp,
    trace,
    scopes,
    superClass,
    literals
) ::=

<<
<leadIn("C++ source")>
*/
// [The "BSD license"]
// Copyright (c) 2005-2009 Gokulakannan Somasundaram, ElectronDB
//
// All rights reserved.
//
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions
// are met:
// 1. Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// 2. Redistributions in binary form must reproduce the above copyright
// notice, this list of conditions and the following disclaimer in the
// documentation and/or other materials provided with the distribution.
// 3. The name of the author may not be used to endorse or promote products
// derived from this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
// IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
// OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.

```

```
// IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
// INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
// NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
// THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

```
<if(actions.(actionScope).header)>
```

```
/* =====
```

```
* This is what the grammar programmer asked us to put at the top of every file.
```

```
*/
```

```
<actions.(actionScope).header>
```

```
/* End of Header action.
```

```
* =====
```

```
*/
```

```
<endif>
```

```
/* -----
```

```
* Include the ANTLR3 generated header file.
```

```
*/
```

```
#include "<name>.hpp"
```

```
<if(trace)>
```

```
#include <iostream>
```

```
<endif>
```

```
<if(recognizer.grammar.delegators)>
```

```
// Include delegator definition header files
```

```
//
```

```
<recognizer.grammar.delegators: {g#include "<g.recognizerName>.hpp" }; separator="\n">
```

```
<endif>
```

```
<actions.(actionScope).postinclude>
```

```
/* ----- */
```

```
<docComment>
```

```
<if(literals)>
```

```
<beginNamespace(actions)>
```

```
/** String literals used by <name> that we must do things like MATCHS() with.
```

```
* C will normally just lay down 8 bit characters, and you can use L"xxx" to
```

```
* get wchar_t, but wchar_t is 16 bits on Windows, which is not UTF32 and so
```

```
* we perform this little trick of defining the literals as arrays of UINT32
```

```
* and passing in the address of these.
```

```
*/
```

```
<literals:{it | static ANTLR_UCHAR lit_<i>[] = <it>;}; separator="\n">
```

```

<endNamespace(actions)>

<endif>

/* ===== */

/* =====
* Start of recognizer
*/

<recognizer>

/* End of code
* =====
*/

>>
headerFileExtension() ::= ".hpp"

beginNamespace(actions) ::= <%
<if(actions.actionScope.namespace)>
namespace <actions.actionScope.namespace> {
<endif>
%>

endNamespace(actions) ::= <%
<if(actions.actionScope.namespace)>
}
<endif>
%>

headerFile( LEXER,
            PARSER,
            TREE_PARSER,
            actionScope,
            actions,
            docComment,
            recognizer,
            name,
            tokens,
            tokenNames,
            rules,
            cyclicDFAs,
            bitsets,
            buildTemplate,
            buildAST,

```

```

        rewriteMode,
        profile,
        backtracking,
        synpreds,
        memoize,
        numRules,
        fileName,
        ANTLRVersion,
        generatedTimestamp,
        trace,
        scopes,
        superClass,
        literals
    ) ::=
<<
<leadIn("C++ header")>
<if(PARSER)>
* The parser <mainName()> has the callable functions (rules) shown below,
<endif>
<if(LEXER)>
* The lexer <mainName()> has the callable functions (rules) shown below,
<endif>
<if(TREE_PARSER)>
* The tree parser <mainName()> has the callable functions (rules) shown below,
<endif>
* which will invoke the code for the associated rule in the source grammar
* assuming that the input stream is pointing to a token/text stream that could begin
* this rule.
*
* For instance if you call the first (topmost) rule in a parser grammar, you will
* get the results of a full parse, but calling a rule half way through the grammar will
* allow you to pass part of a full token stream to the parser, such as for syntax checking
* in editors and so on.
*
*/
// [The "BSD license"]
// Copyright (c) 2005-2009 Gokulakannan Somasundaram.
//
// All rights reserved.
//
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions
// are met:
// 1. Redistributions of source code must retain the above copyright
//    notice, this list of conditions and the following disclaimer.
// 2. Redistributions in binary form must reproduce the above copyright
//    notice, this list of conditions and the following disclaimer in the
//    documentation and/or other materials provided with the distribution.

```

```

// 3. The name of the author may not be used to endorse or promote products
// derived from this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
// IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
// OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
// IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
// INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
// NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
// THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

#ifndef _<name>_H
#define _<name>_H
<actions.(actionScope).preincludes>
/* =====
* Standard antlr3 C++ runtime definitions
*/
#include \<antlr3.hpp>

/* End of standard antlr 3 runtime definitions
* =====
*/

<actions.(actionScope).includes>

<if(recognizer.grammar.delegates)>
// Include delegate definition header files
//
<recognizer.grammar.delegates: {g|#include "<g.recognizerName>.hpp"}; separator="\n">

<endif>

<actions.(actionScope).header>

#ifdef WIN32
// Disable: Unreferenced parameter, - Rules with parameters that are not used
// constant conditional, - ANTLR realizes that a prediction is always true (synpred usually)
// initialized but unused variable - tree rewrite variables declared but not needed
// Unreferenced local variable - lexer rule declares but does not always use _type
// potentially uninitialized variable used - retval always returned from a rule
// unreferenced local function has been removed - susually getTokenNames or freeScope, they can go without
warnigns
//
// These are only really displayed at warning level /W4 but that is the code ideal I am aiming at

```

```

// and the codegen must generate some of these warnings by necessity, apart from 4100, which is
// usually generated when a parser rule is given a parameter that it does not use. Mostly though
// this is a matter of orthogonality hence I disable that one.
//
#pragma warning( disable : 4100 )
#pragma warning( disable : 4101 )
#pragma warning( disable : 4127 )
#pragma warning( disable : 4189 )
#pragma warning( disable : 4505 )
#pragma warning( disable : 4701 )
#endif
<if(backtracking)>

/* =====
* BACKTRACKING IS ENABLED
* =====
*/
<endif>

<beginNamespace(actions)>

<if(recognizer.grammar.delegators)>
// Include delegator definition classes
//
<recognizer.grammar.delegators: {g|class <g.recognizerName>; }; separator="\n">
<endif>

<actions.(actionScope).traits>
typedef <name>Traits <name>ImplTraits;

<rules:{r | <if(r.ruleDescriptor.isSynPred)> struct <r.ruleDescriptor.name> {}; <endif>}; separator="\n">

class <name>Tokens
{
public:
/** Symbolic definitions of all the tokens that the <grammarType()> will work with.
*
* Antlr will define EOF, but we can't use that as it is too common in
* in C header files and that would be confusing. There is no way to filter this out at the moment
* so we just undef it here for now. That isn't the value we get back from C recognizers
* anyway. We are looking for ANTLR_TOKEN_EOF.
*/
enum Tokens
{
EOF_TOKEN = <name>ImplTraits::CommonTokenType::TOKEN_EOF
<tokens:{it | , <it.name> = <it.type> }; separator="\n">
};

```

```

};

/** Context tracking structure for <mainName()>
*/
class <name> : public <componentBaseType()>, public <name>Tokens
{
public:
    typedef <name>ImplTraits ImplTraits;
    typedef <name> ComponentType;
    typedef ComponentType::StreamType StreamType;
    typedef <componentBaseType()> BaseType;
    typedef ImplTraits::RecognizerSharedStateType\<StreamType> RecognizerSharedStateType;
    typedef StreamType InputType;
    <if(recognizer.filterMode)>
        static const bool IsFiltered = true;
    <else>
        static const bool IsFiltered = false;
    <endif>

    <scopes:{ it | <if(it.isDynamicGlobalScope)><globalAttributeScopeDecl(it)><endif> }>
    <rules:{ r | <if(r.ruleDescriptor.ruleScope)><ruleAttributeScopeDecl(scope=r.ruleDescriptor.ruleScope)><endif> }>

private:
    <if(recognizer.grammar.delegates)>
        <recognizer.grammar.delegates:
            {g|<g.recognizerName>* m_<g:delegateName()>;}; separator="\n">
        <endif>
    <if(recognizer.grammar.delegators)>
        <recognizer.grammar.delegators:
            {g|<g.recognizerName>* m_<g:delegateName()>;}; separator="\n">
        <endif>
    <scopes:{ it | <if(it.isDynamicGlobalScope)>
        <globalAttributeScopeDef(it)>
    <endif> }; separator="\n\n">
    <rules: { r |<if(r.ruleDescriptor.ruleScope)>
        <ruleAttributeScopeDef(scope=r.ruleDescriptor.ruleScope)>
    <endif> }>
    <@members>
    <@end>

public:
    <name>(InputType* instream<recognizer.grammar.delegators:{g|, <g.recognizerName>*
    <g:delegateName()>}>);
    <name>(InputType* instream, RecognizerSharedStateType* state<recognizer.grammar.delegators:{g|,
    <g.recognizerName>* <g:delegateName()>}>);

    void init(InputType* instream <recognizer.grammar.delegators:{g|, <g.recognizerName>* <g:delegateName()>}>
);

```

```

    <actions.(actionScope).context>

<if(LEXER)>
<if(recognizer.filterMode)>
    void memoize(ANTLR_MARKER ruleIndex, ANTLR_MARKER ruleParseStart);
    bool alreadyParsedRule(ANTLR_MARKER ruleIndex);
    <filteringNextToken()>
<endif>
    <rules:{r | <if(!r.ruleDescriptor.isSynPred)><headerReturnType(ruleDescriptor=r.ruleDescriptor)>
m<r.ruleDescriptor.name>( <r.ruleDescriptor.parameterScope:parameterScope()>);<endif>}; separator="\n">
    <rules:{r | <if(r.ruleDescriptor.isSynPred)> <headerReturnType(ruleDescriptor=r.ruleDescriptor)> msynpred(
antlr3::ClassForwarder< <r.ruleDescriptor.name> > <r.ruleDescriptor.parameterScope:parameterScope()>);
    void m<r.ruleDescriptor.name>_fragment (<r.ruleDescriptor.parameterScope:parameterScope()>);<endif>};
separator="\n">
<endif>
<if(!LEXER)>
    <rules:{r | <headerReturnScope(ruleDescriptor=r.ruleDescriptor)>}>
    <rules:{r | <if(!r.ruleDescriptor.isSynPred)> <headerReturnType(ruleDescriptor=r.ruleDescriptor)>
<r.ruleDescriptor.name> (<r.ruleDescriptor.parameterScope:parameterScope()>); <endif>}; separator="\n">
    <rules:{r | <if(r.ruleDescriptor.isSynPred)> <headerReturnType(ruleDescriptor=r.ruleDescriptor)> msynpred(
antlr3::ClassForwarder< <r.ruleDescriptor.name> > <r.ruleDescriptor.parameterScope:parameterScope()>);
    void m<r.ruleDescriptor.name>_fragment (<r.ruleDescriptor.parameterScope:parameterScope()>);<endif>};
separator="\n">
<! generate rule/method definitions for imported rules so they
appear to be defined in this recognizer. !>
    // Delegated rules
<recognizer.grammar.delegatedRules:{ruleDescriptor|
    <headerReturnType(ruleDescriptor)>
<ruleDescriptor.name>( <ruleDescriptor.parameterScope:parameterScope()>);}; separator="\n">
<endif>

    const char * getGrammarFileName();
    void reset();
    ~<name>();

};

// Function prototypes for the constructor functions that external translation units
// such as delegators and delegates may wish to call.
//
<if(!recognizer.grammar.grammarIsRoot)>
extern ANTLR_UINT8* <recognizer.grammar.composite.rootGrammar.recognizerName>TokenNames[];
<endif>

/* End of token definitions for <name>
* =====

```

```

*/

<endNamespace(actions)>

#endif

/* END - Note:Keep extra line feed to satisfy UNIX systems */

>>

grammarType() ::= <%
<if(PARSER)>
parser
<endif>
<if(LEXER)>
lexer
<endif>
<if(TREE_PARSER)>
tree parser
<endif>
%>

componentType() ::= <<
<if(PARSER)>
<name>ImplTraits::ParserType
<endif>
<if(LEXER)>
<name>ImplTraits::LexerType
<endif>
<if(TREE_PARSER)>
<name>ImplTraits::TreeParserType
<endif>
>>

componentBaseType() ::= <%
<if(PARSER)>
<name>ImplTraits::BaseParserType
<endif>
<if(LEXER)>
<name>ImplTraits::BaseLexerType
<endif>
<if(TREE_PARSER)>
<name>ImplTraits::BaseTreeParserType
<endif>
%>

streamType() ::= <<
<if(PARSER)>

```

```

<name>ImplTraits::ParserType::StreamType
<endif>
<if(LEXER)>
<name>ImplTraits::LexerType::StreamType
<endif>
<if(TREE_PARSER)>
<name>ImplTraits::TreeParserType::StreamType
<endif>
>>

```

```

mainName() ::= <%
<if(PARSER)>
<name>
<endif>
<if(LEXER)>
<name>
<endif>
<if(TREE_PARSER)>
<name>
<endif>
%>

```

```

headerReturnScope(ruleDescriptor) ::= "<returnScope(scope=ruleDescriptor.returnScope)>"

```

```

headerReturnType(ruleDescriptor) ::= <%
<if(LEXER)>
<if(!ruleDescriptor.isSynPred)>
void
<else>
<returnType()>
<endif>
<else>
<returnType()>
<endif>
%>

```

```

// Produce the lexer output
//
lexer( grammar,
name,
tokens,
scopes,
rules,
numRules,
filterMode,
superClass,
labelType="ImplTraits::CommonTokenType*") ::= <<

```

```

using namespace antlr3;

<beginNamespace(actions)>

<if(filterMode)>

/* Override the normal MEMOIZE and HAVEALREADYPARSED macros as this is a filtering
 * lexer. In filter mode, the memoizing and backtracking are gated at BACKTRACKING > 1 rather
 * than just BACKTRACKING. In some cases this might generate code akin to:
 * if (BACKTRACKING) if (BACKTRACKING > 1) memoize.
 */
void <name>::memoize(ANTLR_MARKER ruleIndex, ANTLR_MARKER ruleParseStart)
{
    BaseType* base = this;
    if ( this->get_backtracking()>1 )
        base->memoize( ruleIndex, ruleParseStart );
}

bool <name>::alreadyParsedRule(ANTLR_MARKER ruleIndex)
{
    BaseType* base = this;
    if ( this->get_backtracking() > 1 )
        return base->haveParsedRule(ruleIndex);
    return false;
}

<endif>

/* =====
 * Lexer matching rules end.
 * =====
 */

<scopes:{it |<if(it.isDynamicGlobalScope)><globalAttributeScope(it)><endif>}>

<actions.lexer.members>

<name>::~~<name>()
{
    <if(memoize)>
    RuleMemoType* rulememo = this->getRuleMemo();
    if(rulememo != NULL)
    {
        delete rulememo;
        this->setRuleMemo(NULL);
    }
}

```

```

<endif>
<if(grammar.directDelegates)>
  // Free the lexers that we delegated to
  // functions to. NULL the state so we only free it once.
  //
  <grammar.directDelegates:
    {g| m_<g:delegateName()->set_lexstate(NULL);
    delete m_<g:delegateName()->; }; separator="\n">
<endif>
}

void
<name>::reset()
{
  this->get_rec()->reset();
}

/** \brief Name of the grammar file that generated this code
*/
static const char fileName[] = "<fileName>";

/** \brief Return the name of the grammar file that generated this code.
*/
const char* <name>::getGrammarFileName()
{
  return fileName;
}

/** \brief Create a new lexer called <name>
*
* \param[in] instream Pointer to an initialized input stream
* \return
* - Success p<name> initialized for the lex start
* - Fail NULL
*/
<name>::<name>(StreamType* instream<grammar.delegators:{g|, <g.recognizerName>* <g:delegateName()->>>
:<name>ImplTraits::BaseLexerType(ANTLR_SIZE_HINT, instream, NULL)
{
  // See if we can create a new lexer with the standard constructor
  //
  this->init(instream <grammar.delegators:{g|, <g:delegateName()->>>);
}

/** \brief Create a new lexer called <name>
*
* \param[in] instream Pointer to an initialized input stream
* \param[state] state Previously created shared recognizer stat
* \return

```

```

* - Success p<name> initialized for the lex start
* - Fail NULL
*/
<name>::<name>(StreamType* instream, RecognizerSharedStateType* state<grammar.delegators:{g|,
<g.recognizerName>* <g.delegateName()>>})
:<name>ImplTraits::BaseLexerType(ANTLR_SIZE_HINT, instream, state)
{
    this->init(instream <grammar.delegators:{g|, <g.delegateName()> >});
}

void <name>::init(StreamType* instream<grammar.delegators:{g|, <g.recognizerName>* <g.delegateName()> > )
{
    /* -----
    * Memory for basic structure is allocated, now to fill in
    * in base ANTLR3 structures. We initialize the function pointers
    * for the standard ANTLR3 lexer function set, but upon return
    * from here, the programmer may set the pointers to provide custom
    * implementations of each function.
    *
    * We don't use the macros defined in <name>.h here so you can get a sense
    * of what goes where.
    */

<if(memoize)>
<if(grammar.grammarIsRoot)>
    // Create a LIST for recording rule memos.
    //
    this->setRuleMemo( new IntTrie(15) ); /* 16 bit depth is enough for 32768 rules! */
<endif>
<endif>

<if(grammar.directDelegates)>
    // Initialize the lexers that we are going to delegate some
    // functions to.
    //
    <grammar.directDelegates:
        {g|m_<g.delegateName()> = new <g.recognizerName>(instream, this->get_lexstate(),
this<grammar.delegators:{g|, <g.delegateName()>>}); separator="\n">
    <endif>
<if(grammar.delegators)>
    // Install the pointers back to lexers that will delegate us to perform certain functions
    // for them.
    //
    <grammar.delegators:
        {g| m_<g.delegateName()> = <g.delegateName()>; separator="\n">
    <endif>
}

```

```

<if(cyclicDFAs)>

/* =====
* DFA tables for the lexer
*/
<cyclicDFAs:cyclicDFA() <! dump tables for all DFA !>
/* =====
* End of DFA tables for the lexer
*/
<endif>

/* =====
* Functions to match the lexer grammar defined tokens from the input stream
*/

<rules; separator="\n\n">

/* =====
* Lexer matching rules end.
* =====
*/
<if(synpreds)>

/* =====
* Lexer syntactic predicates
*/
<synpreds: {p | <lexerSynpred(predname=p)>}>
/* =====
* Lexer syntactic predicates end.
* =====
*/
<endif>

/* End of Lexer code
* =====
* =====
*/

<endNamespace(actions)>

>>

filteringNextToken() ::= <<
<name>ImplTraits::CommonTokenType*
<name>ImplTraits::TokenSourceType::nextToken()
{
  LexerType* lexer;

```

```

typename LexerType::RecognizerSharedStateType* state;

lexer = this->get_super();
state = lexer->get_lexstate();

/* Get rid of any previous token (token factory takes care of
 * any deallocation when this token is finally used up.
 */
state->set_token_present( false );
state->set_error( false ); /* Start out without an exception */
state->set_failedflag(false);

/* Record the start of the token in our input stream.
 */
state->set_tokenStartCharIndex( lexer->index());
state->set_tokenStartCharPositionInLine( lexer->getCharPositionInLine() );
state->set_tokenStartLine( lexer->getLine() );
state->set_text("");

/* Now call the matching rules and see if we can generate a new token
 */
for (;;)
{
if (lexer->LA(1) == ANTLR_CHARSTREAM_EOF)
{
/* Reached the end of the stream, nothing more to do.
 */
CommonTokenType& teof = m_eofToken;

teof.set_startIndex(lexer->getCharIndex());
teof.set_stopIndex(lexer->getCharIndex());
teof.setLine(lexer->getLine());
return &teof;
}

state->set_token_present(false);
state->set_error(false); /* Start out without an exception */

{
ANTLR_MARKER m;

m = this->get_istream()->mark();
state->set_backtracking(1); /* No exceptions */
state->set_failedflag(false);

/* Call the generated lexer, see if it can get a new token together.
 */
lexer->mTokens();

```

```

state->set_backtracking(0);

<! mTokens backtracks with synpred at BACKTRACKING==2
and we set the synpredgate to allow actions at level 1. !>

if(state->get_failed())
{
lexer->rewind(m);
lexer->consume(); <! advance one char and try again !>
}
else
{
lexer->emit(); /* Assemble the token and emit it to the stream */
TokenType& tok = state->get_token();
return &tok;
}
}
}
}
}
>>

actionGate() ::= "this->get_backtracking()==0"

filteringActionGate() ::= "this->get_backtracking()==1"

/** How to generate a parser */
genericParser( grammar, name, scopes, tokens, tokenNames, rules, numRules,
              bitsets, inputStreamType, superClass,
              labelType, members, rewriteElementType,
              filterMode, ASTLabelType="ImplTraits::TreeType*") ::= <<

using namespace antlr3;
<if(grammar.grammarIsRoot)>
/** \brief Table of all token names in symbolic order, mainly used for
*   error reporting.
*/
ANTLR_UINT8* <name>TokenNames[<length(tokenNames)>+4]
= {
  (ANTLR_UINT8*) "\<invalid>", /* String to print to indicate an invalid token */
  (ANTLR_UINT8*) "\<EOR>",
  (ANTLR_UINT8*) "\<DOWN>",
  (ANTLR_UINT8*) "\<UP>",
  <tokenNames:{it |(ANTLR_UINT8*) <it>}; separator=",\n">
};
<endif>

<@members>

```

```

<@end>

/** \brief Name of the grammar file that generated this code
*/
static const char fileName[] = "<fileName>";

/** \brief Return the name of the grammar file that generated this code.
*/
const char* <name>::getGrammarFileName()
{
    return fileName;
}

/** \brief Create a new <name> parser and return a context for it.
*
* \param[in] instream Pointer to an input stream interface.
*
* \return Pointer to new parser context upon success.
*/
<name>::<name>( StreamType* instream<grammar.delegators:{g|,<g.recognizerName>* <g.delegateName()>}>
<constructorInitializerType("NULL")>
{
    // See if we can create a new parser with the standard constructor
    //
    this->init(instream<grammar.delegators:{g|,<g.delegateName()>}>);
}

/** \brief Create a new <name> parser and return a context for it.
*
* \param[in] instream Pointer to an input stream interface.
*
* \return Pointer to new parser context upon success.
*/
<name>::<name>( StreamType* instream, RecognizerSharedStateType* state<grammar.delegators:{g|,
<g.recognizerName>* <g.delegateName()>}>
<constructorInitializerType("state")>
{
    this->init(instream <grammar.delegators:{g|,<g.delegateName()>}>);
}

void <name>::init(StreamType* instream<grammar.delegators:{g|,<g.recognizerName>* <g.delegateName()>}>)
{
    <actions.parser.apifuncs>
<if(memoize)>
<if(grammar.grammarIsRoot)>
    /* Create a LIST for recording rule memos.
    */
    typedef RecognizerSharedStateType::RuleMemoType RuleMemoType;
    this->setRuleMemo( new RuleMemoType(15) ); /* 16 bit depth is enough for 32768 rules! */<\n>

```

```

<endif>
<endif>
<if(grammar.directDelegates)>
// Initialize the lexers that we are going to delegate some
// functions to.
//
<grammar.directDelegates:
    {g|m_<g:delegateName()> = new <g.recognizerName>(instream, this->get_psrstate(),
this<grammar.delegates:{g|, <g:delegateName()>>}); separator="\n">
<endif>
    <if(grammar.delegates)>
// Install the pointers back to lexers that will delegate us to perform certain functions
// for them.
//
    <grammar.delegates: {g| m_<g:delegateName()> = <g:delegateName()>; separator="\n">
<endif>
/* Install the token table
*/
this->get_psrstate()->set_tokenNames( <grammar.composite.rootGrammar.recognizerName>TokenNames );

<@debugStuff()>

}

void
<name>::reset()
{
    this->get_rec()->reset();
}

/** Free the parser resources
*/
<name>::~~<name>()
{
    <@cleanup>
    <@end>
<if(grammar.directDelegates)>
// Free the parsers that we delegated to
// functions to.NULL the state so we only free it once.
//
<grammar.directDelegates:
    {g| m_<g:delegateName()>->set_psrstate( NULL );
    delete m_<g:delegateName()>;}; separator="\n">
<endif>
<if(memoize)>
<if(grammar.grammarIsRoot)>
if(this->getRuleMemo() != NULL)
{

```

```

delete this->getRuleMemo();
this->setRuleMemo(NULL);
}
<endif>
<endif>
}

/** Return token names used by this <grammarType()>
 *
 * The returned pointer is used as an index into the token names table (using the token
 * number as the index).
 *
 * \return Pointer to first char * in the table.
 */
static ANTLR_UINT8** getTokenNames()
{
    return <grammar.composite.rootGrammar.recognizerName>TokenNames;
}

<members>

/* Declare the bitsets
 */
<bitsets:{it | <bitsetDeclare(bitsetName={ FOLLOW_<it.name>_in_<it.inName><it.tokenIndex> },
    words64=it.bits, traits={<name>ImplTraits } )>>}>

<if(cyclicDFAs)>

/* =====
 * DFA tables for the parser
 */
<cyclicDFAs:cyclicDFA()> <! dump tables for all DFA !>
/* =====
 * End of DFA tables for the parser
 */
<endif>

/* =====
 * Parsing rules
 */
<rules; separator="\n\n">
<if(grammar.delegatedRules)>
// Delegated methods that appear to be a part of this
// parser
//
<grammar.delegatedRules:{ruleDescriptor|
    <return Type()> <name>::<ruleDescriptor.name>(<ruleDescriptor.parameterScope:parameterScope()>)

```

```

    {
        <if(ruleDescriptor.hasReturnValue)>return <endif>m_<ruleDescriptor.grammar:delegateName()-
    ><ruleDescriptor.name>(<if(ruleDescriptor.parameterScope)><ruleDescriptor.parameterScope.attributes:{a|<a.name
    e>}; separator=", "><endif>);
        \}); separator="\n">

<endif>
/* End of parsing rules
* =====
*/

/* =====
* Syntactic predicates
*/
<synpreds:{p | <synpred(predname=p)>}>
/* End of syntactic predicates
* =====
*/

>>

constructorInitializerType(rec_state) ::= <<
<if(PARSER)>
    :ImplTraits::BaseParserType(ANTLR_SIZE_HINT, instream, <rec_state>)
<endif>
<if(TREE_PARSER)>
    :ImplTraits::BaseTreeParserType(ANTLR_SIZE_HINT, instream, <rec_state>)
<endif>
>>

parser( grammar,
    name,
    scopes,
    tokens,
    tokenNames,
    rules,
    numRules,
    bitsets,
    ASTLabelType,
    superClass="Parser",
    labelType="ImplTraits::CommonTokenType*",
    members={<actions.parser.members>}
) ::= <<
<beginNamespace(actions)>
<genericParser(inputStreamType="CommonTokenStreamType*", rewriteElementType="Token", filterMode=false,
...)>
<endNamespace(actions)>
>>

```

```

/** How to generate a tree parser; same as parser except the input
 * stream is a different type.
 */
treeParser( grammar,
    name,
    scopes,
    tokens,
    tokenNames,
    globalAction,
    rules,
    numRules,
    bitsets,
    filterMode,
    labelType={ <ASTLabelType> },
    ASTLabelType="ImplTraits::TreeType*",
    superClass="TreeParser",
    members={ <actions.treeparser.members> }
) ::= <<
<beginNamespace(actions)>
<genericParser(inputStreamType="CommonTreeNodeStream*", rewriteElementType="Node", ...)>
<endNamespace(actions)>
>>

/** A simpler version of a rule template that is specific to the imaginary
 * rules created for syntactic predicates. As they never have return values
 * nor parameters etc..., just give simplest possible method. Don't do
 * any of the normal memoization stuff in here either; it's a waste.
 * As predicates cannot be inlined into the invoking rule, they need to
 * be in a rule by themselves.
 */
synpredRule(ruleName, ruleDescriptor, block, description, nakedBlock) ::=
<<
// $ANTLR start <ruleName>
void <name>::m<ruleName>_fragment( <ruleDescriptor.parameterScope:parameterScope(> )
{
<ruleLabelDefs(>
<ruleLabelInitializations(>
<if(trace)>
    ANTLR_PRINTF("enter <ruleName> %d failed = %d, backtracking = %d\\n", this->LT(1),failed,this-
>get_backtracking() );
    <block>
    ANTLR_PRINTF("exit <ruleName> %d, failed = %d, backtracking = %d\\n", this->LT(1),failed,this-
>get_backtracking());

<else>
    <block>
<endif>

```

```

goto rule<ruleDescriptor.name>Ex; /* Prevent compiler warnings */
rule<ruleDescriptor.name>Ex: ;
}
// $ANTLR end <ruleName>
>>

synpred(predname) ::= <<

bool <name>::msynpred( antlr3::ClassForwarder\< <predname> > )
{
    ANTLR_MARKER start;
    bool success;

    this->inc_backtracking();
    <@start()>
    start = this->mark();
    this->m<predname>_fragment(); // can never throw exception
    success = !( this->get_failedflag() );
    this->rewind(start);
    <@stop()>
    this->dec_backtracking();
    this->set_failedflag(false);
    return success;
}<\n>
>>

lexerSynpred(predname) ::= <<
<synpred(predname)>
>>

ruleMemoization(rname) ::= <<
<if(memoize)>
if ( (this->get_backtracking()>0) && (this->haveParsedRule(<ruleDescriptor.index>)) )
{
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(!ruleDescriptor.isSynPred)>
    retval.start = 0;<\n>
<endif>
<endif>
    <(ruleDescriptor.actions.after):execAfter()>
    <finalCode(finalBlock=finally)>
<if(!ruleDescriptor.isSynPred)>
    <scopeClean()><\n>
<endif>
    return <ruleReturnValue()>;
}
<endif>

```

>>

/** How to test for failure and return from rule */

```
checkRuleBacktrackFailure() ::= <<
if (this->hasException())
{
    goto rule<ruleDescriptor.name>Ex;
}
<if(backtracking)>
if (this->hasFailed())
{
    <scopeClean()>
    <@debugClean()>
    return <ruleReturnValue()>;
}
<endif>
>>
```

/** This rule has failed, exit indicating failure during backtrack */

```
ruleBacktrackFailure() ::= <<
<if(backtracking)>
if (this->get_backtracking()>0)
{
    this->set_failedflag( true );
    <scopeClean()>
    return <ruleReturnValue()>;
}
<endif>
>>
```

/** How to generate code for a rule. This includes any return type

* data aggregates required for multiple return values.

*/

```
rule(ruleName,ruleDescriptor,block,emptyRule,description,exceptions,finally,memoize) ::= <<
```

/**

* \$ANTLR start <ruleName>

* <fileName>:<description>

*/

```
<returnType()>
```

```
<name::<ruleName>( <ruleDescriptor.parameterScope:parameterScope()> )
```

```
{
```

```
<if(trace)>ANTLR_PRINTF("enter <ruleName> %s failed=%d, backtracking=%d\n", this->LT(1), this->get_backtracking() );<endif>
```

```
<ruleDeclarations()>
```

```
<ruleDescriptor.actions.declarations>
```

```
<ruleLabelDefs()>
```

```
<ruleInitializations()>
```

```
<ruleDescriptor.actions.init>
```

```

<ruleMemoization(rname=ruleName)>
<ruleLabelInitializations()>

<if(actions.(actionScope).rulecatch)>
  try {
<else>
<if(exceptions)>
  try {
<endif>
<endif>
<@preamble()>
{
  <block>
}
<ruleCleanUp()>

<if(exceptions)>
  <(ruleDescriptor.actions.after):execAfter()>
  <exceptions:{e|<catch(decl=e.decl,action=e.action)><\n}>>
<else>
<if(!emptyRule)>
  if (this->hasException())
  {
    this->preporterror();
    this->precover();
    <@setErrorReturnValue()>
  }
  <if(ruleDescriptor.actions.after)>
  else
  {
    <(ruleDescriptor.actions.after):execAfter()>
  }<\n>
  <endif>
</if(actions.(actionScope).rulecatch)>
} <actions.(actionScope).rulecatch>
<endif>
<endif>
<endif>

<if(trace)>ANTLR_PRINTF("exit <ruleName> %d failed=%s backtracking=%s\n", this->LT(1), failed, this-
>get_backtracking() );<endif>
<memoize()>
<if(finally)>
  <finalCode(finalBlock=finally)>
<endif>
<scopeClean()>
<@postamble()>
return <ruleReturnValue()>;

```

```

}
/* $ANTLR end <ruleName> */
>>

finalCode(finalBlock) ::= <<
{
  <finalBlock>
}

>>

catch(decl,action) ::= <<
/* catch(decl,action)
*/
}catch (<e.decl>) {
  <e.action>
}

>>

ruleDeclarations() ::= <<

<if(ruleDescriptor.hasMultipleReturnValues)>
<returnType()> retval(this);<\n>
<else>
<if(PARSER)>
  <name>ImplTraits::RuleReturnValueType _antlr_rule_exit(this);
<endif>
<if(ruleDescriptor.returnScope)>
<ruleDescriptor.returnScope.attributes:{ a |
<a.type> <a.name>;
}>
<endif>
<endif>
<if(memoize)>
ANTLR_MARKER <ruleDescriptor.name>_StartIndex;
<endif>
>>

ruleInitializations() ::= <<
/* Initialize rule variables
*/
<if(ruleDescriptor.returnScope)>
<if(ruleDescriptor.hasMultipleReturnValues)>
<ruleDescriptor.returnScope.attributes:{ a | <if(a.initValue)>retval.<a.name> = <a.initValue>;<endif> }>
<else>
<ruleDescriptor.returnScope.attributes:{ a | <if(a.initValue)><a.name> = <a.initValue>;<endif> }>
<endif>
<endif>

```

```

<if(memoize)>
<ruleDescriptor.name>_startIndex = this->index();<\n>
<endif>
<ruleDescriptor.useScopes:{it | m_<it>_stack.push(<it>Scope()); }; separator="\n">
<ruleDescriptor.ruleScope:{it | m_<it.name>_stack.push(<it.name>Scope()); }; separator="\n">
>>

ruleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,ruleDescriptor.tokenListLabels,
ruleDescriptor.wildcardTreeLabels,ruleDescriptor.wildcardTreeListLabels]
: {it | <labelType> <it.label.text> = NULL;}; separator="\n"
>
<[ruleDescriptor.tokenListLabels,ruleDescriptor.ruleListLabels,ruleDescriptor.wildcardTreeListLabels]
: {it | ImplTraits::TokenPtrsListType list_<it.label.text>;}; separator="\n"
>
<ruleDescriptor.ruleLabels:ruleLabelDef(); separator="\n">
>>

ruleLabelInitializations() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(!ruleDescriptor.isSynPred)>
retval.call_start_placeholder();
<endif>
<endif>
>>

lexerRuleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,
ruleDescriptor.tokenListLabels,
ruleDescriptor.ruleLabels]
: {it | <labelType> <it.label.text> = NULL;}; separator="\n"
>
<ruleDescriptor.charLabels:{it | ANTLR_UINT32 <it.label.text>;}; separator="\n">
<[ruleDescriptor.tokenListLabels,
ruleDescriptor.ruleListLabels,
ruleDescriptor.ruleListLabels]
: {it | ImplTraits::IntTrieType<CommonTokenType>* list_<it.label.text>;}; separator="\n"
>
>>

lexerRuleLabelInit() ::= <<
<[ruleDescriptor.tokenListLabels,
ruleDescriptor.ruleListLabels,
ruleDescriptor.ruleListLabels]
: {it | list_<it.label.text> = new ImplTraits::IntTrieType<CommonTokenType>(31);}; separator="\n"
>
>>

```

```

lexerRuleLabelFree() ::= <<
<[ruleDescriptor.tokenLabels,
ruleDescriptor.tokenListLabels,
ruleDescriptor.ruleLabels]
:{it |<it.label.text> = NULL;}; separator="\n"
>
<[ruleDescriptor.tokenListLabels,
ruleDescriptor.ruleListLabels,
ruleDescriptor.ruleListLabels]
:{it | delete list_<it.label.text>;}; separator="\n"
>
>>

ruleReturnValue() ::= <%
<if(!ruleDescriptor.isSynPred)>
<if(ruleDescriptor.hasReturnValue)>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnName>
<else>
retval
<endif>
<endif>
<endif>
%>

memoize() ::= <<
<if(memoize)>
<if(backtracking)>
if ( this->get_backtracking() > 0 ) { this->memoize(<ruleDescriptor.index>, <ruleDescriptor.name>_StartIndex); }
<endif>
<endif>
>>

ruleCleanUp() ::= <<

// This is where rules clean up and exit
//
goto rule<ruleDescriptor.name>Ex; /* Prevent compiler warnings */
rule<ruleDescriptor.name>Ex; ;
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(!TREE_PARSER)>
<if(!ruleDescriptor.isSynPred)>
retval.call_stop_placeholder();<\n>
<endif>
<endif>
<endif>
>>

```

```

scopeClean() ::= <<
<ruleDescriptor.useScopes:{it | m_<it>_stack.pop(); }; separator="\n">
<ruleDescriptor.ruleScope:{it | m_<it.name>_stack.pop(); }; separator="\n">

>>

/** How to generate a rule in the lexer; naked blocks are used for
 * fragment rules, which do not produce tokens.
 */
lexerRule(ruleName,nakedBlock,ruleDescriptor,block,memoize) ::= <<
// Comes from: <block.description>
/** \brief Lexer rule generated by ANTLR3
 *
 * $ANTLR start <ruleName>
 *
 * Looks to match the characters the constitute the token <ruleName>
 * from the attached input stream.
 *
 *
 * \remark
 * - lexer->error == true if an exception was thrown.
 */
void <name>::m<ruleName>(<ruleDescriptor.parameterScope:parameterScope(>))
{
    ANTLR_UINT32 _type;
    <ruleDeclarations(>
    <ruleDescriptor.actions.declarations>
    <lexerRuleLabelDefs(>
    <if(trace>
    std::cout <<< "enter <ruleName> " <<< (char)this->LA(1)
        <<< " line=" <<< this->getLine() <<< ":" <<< this->getCharPositionInLine()
        <<< " failed=" <<< this->get_failedflag() <<< " backtracking=" <<< this->get_backtracking() <<<
    std::endl;
    <endif>

    <if(nakedBlock>
        <ruleMemoization(rname=ruleName)>
        <lexerRuleLabelInit(>
        <ruleDescriptor.actions.init>

        <block><\n>
    <else>
        <ruleMemoization(rname=ruleName)>
        <lexerRuleLabelInit(>
        _type = <ruleName>;

        <ruleDescriptor.actions.init>

        <block>

```

```

this->get_lexstate()->set_type(_type);
<endif>
<if(trace)>
std::cout <<< "exit <ruleName> " <<< (char)this->LA(1)
    <<< " line=" <<< this->getLine() <<< ":" <<< this->getCharPositionInLine()
    <<< " failed=" <<< this->get_failedflag() <<< " backtracking=" <<< this->get_backtracking() <<<
std::endl;
<endif>
<ruleCleanUp()>
<lexerRuleLabelFree()>
<(ruleDescriptor.actions.after):execAfter()>
<memoize>
}
// $ANTLR end <ruleName>
>>

/** How to generate code for the implicitly-defined lexer grammar rule
 * that chooses between lexer rules.
 */
tokensRule(ruleName,nakedBlock,args,block,ruleDescriptor) ::= <<
/** This is the entry point in to the lexer from an object that
 * wants to generate the next token, such as a pCOMMON_TOKEN_STREAM
 */
void
<name>::mTokens()
{
    <block><\n>

    goto ruleTokensEx; /* Prevent compiler warnings */
ruleTokensEx: ;
}
>>

// S U B R U L E S

/** A (...) subrule with multiple alternatives */
block(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<

// <fileName>:<description>
{
    int alt<decisionNumber>=<maxAlt>;
    <decls>
    <@predecision()>
    <decision>
    <@postdecision()>
    <@prebranch()>
    switch (alt<decisionNumber>)
    {

```

```

<alts:{ a | <altSwitchCase(i,a)> }>
}
<@postbranch()>
}
>>

/** A rule block with multiple alternatives */
ruleBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
{
// <fileName>:<description>

ANTLR_UINT32 alt<decisionNumber>;

alt<decisionNumber>=<maxAlt>;

<decls>
<@predecision()>
<decision>
<@postdecision()>
switch (alt<decisionNumber>)
{
<alts:{ a | <altSwitchCase(i,a)> }>
}
}
>>

ruleBlockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
// <fileName>:<description>
<decls>
<@prealt()>
<alts>
<@postalt()>
>>

/** A special case of a (...) subrule with a single alternative */
blockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
// <fileName>:<description>
<decls>
<@prealt()>
<alts>
<@postalt()>
>>

/** A (..)+ block with 1 or more alternatives */
positiveClosureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
// <fileName>:<description>
{

```

```

int cnt<decisionNumber>=0;
<decls>
<@preloop()>

for (;;)
{
    int alt<decisionNumber>=<maxAlt>;
<@predecision()>
<decision>
<@postdecision()>
switch (alt<decisionNumber>)
{
    <alts:{ a | <altSwitchCase(i,a)>}>
    default:

if ( cnt<decisionNumber> >= 1 )
{
    goto loop<decisionNumber>;
}
<ruleBacktrackFailure()>
<earlyExitEx()>
<@earlyExitException()>
goto rule<ruleDescriptor.name>Ex;
}
cnt<decisionNumber>++;
}
loop<decisionNumber>; /* Jump to here if this rule does not match */
<@postloop()>
}
>>

earlyExitEx() ::= <<
/* mismatchedSetEx()
*/
new ANTLR_Exception<< <name>ImplTraits, EARLY_EXIT_EXCEPTION, StreamType>( this->get_rec(), "" );
<\n>
>>
positiveClosureBlockSingleAlt ::= positiveClosureBlock

/** A (..)* block with 1 or more alternatives */
closureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::=
<<

// <fileName>:<description>
<decls>

<@preloop()>
for (;;)

```

```

{
  int alt<decisionNumber>=<maxAlt>;
  <@predecision()>
  <decision>
  <@postdecision()>
  switch (alt<decisionNumber>)
  {
    <alts:{ a | <altSwitchCase(i,a)>}>
  default:
    goto loop<decisionNumber>; /* break out of the loop */
    break;
  }
}
loop<decisionNumber>; /* Jump out to here if this rule does not match */
<@postloop()>
>>

closureBlockSingleAlt ::= closureBlock

/** Optional blocks (x)? are translated to (x|) by antlr before code generation
 * so we can just use the normal block template
 */
optionalBlock ::= block

optionalBlockSingleAlt ::= block

/** A case in a switch that jumps to an alternative given the alternative
 * number. A DFA predicts the alternative and then a simple switch
 * does the jump to the code that actually matches that alternative.
 */
altSwitchCase(altNum,alt) ::= <<
case <altNum>:
  <@prealt()>
  <alt>
  break;<\n>
>>

/** An alternative is just a list of elements; at outermost level */
alt(elements,altNum,description,autoAST,outerAlt,treeLevel,rew) ::= <<
// <fileName>:<description>
{
  <@declarations()>
  <@initializations()>
  <elements:element()>
  <rew>
  <@cleanup()>
}
>>

```

```

// ELEMENTS
/** What to emit when there is no rewrite. For auto build
 * mode, does nothing.
 */
noRewrite(rewriteBlockLevel, treeLevel) ::= ""

/** Dump the elements one per line */
element(e) ::= <<
<@prematch(>
<e.el><\n>
>>

/** match a token optionally with a label in front */
tokenRef(token,label,elementIndex,terminalOptions) ::= <<
<if(label)><label> = <endif> this->matchToken(<token>, &FOLLOW_<token>_in_<ruleName><elementIndex>);
<checkRuleBacktrackFailure(>
>>

/** ids+=ID */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<tokenRef(...>
<listLabel(elem=label,...>
>>

listLabel(label,elem) ::= <<
list_<label>.push_back(<elem>);
>>

/** match a character */
charRef(char,label) ::= <<
<if(label)>
<label> = this->LA(1);<\n>
<endif>
this->matchc(<char>);
<checkRuleBacktrackFailure(>
>>

/** match a character range */
charRangeRef(a,b,label) ::= <<
<if(label)>
<label> = this->LA(1);<\n>
<endif>
this->matchRange(<a>, <b>);
<checkRuleBacktrackFailure(>
>>

```

```

/** For now, sets are interval tests and must be tested inline */
matchSet(s,label,elementIndex,terminalOptions,postmatchCode="") ::= <<
<if(label)>
<if(LEXER)>
<label>= this->LA(1);<\n>
<else>
<label>=(<labelType>) this->LT(1);<\n>
<endif>
<endif>
if ( <s> )
{
    this->consume();
    <postmatchCode>
<if(!LEXER)>
    this->set_perror_recovery(false);
<endif>
    <if(backtracking)> this->set_failedflag(false); <\n><endif>
}
else
{
    <ruleBacktrackFailure()>
    <mismatchedSetEx()>
    <@mismatchedSetException()>
<if(LEXER)>
    this->recover();
<else>
<! use following code to make it recover inline;
    this->recoverFromMismatchedSet(&FOLLOW_set_in_<ruleName><elementIndex>);
!>
<endif>
    goto rule<ruleDescriptor.name>Ex;
}<\n>
>>

mismatchedSetEx() ::= <<
new ANTLR_Exception\< <name>ImplTraits, MISMATCHED_SET_EXCEPTION, StreamType>( this->get_rec(),
"" );
<if(PARSER)>
this->get_exception()->set_expectngSet(NULL);
<! use following code to make it recover inline;
this->get_exception()->set_expectngSet( &FOLLOW_set_in_<ruleName><elementIndex> );
!>
<endif>
>>

matchRuleBlockSet ::= matchSet

matchSetAndListLabel(s,label,elementIndex,postmatchCode) ::= <<

```

```

<matchSet(...)>
<listLabel(elem=label,...)>
>>

/** Match a string literal */
lexerStringRef(string,label,elementIndex) ::= <<
<if(label)>
ANTLR_MARKER <label>Start = this->getCharIndex();
ANTLR_UINT32 <label>StartLine<elementIndex> = this->getLine();
ANTLR_UINT32 <label>StartCharPos<elementIndex> = this->getCharPositionInLine();
this->matchs(<string>);
<checkRuleBacktrackFailure()>
<label> = new CommonTokenType;
<label>->set_type( CommonTokenType::TOKEN_INVALID );
<label>->set_startIndex( <label>Start);
<label>->set_stopIndex( this->getCharIndex()-1);
<label>->set_input( this->get_input() );
<label>->set_line( <label>StartLine<elementIndex> );
<label>->set_charPositionInLine( <label>StartCharPos<elementIndex> );
<else>
this->matchs(<string>);
<checkRuleBacktrackFailure()><\n>
<endif>
>>

wildcard(token,label,elementIndex,terminalOptions) ::= <<
<if(label)>
<label>=(<labelType>)this->LT(1);<\n>
<endif>
this->matchAnyToken();
<checkRuleBacktrackFailure()>
>>

wildcardAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<wildcard(...)>
<listLabel(elem=label,...)>
>>

/** Match . wildcard in lexer */
wildcardChar(label, elementIndex) ::= <<
<if(label)>
<label> = this->LA(1);<\n>
<endif>
this->matchAny();
<checkRuleBacktrackFailure()>
>>

wildcardCharListLabel(label, elementIndex) ::= <<

```

```

<wildcardChar(...)>
<listLabel(elem=label,...)>
>>

/** Match a rule reference by invoking it possibly with arguments
 * and a return value or values. The 'rule' argument was the
 * target rule name, but now is type Rule, whose toString is
 * same: the rule name. Now though you can access full rule
 * descriptor stuff.
 */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
this->followPush(FOLLOW_<rule.name>_in_<ruleName><elementIndex>);
<if(label)><label>=<endif><if(scope)>m_<scope:delegateName()-><endif><rule.name><(if(args)><args;
separator=", "><endif>);<n>
this->followPop();
<checkRuleBacktrackFailure()>
>>

/** ids+=r */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(elem=label,...)>
>>

/** A lexer rule reference
 * The 'rule' argument was the target rule name, but now
 * is type Rule, whose toString is same: the rule name.
 * Now though you can access full rule descriptor stuff.
 */
lexerRuleRef(rule,label,args,elementIndex,scope) ::= <<
/* <description> */
<if(label)>
{
    ANTLR_MARKER <label>Start<elementIndex> = this->getCharIndex();
    ANTLR_UINT32 <label>StartLine<elementIndex> = this->getLine();
    ANTLR_UINT32 <label>StartCharPos<elementIndex> = this->getCharPositionInLine();
    <if(scope)>m_<scope:delegateName()-><endif>m<rule.name><(if(scope)>m_<scope:delegateName()-><endif>
<if(args)>, <endif><args; separator=", ">);
    <checkRuleBacktrackFailure()>
    <label> = new CommonTokenType();
    <label>->set_type( CommonTokenType::TOKEN_INVALID);
    <label>->set_startIndex( <label>Start<elementIndex> );
    <label>->set_stopIndex( this->getCharIndex()-1 );
    <label>->set_input( this->get_input() );
    <label>->set_line( <label>StartLine<elementIndex> );
    <label>->set_charPositionInLine( <label>StartCharPos<elementIndex> );
}
<else>

```

```

<if(scope)>m_<scope:delegateName()-><endif>m<rule.name>(<args; separator=", ">);
<checkRuleBacktrackFailure()>
<endif>
>>

/** i+=INT in lexer */
lexerRuleRefAndListLabel(rule,label,args,elementIndex,scope) ::= <<
<lexerRuleRef(...)>
<listLabel(elem=label,...)>
>>

/** EOF in the lexer */
lexerMatchEOF(label,elementIndex) ::= <<
<if(label)>
{
    ANTLR_UINT32 <label>Start<elementIndex>;
    ANTLR_UINT32 <label>StartLine<elementIndex> = this->getLine();
    ANTLR_UINT32 <label>StartCharPos<elementIndex> = this->getCharPositionInLine();
    <labelType> <label>;
    <label>Start<elementIndex> = this->getCharIndex();
    this->matchc(ANTLR_CHARSTREAM_EOF);
    <checkRuleBacktrackFailure()>
    <label> = new CommonTokenType();
    <label>->set_type( CommonTokenType::TOKEN_EOF );
    <label>->set_startIndex(<label>Start<elementIndex>);
    <label>->set_stopIndex(this->getCharIndex()-1);
    <label>->set_input( this->get_input() );
    <label>->set_line( <label>StartLine<elementIndex> );
    <label>->set_charPositionInLine( <label>StartCharPos<elementIndex> );
}
<else>
    this->matchc(ANTLR_CHARSTREAM_EOF);
    <checkRuleBacktrackFailure()>
<endif>
>>

// used for left-recursive rules
recRuleDefArg()          ::= "int <recRuleArg()>"
recRuleArg()            ::= "_p"
recRuleAltPredicate(ruleName,opPrec) ::= "<recRuleArg() \<= <opPrec>"
recRuleSetResultAction() ::= "root_0=$<ruleName>_primary.tree;"
recRuleSetReturnAction(src,name) ::= "$<name>=$<src>.<name>;"

/** match ^(root children) in tree parser */
tree(root, actionsAfterRoot, children, nullableChildList, enclosingTreeLevel, treeLevel) ::= <<
<root:element()>
<actionsAfterRoot:element()>
<if(nullableChildList)>

```

```

if ( this->LA(1)== CommonTokenType::TOKEN_DOWN ) {
    this->matchToken(CommonTokenType::TOKEN_DOWN, NULL);
    <checkRuleBacktrackFailure()>
    <children:element()>
    this->matchToken(CommonTokenType::TOKEN_UP, NULL);
    <checkRuleBacktrackFailure()>
}
<else>
this->matchToken(CommonTokenType::TOKEN_DOWN, NULL);
<checkRuleBacktrackFailure()>
<children:element()>
this->matchToken(CommonTokenType::TOKEN_UP, NULL);
<checkRuleBacktrackFailure()>
<endif>
>>

/** Every predicate is used as a validating predicate (even when it is
 * also hoisted into a prediction expression).
 */
validateSemanticPredicate(pred,description) ::= <<
if ( !( <evalPredicate(...)> ) )
{
    <ruleBacktrackFailure()>
    <newFPE(...)>
}
>>

newFPE() ::= <<
    ExceptionBaseType* ex = new ANTLR_Exception\< <name>ImplTraits, FAILED_PREDICATE_EXCEPTION,
StreamType>( this->get_rec(), "<description>" );
    ex->set_ruleName( "<ruleName>" );
    <\n>
>>

// F i x e d D F A (if-then-else)

dfaState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<

{
    int LA<decisionNumber>_<stateNumber> = this->LA(<k>);
    <edges; separator="\nelse ">
    else
    {
<if(eotPredictsAlt)>
        alt<decisionNumber>=<eotPredictsAlt>;
    <else>
        <ruleBacktrackFailure()>
    }
}

```

```

    <newNVException()>
    goto rule<ruleDescriptor.name>Ex;

<endif>
}
}
>>

newNVException() ::= <<
ExceptionBaseType* ex = new ANTLR_Exception\< <name>ImplTraits, NO_VIABLE_ALT_EXCEPTION,
StreamType>( this->get_rec(), "<description>" );
ex->set_decisionNum( <decisionNumber> );
ex->set_state( <stateNumber> );
<@noViableAltException()>
<\n>
>>

/** Same as a normal DFA state except that we don't examine lookahead
 * for the bypass alternative. It delays error detection but this
 * is faster, smaller, and more what people expect. For (X)? people
 * expect "if ( LA(1)==X ) match(X);" and that's it.
 */
dfaOptionalBlockState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
{
    int LA<decisionNumber>_<stateNumber> = this->LA(<k>);
    <edges; separator="\nelse ">
}
>>

/** A DFA state that is actually the loopback decision of a closure
 * loop. If end-of-token (EOT) predicts any of the targets then it
 * should act like a default clause (i.e., no error can be generated).
 * This is used only in the lexer so that for ('a')* on the end of a rule
 * anything other than 'a' predicts exiting.
 */

dfaLoopbackStateDecls() ::= <<
ANTLR_UINT32 LA<decisionNumber>_<stateNumber>;
>>

dfaLoopbackState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
{
    /* dfaLoopbackState(k,edges,eotPredictsAlt,description,stateNumber,semPredState)
    */
    int LA<decisionNumber>_<stateNumber> = this->LA(<k>);
    <edges; separator="\nelse "><\n>
    <if(eotPredictsAlt)>
    <if(!edges)>
    alt<decisionNumber>=<eotPredictsAlt>; <! if no edges, don't gen ELSE !>

```

```

<else>
  else
  {
alt<decisionNumber>=<eotPredictsAlt>;
  }<\n>
  <endif>
  <endif>
}
>>

/** An accept state indicates a unique alternative has been predicted */
dfaAcceptState(alt) ::= "alt<decisionNumber>=<alt>";

/** A simple edge with an expression. If the expression is satisfied,
 * enter to the target state. To handle gated productions, we may
 * have to evaluate some predicates for this edge.
 */
dfaEdge(labelExpr, targetState, predicates) ::= <<
if ( (<labelExpr><if(predicates)> && (<predicates>)<endif>)
{
  <targetState>
}
>>

// F i x e d D F A (switch case)

/** A DFA state where a SWITCH may be generated. The code generator
 * decides if this is possible: CodeGenerator.canGenerateSwitch().
 */
dfaStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ( this->LA(<k>) )
{
<edges; separator="\n">

default:
<if(eotPredictsAlt)>
  alt<decisionNumber>=<eotPredictsAlt>;
<else>
  <ruleBacktrackFailure()>
  <newNVEException()>
  goto rule<ruleDescriptor.name>Ex;<\n>
<endif>
}<\n>
>>

dfaOptionalBlockStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ( this->LA(<k>) )
{

```

```

    <edges; separator="\n">
} <\n>
>>

dfaLoopbackStateSwitch(k, edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ( this->LA(<k>) )
{
<edges; separator="\n"><\n>
<if(eotPredictsAlt)>
default:
    alt<decisionNumber>=<eotPredictsAlt>;
    break;<\n>
<endif>
} <\n>
>>

dfaEdgeSwitch(labels, targetState) ::= <<
<labels: {it | case <it>:}; separator="\n">
{
    <targetState>
}
    break;
>>

// C y c l i c D F A

/** The code to initiate execution of a cyclic DFA; this is used
 * in the rule to predict an alt just like the fixed DFA case.
 * The <name> attribute is inherited via the parser, lexer, ...
 */
dfaDecision(decisionNumber,description) ::= <<
alt<decisionNumber> = cdfa<decisionNumber>.predict(this, this->get_rec(), this->get_istream(),
cdfa<decisionNumber> );
<checkRuleBacktrackFailure()>
>>

/* Dump DFA tables as static initialized arrays of shorts(16 bits)/characters(8 bits)
 * which are then used to statically initialize the dfa structure, which means that there
 * is no runtime initialization whatsoever, other than anything the C compiler might
 * need to generate. In general the C compiler will lay out memory such that there is no
 * runtime code required.
 */
cyclicDFA(dfa) ::= <<
/** Static dfa state tables for Cyclic dfa:
 * <dfa.description>
 */
static const ANTLR_INT32 dfa<dfa.decisionNumber>_eot[<dfa.numberofStates>] =
{

```

```

<dfa.eof; wrap="\n", separator=" ", null="-1">
};
static const ANTLR_INT32 dfa<dfa.decisionNumber>_eof[<dfa.numberofStates>] =
{
<dfa.eof; wrap="\n", separator=" ", null="-1">
};
static const ANTLR_INT32 dfa<dfa.decisionNumber>_min[<dfa.numberofStates>] =
{
<dfa.min; wrap="\n", separator=" ", null="-1">
};
static const ANTLR_INT32 dfa<dfa.decisionNumber>_max[<dfa.numberofStates>] =
{
<dfa.max; wrap="\n", separator=" ", null="-1">
};
static const ANTLR_INT32 dfa<dfa.decisionNumber>_accept[<dfa.numberofStates>] =
{
<dfa.accept; wrap="\n", separator=" ", null="-1">
};
static const ANTLR_INT32 dfa<dfa.decisionNumber>_special[<dfa.numberofStates>] =
{
<dfa.special; wrap="\n", separator=" ", null="-1">
};

/** Used when there is no transition table entry for a particular state */
static const ANTLR_INT32* dfa<dfa.decisionNumber>_T_empty = NULL;

<dfa.edgeTransitionClassMap.keys:{ table |
static const ANTLR_INT32 dfa<dfa.decisionNumber>_T<i0>[] =
{
<table; separator=" ", wrap="\n", null="-1">
\};<\n>; null = "">

/* Transition tables are a table of sub tables, with some tables
* reused for efficiency.
*/
static const ANTLR_INT32 * const dfa<dfa.decisionNumber>_transitions[] =
{
<dfa.transitionEdgeTables:{ xref[dfa<dfa.decisionNumber>_T<xref>]; separator=" ", wrap="\n", null="NULL">
};

<@errorMethod()>

/* Declare tracking structure for Cyclic DFA <dfa.decisionNumber>
*/
class <name>CyclicDFA<dfa.decisionNumber> : public CyclicDFA\< <name>ImplTraits, <name> >, public
<name>Tokens
{
public:

```

```

typedef CyclicDFA\< <name>ImplTraits, <name> > BaseType;
typedef BaseType::ContextType CtxType;

private:
<if(dfa.specialStateSTs)>
//to maintain C-Target compatibility, we need to make some of ctx functions look like member funcs
CtxType* m_ctx;
<endif>

public:
<name>CyclicDFA<dfa.decisionNumber>( ANTLR_INT32 decisionNumber
, const ANTLR_UCHAR* description
, const ANTLR_INT32* const eot
, const ANTLR_INT32* const eof
, const ANTLR_INT32* const min
, const ANTLR_INT32* const max
, const ANTLR_INT32* const accept
, const ANTLR_INT32* const special
, const ANTLR_INT32* const *const transition)
:BaseType( decisionNumber, description, eot, eof, min, max, accept,
special, transition )
{
<if(dfa.specialStateSTs)>
m_ctx = NULL;
<endif>
}

<if(dfa.specialStateSTs)>
ANTLR_UINT32 LA(ANTLR_INT32 i)
{
return m_ctx->LA(i);
}

<if(PARSER)>
const CtxType::CommonTokenType* LT(ANTLR_INT32 k)
{
return m_ctx->LT(k);
}
<endif>
<if(synpreds)>
template\<typename PredType>
bool msynpred( PredType pred )
{
return m_ctx->msynpred(pred);
}
<endif>

ANTLR_INT32 specialStateTransition(CtxType * ctx, RecognizerType* recognizer, IntStreamType* is,

```

```

ANTLR_INT32 s)
{
    ANTLR_INT32 _s;

    m_ctx = ctx;
    _s = s;
    switch (s)
    {
        <dfa.specialStateSTs:{state |
        case <i0>:

        <state>}; separator="\n">
        }
        <if(backtracking)>
            if ( ctx->get_backtracking() > 0)
            {
                ctx->set_failedflag( true );
                return -1;
            }
        <endif>
        ExceptionBaseType* ex = new ANTLR_Exception\< <name>ImplTraits, NO_VIABLE_ALT_EXCEPTION,
StreamType>( recognizer, "<dfa.description>" );
        ex->set_decisionNum( <dfa.decisionNumber> );
        ex->set_state(_s);
        <@noViableAltException()>
        return -1;
    }
    <endif>
};

static <name>CyclicDFA<dfa.decisionNumber> cdfa<dfa.decisionNumber>(
    <dfa.decisionNumber>, /* Decision number of this dfa */
    /* Which decision this represents: */
    (const ANTLR_UCHAR*)<dfa.description>",
    dfa<dfa.decisionNumber>_eot, /* EOT table */
    dfa<dfa.decisionNumber>_eof, /* EOF table */
    dfa<dfa.decisionNumber>_min, /* Minimum tokens for each state */
    dfa<dfa.decisionNumber>_max, /* Maximum tokens for each state */
    dfa<dfa.decisionNumber>_accept, /* Accept table */
    dfa<dfa.decisionNumber>_special, /* Special transition states */
    dfa<dfa.decisionNumber>_transitions /* Table of transition tables */

);

/* End of Cyclic DFA <dfa.decisionNumber>
* -----
*/

```

```

>>

/** A state in a cyclic DFA; it's a special state and part of a big switch on
 * state.
 */
cyclicDFAState(decisionNumber,stateNumber,edges,needErrorClause,semPredState) ::= <<
{
  ANTLR_UINT32 LA<decisionNumber>_<stateNumber>;<\n>
  ANTLR_MARKER index<decisionNumber>_<stateNumber>;<\n>

  LA<decisionNumber>_<stateNumber> = ctx->LA(1);<\n>
  <if(semPredState)> <! get next lookahead symbol to test edges, then rewind !>
  index<decisionNumber>_<stateNumber> = ctx->index();<\n>
  ctx->rewindLast();<\n>
  <endif>
  s = -1;
  <edges; separator="\nelse ">
  <if(semPredState)> <! return input cursor to state before we rewound !>
  ctx->seek(index<decisionNumber>_<stateNumber>);<\n>
  <endif>
  if ( s>=0 )
  {
  return s;
  }
}
break;
>>

/** Just like a fixed DFA edge, test the lookahead and indicate what
 * state to jump to next if successful.
 */
cyclicDFAEdge(labelExpr, targetStateNumber, edgeNumber, predicates) ::= <<
if ( (<labelExpr> <if(predicates)>&& (<predicates>)<endif> )
{
  s = <targetStateNumber>;
}<\n>
>>

/** An edge pointing at end-of-token; essentially matches any char;
 * always jump to the target.
 */
eotDFAEdge(targetStateNumber,edgeNumber, predicates) ::= <<
s = <targetStateNumber>;<\n>
>>

// D F A E X P R E S S I O N S

```

```

andPredicates(left,right) ::= "( (<left> && (<right> ) )"

orPredicates(operands) ::= "<operands:{o|(<o>)}; separator=\"||\">"

notPredicate(pred) ::= "!(<evalPredicate(pred,{ })>)"

evalPredicate(pred,description) ::= "<pred>"

evalSynPredicate(pred,description) ::= "this->msynpred( antlr3::ClassForwarder\<<pred>>() )"

lookaheadTest(atom,k,atomAsInt) ::= "LA<decisionNumber>_<stateNumber> == <atom>"

/** Sometimes a lookahead test cannot assume that LA(k) is in a temp variable
 * somewhere. Must ask for the lookahead directly.
 */
isolatedLookaheadTest(atom,k,atomAsInt) ::= "this->LA(<k>) == <atom>"

lookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= <%
((LA<decisionNumber>_<stateNumber> >= <lower>) && (LA<decisionNumber>_<stateNumber> \<= <upper>))
%>

isolatedLookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= "((this->LA(<k>) >=
<lower>) && (this->LA(<k>) \<= <upper>))"

setTest(ranges) ::= "<ranges; separator=\" || \">"

// A T T R I B U T E S

makeScopeSet() ::= <<
/* makeScopeSet()
 */
/** Definition of the <scope.name> scope variable tracking
 * structure. An instance of this structure is created by calling
 * <name>_<scope.name>Push().
 */
struct <scopeStruct(sname=scope.name,...)>
{
/* =====
 * Programmer defined variables...
 */
<scope.attributes:{it |<it.decl>;}; separator="\n">

/* End of programmer defined variables
 * =====
 */
};

>>

```

```

globalAttributeScopeDecl(scope) ::= <<
<if(scope.attributes)>
/* globalAttributeScopeDecl(scope)
*/
<makeScopeSet(...)>
<endif>
>>

```

```

ruleAttributeScopeDecl(scope) ::= <<
<if(scope.attributes)>
/* ruleAttributeScopeDecl(scope)
*/
<makeScopeSet(...)>
<endif>
>>

```

```

globalAttributeScopeDef(scope) ::=
<<
/* globalAttributeScopeDef(scope)
*/
<if(scope.attributes)>

```

```

StackType\< <scopeStruct(sname=scope.name)> > <scopeStack(sname=scope.name)>;

```

```

<endif>
>>

```

```

ruleAttributeScopeDef(scope) ::= <<
<if(scope.attributes)>
/* ruleAttributeScopeDef(scope)
*/
StackType\< <scopeStruct(sname=scope.name)> > <scopeStack(sname=scope.name,...)>;

```

```

<endif>
>>

```

```

scopeStruct(sname) ::= <<
<sname>Scope
>>

```

```

scopeStack(sname) ::= <<
m_<sname>_stack
>>

```

```

returnStructName(r) ::= "<r.name>_return"

```

```

returnType() ::= <%

```

```

<if(!ruleDescriptor.isSynPred)>
<if(ruleDescriptor.hasMultipleReturnValues)>
<ruleDescriptor.grammar.recognizerName>::<ruleDescriptor:returnStructName()>
<else>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnType>
<else>
void
<endif>
<endif>
<else>
bool
<endif>
%>

```

```

/** Generate the C type associated with a single or multiple return
 * value(s).
 */

```

```

ruleLabelType(referencedRule) ::= <%
<if(referencedRule.hasMultipleReturnValues)>
<referencedRule.name>_return
<else>
<if(referencedRule.hasSingleReturnValue)>
<referencedRule.singleValueReturnType>
<else>
void
<endif>
<endif>
%>

```

```

delegateName(d) ::= <<
<if(d.label)><d.label><else>g<d.name><endif>
>>

```

```

/** Using a type to init value map, try to init a type; if not in table
 * must be an object, default value is "0".
 */

```

```

initValue(typeName) ::= <<
= <cTypeInitMap.(typeName)>
>>

```

```

/** Define a rule label */
ruleLabelDef(label) ::= <<
<ruleLabelType(referencedRule=label.referencedRule)> <label.label.text>;
>>

```

```

/** Rule label default value */
ruleLabelInitVal(label) ::= <<
>>

```

```

ASTLabelType() ::=
"<if(recognizer.ASTLabelType)><recognizer.ASTLabelType><else>ImplTraits::TreeType*<endif>"

/** Define a return struct for a rule if the code needs to access its
 * start/stop tokens, tree stuff, attributes, ... Leave a hole for
 * subgroups to stick in members.
 */
returnScope(scope) ::= <<
<if(!ruleDescriptor.isSynPred)>
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(!TREE_PARSER)>
struct <ruleDescriptor:returnStructName()> : public <name>ImplTraits::RuleReturnValueType
{
public:
    typedef <name>ImplTraits::RuleReturnValueType BaseType;
    <ruleDescriptor:returnStructName()>()
        : BaseType()
        <if(scope)>, <scope.attributes:{it | <it.name>() }; separator=","><endif>
        { init(); }
    <ruleDescriptor:returnStructName()>( BaseParserType* parser )
        : BaseType(parser)
        <if(scope)>, <scope.attributes:{it | <it.name>() }; separator=","><endif>
        { init(); }
    <ruleDescriptor:returnStructName()>( const <ruleDescriptor:returnStructName()>& other )
        : BaseType(other)
        <if(scope)>, <scope.attributes:{it | <it.name>(other.<it.name>) }; separator=","><endif>
        { copy(other); }
    ~<ruleDescriptor:returnStructName()>()
    {
        <@ruleReturnMembersDelete()>
    }

    <ruleDescriptor:returnStructName()>&
operator=( const <ruleDescriptor:returnStructName()>& other )
    {
        BaseType::operator=( other );
        <if(scope)><scope.attributes:{it | <it.name> = other.<it.name>; }; separator="\n"><endif>
        copy(other);
        return *this;
    }
    <@ruleReturnMembers()>
    void init() { <@ruleReturnMembersInit()> }
    void copy( const <ruleDescriptor:returnStructName()>& other ) { <@ruleReturnMembersCopy()> }
<else>
struct <ruleDescriptor:returnStructName()>
{
public:

```

```

<name>ImplTraits::<recognizer.ASTLabelType>    start;
<name>ImplTraits::<recognizer.ASTLabelType>    stop;
<ruleDescriptor:returnStructName()>( const <ruleDescriptor:returnStructName()>& other )
<if(scope.attributes)>
<scope.attributes:{it | <it.name>(other.<it.name> ) }; separator=",">
<endif>
{
    start = other.start;
    stop  = other.stop;
}

<ruleDescriptor:returnStructName()>&
operator=( const <ruleDescriptor:returnStructName()>& other )
{
    start = other.start;
    stop  = other.stop;

    <scope.attributes:{it | <it.name> = other.<it.name>; }; separator="\n">
    return *this;
}
<endif>
<if(scope)><scope.attributes:{it |<it.type> <it.name>; }; separator="\n"><endif>
};

<endif>
<endif>
>>

parameterScope(scope) ::= <<
<scope.attributes:{it |<it.decl>}; separator="," >>
>>

parameterAttributeRef(attr) ::= "<attr.name>"
parameterSetAttributeRef(attr,expr) ::= "<attr.name>=<expr>";

/** Note that the scopeAttributeRef does not have access to the
 * grammar name directly
 */
scopeAttributeRef(scope,attr,index,negIndex) ::= <%
<if(negIndex)>
    m_<scope>_stack.at( m_<scope>_stack.size()-<negIndex>-1).<attr.name>
<else>
<if(index)>
    m_<scope>_stack.at(<index>).<attr.name>
<else>
    m_<scope>_stack.peek().<attr.name>
<endif>
<endif>
<endif>

```

```

%>

scopeSetAttributeRef(scope,attr,expr,index,negIndex) ::= <%
<if(negIndex)>
  m_<scope>_stack.at( m_<scope>_stack.size()-<negIndex>-1).<attr.name> = <expr>;
<else>
<if(index)>
  m_<scope>_stack.at(<index>).<attr.name> = <expr>;
<else>
  m_<scope>_stack.peek().<attr.name> =<expr>;
<endif>
<endif>
%>

/** $x is either global scope or x is rule with dynamic scope; refers
 * to stack itself not top of stack. This is useful for predicates
 * like {$function.size()>0 && $function::name.equals("foo")}?
 */
isolatedDynamicScopeRef(scope) ::= "<scope>_stack"

/** reference an attribute of rule; might only have single return value */
ruleLabelRef(referencedRule,scope,attr) ::= <<
<if(referencedRule.hasMultipleReturnValues)>
<scope>.<attr.name>
<else>
<scope>
<endif>
>>

returnAttributeRef(ruleDescriptor,attr) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name>
<else>
<attr.name>
<endif>
>>

returnSetAttributeRef(ruleDescriptor,attr,expr) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name>=<expr>;
<else>
<attr.name>=<expr>;
<endif>
>>

/** How to translate $tokenLabel */
tokenLabelRef(label) ::= "<label>"

```

```

/** ids+=ID {$ids} or e+=expr {$e} */
listLabelRef(label) ::= "list_<label>"

// not sure the next are the right approach
//
tokenLabelPropertyRef_text(scope,attr) ::= "<scope>->getText()"
tokenLabelPropertyRef_type(scope,attr) ::= "<scope>->get_type()"
tokenLabelPropertyRef_line(scope,attr) ::= "<scope>->get_line()"
tokenLabelPropertyRef_pos(scope,attr) ::= "<scope>->get_charPositionInLine()"
tokenLabelPropertyRef_channel(scope,attr) ::= "<scope>->get_channel()"
tokenLabelPropertyRef_index(scope,attr) ::= "<scope>->get_tokenIndex()"
tokenLabelPropertyRef_tree(scope,attr) ::= "<scope>->get_tree()"
tokenLabelPropertyRef_int(scope,attr) ::= "<name>ImplTraits::ConvertToInt32(<scope>->getText())"

ruleLabelPropertyRef_start(scope,attr) ::= "<scope>.start"
ruleLabelPropertyRef_stop(scope,attr) ::= "<scope>.stop"
ruleLabelPropertyRef_tree(scope,attr) ::= "<scope>.tree"
ruleLabelPropertyRef_text(scope,attr) ::= <<
<if(TREE_PARSER)>
(this->get_strstream()->toStringSS(<scope>.start, <scope>.start))
<else>
(this->get_strstream()->toStringTT(<scope>.start, <scope>.stop))
<endif>
>>

ruleLabelPropertyRef_st(scope,attr) ::= "<scope>.st"

/** Isolated $RULE ref ok in lexer as it's a Token */
lexerRuleLabel(label) ::= "<label>"

lexerRuleLabelPropertyRef_type(scope,attr) ::= "<scope>->get_type()"
lexerRuleLabelPropertyRef_line(scope,attr) ::= "<scope>->get_line()"
lexerRuleLabelPropertyRef_pos(scope,attr) ::= "<scope>->get_charPositionInLine()"
lexerRuleLabelPropertyRef_channel(scope,attr) ::= "<scope>->get_channel()"
lexerRuleLabelPropertyRef_index(scope,attr) ::= "<scope>->get_tokenIndex()"
lexerRuleLabelPropertyRef_text(scope,attr) ::= "<scope>->getText()"

// Somebody may ref $template or $tree or $stop within a rule:
rulePropertyRef_start(scope,attr) ::= "retval.start"
rulePropertyRef_stop(scope,attr) ::= "retval.stop"
rulePropertyRef_tree(scope,attr) ::= "retval.tree"
rulePropertyRef_text(scope,attr) ::= <<
<if(TREE_PARSER)>
this->get_input()->toStringSS( this->get_adaptor()->getTokenStartIndex(retval.start), this->get_adaptor()-
>getTokenStopIndex(retval.start))
<else>
this->get_strstream()->toStringTT(retval.start, this->LT(-1))

```

```

<endif>
>>
rulePropertyRef_st(scope,attr) ::= "retval.st"

lexerRulePropertyRef_text(scope,attr) ::= "this->getText()"
lexerRulePropertyRef_type(scope,attr) ::= "_type"
lexerRulePropertyRef_line(scope,attr) ::= "this->get_state()->get_tokenStartLine()"
lexerRulePropertyRef_pos(scope,attr) ::= "this->get_state()->get_tokenStartCharPositionInLine()"
lexerRulePropertyRef_channel(scope,attr) ::= "this->get_state()->get_channel()"
lexerRulePropertyRef_start(scope,attr) ::= "this->get_state()->get_tokenStartCharIndex()"
lexerRulePropertyRef_stop(scope,attr) ::= "(this->getCharIndex()-1)"
lexerRulePropertyRef_index(scope,attr) ::= "-1" // undefined token index in lexer
lexerRulePropertyRef_int(scope,attr) ::= "(<name>ImplTraits::ConvertToInt32(<scope>->getText()))"

// setting $st and $tree is allowed in local rule. everything else is flagged as error
ruleSetPropertyRef_tree(scope,attr,expr) ::= "retval.tree=<expr>";
ruleSetPropertyRef_st(scope,attr,expr) ::= "retval.st=<expr>";

/** How to deal with an @after for C targets. Because we cannot rely on
 * any garbage collection, after code is executed even in backtracking
 * mode. Must be documented clearly.
 */
execAfter(action) ::= <<
{
  <action>
}
>>

/** How to execute an action (when not backtracking) */
execAction(action) ::= <<
<if(backtracking)>
<if(actions.(actionScope).synpredgate)>
if ( <actions.(actionScope).synpredgate> )
{
  <action>
}
<else>
if ( BACKTRACKING == 0 )
{
  <action>
}
<endif>
<else>
{
  <action>
}

```

```

<endif>
>>

// M I S C (properties, etc...)

bitsetDeclare(bitsetname, words64, traits) ::= <<

/** Bitset defining follow set for error recovery in rule state: <name> */
static ANTLR_BITWORD <bitsetname>_bits[] = { <words64:{it |ANTLR_UINT64_LIT(<it>)}; separator=", "> };
static <traits>::BitsetListType <bitsetname>(<bitsetname>_bits, <length(words64)>);
>>

codeFileExtension() ::= ".cpp"

true_value() ::= "true"
false_value() ::= "false"

Found in path(s):
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/Cpp/Cpp.stg
No license file was found, but licenses were detected in source scan.

/*
[The "BSD license"]
Copyright (c) 2005-2006 Terence Parr
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

```

```

/** Template overrides to add debugging to AST stuff. Dynamic inheritance
 * hierarchy is set up as ASTDbg : AST : Dbg : Java by code generator.
 */

parserMembers() ::= <<
protected DebugTreeAdaptor adaptor;
public void setTreeAdaptor(TreeAdaptor adaptor) {
<if(grammar.grammarIsRoot)>
this.adaptor = new DebugTreeAdaptor(dbg,adaptor);
<else>
this.adaptor = (DebugTreeAdaptor)adaptor; // delegator sends dbg adaptor
<endif>
<grammar.directDelegates: { g|<g:delegateName()>.setTreeAdaptor(this.adaptor); }>
}
public TreeAdaptor getTreeAdaptor() {
return adaptor;
}
>>

parserCtorBody() ::= <<
<super.parserCtorBody()>
>>

createListenerAndHandshake() ::= <<
DebugEventSocketProxy proxy =
new DebugEventSocketProxy(this,port,<if(TREE_PARSER)>input.getTreeAdaptor()<else>adaptor<endif>);
setDebugListener(proxy);
set<inputStreamType>(new Debug<inputStreamType>(input,proxy));
try {
proxy.handshake();
}
catch (IOException ioe) {
reportError(ioe);
}
>>

@ctorForRootGrammar.finally() ::= <<
TreeAdaptor adap = new CommonTreeAdaptor();
setTreeAdaptor(adap);
proxy.setTreeAdaptor(adap);
>>

@ctorForProfilingRootGrammar.finally() ::= <<
TreeAdaptor adap = new CommonTreeAdaptor();
setTreeAdaptor(adap);
>>

```

```
@ctorForPredefinedListener.superClassRef() ::= "super(input, dbg);"
```

```
@ctorForPredefinedListener.finally() ::=<<<  
<if(grammar.grammarIsRoot)> <! don't create new adaptor for delegates !>  
TreeAdaptor adap = new CommonTreeAdaptor();  
setTreeAdaptor(adap);  
<endif>  
>>
```

```
@rewriteElement.pregen() ::= "dbg.location(<e.line>,<e.pos>);"
```

Found in path(s):

```
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-  
jar/org/antlr/codegen/templates/Java/ASTDbg.stg
```

No license file was found, but licenses were detected in source scan.

```
/*
```

```
[The "BSD license"]
```

```
Copyright (c) 2005-2009 Terence Parr
```

```
All rights reserved.
```

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

```
THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR  
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES  
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.  
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,  
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT  
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,  
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY  
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT  
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF  
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

```
*/
```

```
/** Template overrides to add debugging to AST stuff. Dynamic inheritance
```

```
* hierarchy is set up as ASTDbg : AST : Dbg : Python by code generator.
```

```
*/
```

```
group ASTDbg;
```

```

astAccessor() ::= <<
def setTreeAdaptor(self, adaptor):
<if(grammar.grammarIsRoot)>
    self._adaptor = DebugTreeAdaptor(self.dbg, adaptor)
<else>
    self._adaptor = adaptor # delegator sends dbg adaptor
<endif>
    <grammar.directDelegates:{g|<g.delegateName()>.setTreeAdaptor(self._adaptor)}>

def getTreeAdaptor(self):
    return self._adaptor

adaptor = property(getTreeAdaptor, setTreeAdaptor)<\n>
>>

createListenerAndHandshake() ::= <<
proxy = DebugEventSocketProxy(self,
adaptor=<if(TREE_PARSER)>self.input.getTreeAdaptor(<else>self._adaptor<endif>,
                debug=debug_socket, port=port)
self.setDebugListener(proxy)
self.adaptor.setDebugListener(proxy)
self.input.setDebugListener(proxy)
#self.set<inputStreamType>(Debug<inputStreamType>(self.input, proxy))
proxy.handshake()
>>

@rewriteElement.pregen() ::= "self._dbg.location(<e.line>, <e.pos>)"

```

Found in path(s):

* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/Python/ASTDbg.stg

No license file was found, but licenses were detected in source scan.

/*

[The "BSD license"]

Copyright (c) 2005-2006 Terence Parr

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products

derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

```
@outputFile.imports() ::= <<
<@super.imports()>
<if(!TREE_PARSER)><! tree parser would already have imported !>
import org.antlr.runtime.tree.*;<\n>
<endif>
>>
```

```
@genericParser.members() ::= <<
<@super.members()>
<parserMembers()>
>>
```

```
/** Add an adaptor property that knows how to build trees */
parserMembers() ::= <<
protected TreeAdaptor adaptor = new CommonTreeAdaptor();
```

```
public void setTreeAdaptor(TreeAdaptor adaptor) {
    this.adaptor = adaptor;
    <grammar.directDelegates: {g|<g:delegateName()>.setTreeAdaptor(this.adaptor);}>
}
public TreeAdaptor getTreeAdaptor() {
    return adaptor;
}
>>
```

```
@returnScope.ruleReturnMembers() ::= <<
<ASTLabelType> tree;
@Override
public <ASTLabelType> getTree() { return tree; }
>>
```

```
/** Add a variable to track rule's return AST */
ruleDeclarations() ::= <<
<super.ruleDeclarations()>
```

```

<ASTLabelType> root_0 = null;<\n>
>>

ruleLabelDefs() ::= <<
<super.ruleLabelDefs()>
<if(!ruleDescriptor.isSynPred)>
<[ruleDescriptor.tokenLabels,ruleDescriptor.wildcardTreeLabels,
ruleDescriptor.wildcardTreeListLabels]:{it | <ASTLabelType> <it.label.text>_tree=null;}; separator="\n">
<ruleDescriptor.tokenListLabels:{it | <ASTLabelType> <it.label.text>_tree=null;}; separator="\n">
<if(ruleDescriptor.supportsLabelOptimization)>
<ruleDescriptor.allTokenRefsInRewrites
:{it | RewriteRule<rewriteElementType>Stream stream_<it>=new
RewriteRule<rewriteElementType>Stream(adaptor,"token <it>");}; separator="\n">
<ruleDescriptor.allRuleRefsInRewrites
:{it | RewriteRuleSubtreeStream stream_<it>=new RewriteRuleSubtreeStream(adaptor,"rule <it>");};
separator="\n">
<else>
<ruleDescriptor.allTokenRefsInAltsWithRewrites
:{it | RewriteRule<rewriteElementType>Stream stream_<it>=new
RewriteRule<rewriteElementType>Stream(adaptor,"token <it>");}; separator="\n">
<ruleDescriptor.allRuleRefsInAltsWithRewrites
:{it | RewriteRuleSubtreeStream stream_<it>=new RewriteRuleSubtreeStream(adaptor,"rule <it>");};
separator="\n">
<endif>
<endif>
>>

/** When doing auto AST construction, we must define some variables;
 * These should be turned off if doing rewrites. This must be a "mode"
 * as a rule could have both rewrite and AST within the same alternative
 * block.
 */
@alt.declarations() ::= <<
<if(autoAST)>
<if(outerAlt)>
<if(!rewriteMode && !ruleDescriptor.isSynPred)>
root_0 = (<ASTLabelType>)adaptor.nil();<\n>
<endif>
<endif>
<endif>
>>

// Tracking Rule Elements

/** ID and track it for use in a rewrite rule */
tokenRefTrack(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRefBang(...)> <! Track implies no auto AST construction!>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>stream_<token>.add(<label>);<\n>

```

```

>>

/** ids+=ID and track it for use in a rewrite rule; adds to ids *and*
 * to the tracking list stream_ID for use in the rewrite.
 */
tokenRefTrackAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRefTrack(...)>
<listLabel(elem=label, ...)>
>>

/** ^(ID ...) track for rewrite */
tokenRefRuleRootTrack(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRefBang(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>stream_<token>.add(<label>);<\n>
>>

/** Match ^(label+=TOKEN ...) track for rewrite */
tokenRefRuleRootTrackAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRefRuleRootTrack(...)>
<listLabel(elem=label, ...)>
>>

/** rule when output=AST and tracking for rewrite */
ruleRefTrack(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>stream_<rule.name>.add(<label>.getTree());
>>

/** x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefTrack(...)>
<listLabel(label, {<label>.getTree()})>
>>

/** ^(rule ...) rewrite */
ruleRefRuleRootTrack(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>stream_<rule>.add(<label>.getTree());
>>

/** ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRootTrack(...)>
<listLabel(label, {<label>.getTree()})>
>>

// R e w r i t e

```

```

rewriteCode(
  alts, description,
  referencedElementsDeep, // ALL referenced elements to right of ->
  referencedTokenLabels,
  referencedTokenListLabels,
  referencedRuleLabels,
  referencedRuleListLabels,
  referencedWildcardLabels,
  referencedWildcardListLabels,
  rewriteBlockLevel, enclosingTreeLevel, treeLevel) ::=
<<

// AST REWRITE
// elements: <referencedElementsDeep; separator=", ">
// token labels: <referencedTokenLabels; separator=", ">
// rule labels: <referencedRuleLabels; separator=", ">
// token list labels: <referencedTokenListLabels; separator=", ">
// rule list labels: <referencedRuleListLabels; separator=", ">
// wildcard labels: <[referencedWildcardLabels, referencedWildcardListLabels]; separator=", ">
<if(backtracking)>
if ( <actions.(actionScope).synpredgate> ) {
<endif>
<prevRuleRootRef(>).tree = root_0;
<rewriteCodeLabels(>
root_0 = (<ASTLabelType>)adaptor.nil();
<alts:rewriteAlt(> separator="else ">
<! if tree parser and rewrite=true !>
<if(TREE_PARSER)>
<if(rewriteMode)>
<prevRuleRootRef(>).tree = (<ASTLabelType>)adaptor.rulePostProcessing(root_0);
input.replaceChildren(adaptor.getParent(retval.start),
  adaptor.getChildIndex(retval.start),
  adaptor.getChildIndex(_last),
  retval.tree);
<endif>
<endif>
<! if parser or tree-parser && rewrite!=true, we need to set result !>
<if(!TREE_PARSER)>
<prevRuleRootRef(>).tree = root_0;
<else>
<if(!rewriteMode)>
<prevRuleRootRef(>).tree = root_0;
<endif>
<endif>
<endif>
<if(backtracking)>
}
<endif>
>>

```

```

rewriteCodeLabels() ::= <<
<referencedTokenLabels
: {it | RewriteRule<rewriteElementType>Stream stream_<it>=new
RewriteRule<rewriteElementType>Stream(adaptor,"token <it>",<it>)};
separator="\n"
>
<referencedTokenListLabels
: {it | RewriteRule<rewriteElementType>Stream stream_<it>=new
RewriteRule<rewriteElementType>Stream(adaptor,"token <it> ", list_<it>)};
separator="\n"
>
<referencedWildcardLabels
: {it | RewriteRuleSubtreeStream stream_<it>=new RewriteRuleSubtreeStream(adaptor,"wildcard <it>",<it>)};
separator="\n"
>
<referencedWildcardListLabels
: {it | RewriteRuleSubtreeStream stream_<it>=new RewriteRuleSubtreeStream(adaptor,"wildcard <it> ",list_<it>)};
separator="\n"
>
<referencedRuleLabels
: {it | RewriteRuleSubtreeStream stream_<it>=new RewriteRuleSubtreeStream(adaptor,"rule
<it>",<it>!=null?<it>.getTree():null)};
separator="\n"
>
<referencedRuleListLabels
: {it | RewriteRuleSubtreeStream stream_<it>=new RewriteRuleSubtreeStream(adaptor,"token <it> ",list_<it>)};
separator="\n"
>
>>

/** Generate code for an optional rewrite block; note it uses the deep ref'd element
 * list rather shallow like other blocks.
 */
rewriteOptionalBlock(
alt,rewriteBlockLevel,
referencedElementsDeep, // all nested refs
referencedElements, // elements in immediately block; no nested blocks
description) ::=
<<
// <fileName>:<description>
if ( <referencedElementsDeep:{el | stream_<el>.hasNext()} ; separator="||"> ) {
<alt>
}
<referencedElementsDeep:{el | stream_<el>.reset();<\n>}>
>>

rewriteClosureBlock(

```

```

alt,rewriteBlockLevel,
referencedElementsDeep, // all nested refs
referencedElements, // elements in immediately block; no nested blocks
description) ::=
<<
// <fileName>:<description>
while ( <referencedElements:{el | stream_<el>.hasNext()}; separator="||"> ) {
  <alt>
}
<referencedElements:{el | stream_<el>.reset();<\n>}>
>>

```

```

rewritePositiveClosureBlock(
alt,rewriteBlockLevel,
referencedElementsDeep, // all nested refs
referencedElements, // elements in immediately block; no nested blocks
description) ::=
<<
if ( !(<referencedElements:{el | stream_<el>.hasNext()}; separator="||">) ) {
  throw new RewriteEarlyExitException();
}
while ( <referencedElements:{el | stream_<el>.hasNext()}; separator="||"> ) {
  <alt>
}
<referencedElements:{el | stream_<el>.reset();<\n>}>
>>

```

```

rewriteAlt(a) ::= <<
// <a.description>
<if(a.pred)>
if (<a.pred>) {
  <a.alt>
}<\n>
<else>
{
  <a.alt>
}<\n>
<endif>
>>

```

```

/** For empty rewrites: "r: ... -> ;" */
rewriteEmptyAlt() ::= "root_0 = null;"

```

```

rewriteTree(root,children,description,enclosingTreeLevel,treeLevel) ::= <<
// <fileName>:<description>
{
  <ASTLabelType> root_<treeLevel> = (<ASTLabelType>)adaptor.nil();
  <root:rewriteElement()>
}

```

```

<children:rewriteElement()>
adaptor.addChild(root_<enclosingTreeLevel>, root_<treeLevel>);
}<\n>
>>

rewriteElementList(elements) ::= "<elements:rewriteElement()>"

rewriteElement(e) ::= <<
<@pregen()>
<e.el>
>>

/** Gen ID or ID[args] */
rewriteTokenRef(token,elementIndex,args,terminalOptions={}) ::= <<
adaptor.addChild(root_<treeLevel>, <createRewriteNodeFromElement(...)>);
>>

/** Gen $label ... where defined via label=ID */
rewriteTokenLabelRef(label,elementIndex) ::= <<
adaptor.addChild(root_<treeLevel>, stream_<label>.nextNode());
>>

/** Gen $label ... where defined via label+=ID */
rewriteTokenListLabelRef(label,elementIndex) ::= <<
adaptor.addChild(root_<treeLevel>, stream_<label>.nextNode());
>>

/** Gen ^($label ...) */
rewriteTokenLabelRefRoot(label,elementIndex) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.becomeRoot(stream_<label>.nextNode(), root_<treeLevel>);
>>

/** Gen ^($label ...) where label+=... */
rewriteTokenListLabelRefRoot ::= rewriteTokenLabelRefRoot

/** Gen ^(ID ...) or ^(ID[args] ...) */
rewriteTokenRefRoot(token,elementIndex,args,terminalOptions={}) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.becomeRoot(<createRewriteNodeFromElement(...)>,
root_<treeLevel>);
>>

rewriteImaginaryTokenRef(args,token,elementIndex,terminalOptions={}) ::= <<
adaptor.addChild(root_<treeLevel>, <createImaginaryNode(token,args,terminalOptions)>);
>>

rewriteImaginaryTokenRefRoot(args,token,elementIndex,terminalOptions={}) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.becomeRoot(<createImaginaryNode(token,args,terminalOptions)>,
root_<treeLevel>);

```

```

>>

/** plain -> {foo} action */
rewriteAction(action) ::= <<
root_0 = <action>;
>>

/** What is the name of the previous value of this rule's root tree? This
 * let's us refer to $rule to mean previous value. I am reusing the
 * variable 'tree' sitting in retval struct to hold the value of root_0 right
 * before I set it during rewrites. The assign will be to retval.tree.
 */
prevRuleRootRef() ::= "retval"

rewriteRuleRef(rule) ::= <<
adaptor.addChild(root_<treeLevel>, stream_<rule>.nextTree());
>>

rewriteRuleRefRoot(rule) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.becomeRoot(stream_<rule>.nextNode(), root_<treeLevel>);
>>

rewriteNodeAction(action) ::= <<
adaptor.addChild(root_<treeLevel>, <action>);
>>

rewriteNodeActionRoot(action) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.becomeRoot(<action>, root_<treeLevel>);
>>

/** Gen $ruleLabel ... where defined via ruleLabel=rule */
rewriteRuleLabelRef(label) ::= <<
adaptor.addChild(root_<treeLevel>, stream_<label>.nextTree());
>>

/** Gen $ruleLabel ... where defined via ruleLabel+=rule */
rewriteRuleListLabelRef(label) ::= <<
adaptor.addChild(root_<treeLevel>, stream_<label>.nextTree());
>>

/** Gen ^($ruleLabel ...) where ruleLabel=rule */
rewriteRuleLabelRefRoot(label) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.becomeRoot(stream_<label>.nextNode(), root_<treeLevel>);
>>

/** Gen ^($ruleLabel ...) where ruleLabel+=rule */
rewriteRuleListLabelRefRoot(label) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.becomeRoot(stream_<label>.nextNode(), root_<treeLevel>);

```

>>

```
rewriteWildcardLabelRef(label) ::= <<
adaptor.addChild(root_<treeLevel>, stream_<label>.nextTree());
>>
```

```
createImaginaryNode(tokenType,args,terminalOptions={}) ::= <%
<if(terminalOptions.node)>
<! new MethodNode(IDLabel, args) !>
new <terminalOptions.node>(<tokenType><if(args)>, <args; separator=", "><endif>)
<else>
(<ASTLabelType>)adaptor.create(<tokenType>, <args; separator=", "><if(!args)>"<tokenType>"<endif>)
<endif>
%>
```

```
createRewriteNodeFromElement(token,args,terminalOptions={}) ::= <%
<if(terminalOptions.node)>
new <terminalOptions.node>(stream_<token>.nextToken(<if(args)>, <args; separator=", "><endif>)
<else>
<if(args)> <! must create new node from old !>
adaptor.create(<token>, <args; separator=", ">)
<else>
stream_<token>.nextNode()
<endif>
<endif>
%>
```

Found in path(s):

```
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/Java/AST.stg
No license file was found, but licenses were detected in source scan.
```

/*

[The "BSD license"]

Copyright (c) 2005-2012 Terence Parr

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```
*/
```

```
/** Template subgroup to add template rewrite output
```

```
* If debugging, then you'll also get STDbg.stg loaded.
```

```
*/
```

```
@outputFile.imports() ::= <<
```

```
<@super.imports(>
```

```
import stringtemplate3
```

```
>>
```

```
/** Add this to each rule's return value struct */
```

```
@returnScope.ruleReturnInit() ::= <<
```

```
self.st = None
```

```
>>
```

```
@returnScope.ruleReturnMembers() ::= <<
```

```
def getTemplate(self):
```

```
    return self.st
```

```
def toString(self):
```

```
    if self.st is not None:
```

```
        return self.st.toString()
```

```
    return None
```

```
__str__ = toString
```

```
>>
```

```
@genericParser.init() ::= <<
```

```
<@super.init(>
```

```
self.templateLib = stringtemplate3.StringTemplateGroup(
```

```
    '<name>Templates', lexer='angle-bracket'
```

```
    )
```

```
>>
```

```
@genericParser.members() ::= <<
```

```
<@super.members(>
```

```
def setTemplateLib(self, templateLib):
```

```

self.templateLib = templateLib

def getTemplateLib(self):
    return self.templateLib

>>

/** x+=rule when output=template */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(label, {<label>.st})>
>>

rewriteTemplate(alts) ::= <<
# TEMPLATE REWRITE
<if(backtracking)>
if <actions.(actionScope).synpredgate>:
    <first(alts):rewriteTemplateAltFirst()>
    <rest(alts):{it | el<rewriteTemplateAlt(it)>}>
    <if(rewriteMode)><replaceTextInLine()><endif>

<else>
<first(alts):rewriteTemplateAltFirst()>
<rest(alts):{it | el<rewriteTemplateAlt(it)>}>
<if(rewriteMode)><replaceTextInLine()><endif>
<endif>
>>

replaceTextInLine() ::= <<
<if(TREE_PARSER)>
self.input.getTokenStream().replace(
    self.input.getTreeAdaptor().getTokenStartIndex(retval.start),
    self.input.getTreeAdaptor().getTokenStopIndex(retval.start),
    retval.st
)
<else>
self.input.replace(
    retval.start.getTokenIndex(),
    self.input.LT(-1).getTokenIndex(),
    retval.st
)
<endif>
>>

rewriteTemplateAltFirst(alt) ::= <<
<if(alt.pred)>
if <alt.pred>:
    # <alt.description>

```

```

    retval.st = <alt.alt>
<\n>
<else>
# <alt.description>
retval.st = <alt.alt>
<\n>
<endif>
>>

```

```

rewriteTemplateAlt(alt) ::= <<
<if(alt.pred)>if <alt.pred>:
    # <alt.description>
    retval.st = <alt.alt>
<\n>
<else>se:
    # <alt.description>
    retval.st = <alt.alt>
<\n>
<endif>
>>

```

```

rewriteEmptyTemplate(alts) ::= <<
None
>>

```

/** Invoke a template with a set of attribute name/value pairs.

* Set the value of the rule's template *after* having set
* the attributes because the rule's template might be used as
* an attribute to build a bigger template; you get a self-embedded
* template.
*/

```

rewriteExternalTemplate(name,args) ::= <%
self.templateLib.getInstanceOf("<name><if(args)>, attributes={<args:{ a | "<a.name>": <a.value>}; separator=",
"><endif>
">}<endif>
%>

```

/** expr is a string expression that says what template to load */

```

rewriteIndirectTemplate(expr,args) ::= <%
self.templateLib.getInstanceOf(<expr><if(args)>, attributes={<args:{ a | "<a.name>": <a.value>}; separator=",
"><endif>
">}<endif>
%>

```

/** Invoke an inline template with a set of attribute name/value pairs */

```

rewriteInlineTemplate(args, template) ::= <%
stringtemplate3.StringTemplate("<template>", group=self.templateLib<if(args)>, attributes={<args:{ a |
"<a.name>": <a.value>}; separator=", "><endif>
%>

```

```

/** plain -> {foo} action */
rewriteAction(action) ::= <<
<action>
>>

/** An action has %st.attrName=expr; or % {st}.attrName=expr; */
actionSetAttribute(st,attrName,expr) ::= <<
(<st>)["<attrName>"] = <expr>
>>

/** Translate %{stringExpr} */
actionStringConstructor(stringExpr) ::= <<
stringtemplate3.StringTemplate(<stringExpr>, group=self.templateLib)
>>

```

Found in path(s):

```

* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-
jar/org/antlr/codegen/templates/Python3/ST.stg

```

No license file was found, but licenses were detected in source scan.

```

/*
[The "BSD license"]
Copyright (c) 2005-2006 Terence Parr
All rights reserved.

```

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```

*/
/** Template subgroup to add template rewrite output

```

```

* If debugging, then you'll also get STDbg.stg loaded.
*/

@outputFile.imports() ::= <<
<@super.imports(>
import org antlr.stringtemplate.*;
import org antlr.stringtemplate.language.*;
import java.util.HashMap;
>>

/** Add this to each rule's return value struct */
@returnScope.ruleReturnMembers() ::= <<
public StringTemplate st;
public Object getTemplate() { return st; }
public String toString() { return st==null?null:st.toString(); }
>>

@genericParser.members() ::= <<
<@super.members(>
protected StringTemplateGroup templateLib =
    new StringTemplateGroup("<name>Templates", AngleBracketTemplateLexer.class);

public void setTemplateLib(StringTemplateGroup templateLib) {
    this.templateLib = templateLib;
}
public StringTemplateGroup getTemplateLib() {
    return templateLib;
}
/** allows convenient multi-value initialization:
 * "new STAttrMap().put(...).put(...)"
 */
@SuppressWarnings("serial")
public static class STAttrMap extends HashMap<String, Object> {
    public STAttrMap put(String attrName, Object value) {
        super.put(attrName, value);
        return this;
    }
}
>>

/** x+=rule when output=template */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(rule,label,elementIndex,args,scope)>
<listLabel(label, {<label>.getTemplate()})>
>>

rewriteTemplate(alts) ::= <<

```

```

// TEMPLATE REWRITE
<if(backtracking)>
if ( <actions.(actionScope).synpredgate> ) {
  <alts:rewriteTemplateAlt(); separator="else ">
  <if(rewriteMode)><replaceTextInLine()><endif>
}
<else>
<alts:rewriteTemplateAlt(); separator="else ">
<if(rewriteMode)><replaceTextInLine()><endif>
<endif>
>>

replaceTextInLine() ::= <<
<if(TREE_PARSER)>
((TokenRewriteStream)input.getTokenStream()).replace(
  input.getTreeAdaptor().getTokenStartIndex(retval.start),
  input.getTreeAdaptor().getTokenStopIndex(retval.start),
  retval.st);
<else>
((TokenRewriteStream)input).replace(
  ((Token)retval.start).getTokenIndex(),
  input.LT(-1).getTokenIndex(),
  retval.st);
<endif>
>>

rewriteTemplateAlt(alt) ::= <<
// <alt.description>
<if(alt.pred)>
if (<alt.pred>) {
  retval.st = <alt.alt>;
}<\n>
<else>
{
  retval.st = <alt.alt>;
}<\n>
<endif>
>>

rewriteEmptyTemplate(alts) ::= <<
null;
>>

/** Invoke a template with a set of attribute name/value pairs.
 * Set the value of the rule's template *after* having set
 * the attributes because the rule's template might be used as
 * an attribute to build a bigger template; you get a self-embedded
 * template.

```

```

*/
rewriteExternalTemplate(name,args) ::= <%
templateLib.getInstanceOf("<name>"<if(args)>,
new STAttrMap()<args:{a | .put("<a.name>", <a.value>)}>
<endif>)
%>

/** expr is a string expression that says what template to load */
rewriteIndirectTemplate(expr,args) ::= <%
templateLib.getInstanceOf(<expr><if(args)>,
new STAttrMap()<args:{a | .put("<a.name>", <a.value>)}>
<endif>)
%>

/** Invoke an inline template with a set of attribute name/value pairs */
rewriteInlineTemplate(args, template) ::= <%
new StringTemplate(templateLib, "<template>"<if(args)>,
new STAttrMap()<args:{a | .put("<a.name>", <a.value>)}>
<endif>)
%>

/** plain -> {foo} action */
rewriteAction(action) ::= <<
<action>
>>

/** An action has %st.attrName=expr; or % {st}.attrName=expr; */
actionSetAttribute(st,attrName,expr) ::= <<
(<st>).setAttribute("<attrName>",<expr>);
>>

/** Translate % {stringExpr} */
actionStringConstructor(stringExpr) ::= <<
new StringTemplate(templateLib,<stringExpr>)
>>

```

Found in path(s):

* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/Java/ST.stg
No license file was found, but licenses were detected in source scan.

```

/*
[The "BSD license"]
Copyright (c) 2005-2006 Terence Parr
All rights reserved.

```

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

/** Templates for building ASTs during tree parsing.

*

* Deal with many combinations. Dimensions are:

* Auto build or rewrite

* no label, label, list label (label/no-label handled together)

* child, root

* token, set, rule, wildcard

*

* Each combination has its own template except that label/no label

* is combined into tokenRef, ruleRef, ...

*/

/** Add a variable to track last element matched */

```
ruleDeclarations() ::= <<
```

```
<super.ruleDeclarations()>
```

```
var _first_0:<ASTLabelType> = null;
```

```
var _last:<ASTLabelType> = null;<\n>
```

```
>>
```

/** What to emit when there is no rewrite rule. For auto build

* mode, does nothing.

*/

```
noRewrite(rewriteBlockLevel, treeLevel) ::= <<
```

```
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
```

```
<if(rewriteMode)>
```

```
retval.tree = <ASTLabelType>(_first_0);
```

```
if ( adaptor.getParent(retval.tree)!=null && adaptor.isNil( adaptor.getParent(retval.tree) ) )
```

```
    retval.tree = <ASTLabelType>(adaptor.getParent(retval.tree));
```

```

<endif>
<if(backtracking)>}<endif>
>>

/** match ^(root children) in tree parser; override here to
 * add tree construction actions.
 */
tree(root, actionsAfterRoot, children, nullableChildList,
      enclosingTreeLevel, treeLevel) ::= <<
_last = <ASTLabelType>(input.LT(1));
{
var _save_last_<treeLevel>:<ASTLabelType> = _last;
var _first_<treeLevel>:<ASTLabelType> = null;
<if(!rewriteMode)>
var root_<treeLevel>:<ASTLabelType> = <ASTLabelType>(adaptor.nil());
<endif>
<root:element()>
<if(rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> )<endif>
<if(root.el.rule)>
if ( _first_<enclosingTreeLevel>==null ) _first_<enclosingTreeLevel> = <root.el.label>.tree;
<else>
if ( _first_<enclosingTreeLevel>==null ) _first_<enclosingTreeLevel> = <root.el.label>;
<endif>
<endif>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if ( input.LA(1)==TokenConstants.DOWN ) {
  matchStream(input, TokenConstants.DOWN, null); <checkRuleBacktrackFailure()>
  <children:element()>
  matchStream(input, TokenConstants.UP, null); <checkRuleBacktrackFailure()>
}
<else>
matchStream(input, TokenConstants.DOWN, null); <checkRuleBacktrackFailure()>
<children:element()>
matchStream(input, TokenConstants.UP, null); <checkRuleBacktrackFailure()>
<endif>
<if(!rewriteMode)>
adaptor.addChild(root_<enclosingTreeLevel>, root_<treeLevel>);
<endif>
_last = _save_last_<treeLevel>;
}<\n>
>>

// TOKEN AST STUFF

/** ID! and output=AST (same as plain tokenRef) 'cept add
 * setting of _last

```

```

*/
tokenRefBang(token,label,elementIndex,terminalOptions) ::= <<
  _last = <ASTLabelType>(input.LT(1));
  <super.tokenRef(...)>
>>

/** ID auto construct */
tokenRef(token,label,elementIndex,terminalOptions) ::= <<
  _last = <ASTLabelType>(input.LT(1));
  <super.tokenRef(...)>
  <if(!rewriteMode)>
  <if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
  <if(terminalOptions.node)>
  <label>_tree = new <terminalOptions.node>( <label> );
  <else>
  <label>_tree = <ASTLabelType>(adaptor.dupNode(<label>));
  <endif><\n>
  adaptor.addChild(root_<treeLevel>, <label>_tree);
  <if(backtracking)>}<endif>
  <else> <! rewrite mode !>
  <if(backtracking)>if ( <actions.(actionScope).synpredgate> )<endif>
  if ( _first_<treeLevel>==null ) _first_<treeLevel> = <label>;
  <endif>
>>

/** label+=TOKEN auto construct */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
  <tokenRef(...)>
  <listLabel(elem=label,...)>
>>

/** ^(ID ...) auto construct */
tokenRefRuleRoot(token,label,elementIndex,terminalOptions) ::= <<
  _last = <ASTLabelType>(input.LT(1));
  <super.tokenRef(...)>
  <if(!rewriteMode)>
  <if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
  <if(terminalOptions.node)>
  <label>_tree = new <terminalOptions.node>( <label> );
  <else>
  <label>_tree = <ASTLabelType>(adaptor.dupNode(<label>));
  <endif><\n>
  root_<treeLevel> = <ASTLabelType>(adaptor.becomeRoot(<label>_tree, root_<treeLevel>));
  <if(backtracking)>}<endif>
  <endif>
>>

/** Match ^(label+=TOKEN ...) auto construct */

```

```

tokenRefRuleRootAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<tokenRefRuleRoot(...)>
<listLabel(elem=label,...)>
>>

/** Match . wildcard and auto dup the node/subtree */
wildcard(token,label,elementIndex,terminalOptions) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.wildcard(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<label>_tree = (<ASTLabelType>)adaptor.dupTree(<label>);
adaptor.addChild(root_<treeLevel>, <label>_tree);
<if(backtracking)>}<endif>
<else><! rewrite mode !>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> )<endif>
if ( _first_<treeLevel>==null ) _first_<treeLevel> = <label>;
<endif>
>>

// SET AST

matchSet(s,label,terminalOptions,elementIndex,postmatchCode) ::= <<
_last = <ASTLabelType>(input.LT(1));
<super.matchSet(postmatchCode={
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<if(terminalOptions.node)>
<label>_tree = new <terminalOptions.node>(<label>);
<else>
<label>_tree = <ASTLabelType>(adaptor.dupNode(<label>));
<endif><\n>
adaptor.addChild(root_<treeLevel>, <label>_tree);
<if(backtracking)>}\}<endif>
<endif>
}, ...
)>
>>

matchRuleBlockSet(s,label,terminalOptions,elementIndex,postmatchCode,treeLevel="0") ::= <<
<matchSet(...)>
<noRewrite(...)><! set return tree !>
>>

matchSetBang(s,label,terminalOptions,elementIndex,postmatchCode) ::= <<
_last = <ASTLabelType>(input.LT(1));
<super.matchSet(...)>
>>

```

```

matchSetRuleRoot(s,label,terminalOptions,elementIndex,debug) ::= <<
<super.matchSet(postmatchCode={
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<if(terminalOptions.node)>
<label>_tree = new <terminalOptions.node>(<label>);
<else>
<label>_tree = <ASTLabelType>(adaptor.dupNode(<label>));
<endif><\n>
root_<treeLevel> = <ASTLabelType>(adaptor.becomeRoot(<label>_tree, root_<treeLevel>));
<if(backtracking)>}<endif>
<endif>
}, ...
)>
>>

// RULE REF AST

/** rule auto construct */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
_last = <ASTLabelType>(input.LT(1));
<super.ruleRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>
<if(!rewriteMode)>
adaptor.addChild(root_<treeLevel>, <label>.tree);
<else> <! rewrite mode !>
if ( _first_<treeLevel>==null ) _first_<treeLevel> = <label>.tree;
<endif>
>>

/** x+=rule auto construct */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(label, {<label>.tree})>
>>

/** ^(rule ...) auto construct */
ruleRefRuleRoot(rule,label,elementIndex,args,scope) ::= <<
_last = <ASTLabelType>(input.LT(1));
<super.ruleRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>root_<treeLevel> =
<ASTLabelType>(adaptor.becomeRoot(<label>.tree, root_<treeLevel>));
<endif>
>>

/** ^(x+=rule ...) auto construct */

```

```

ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<listLabel(label, {<label>.tree})>
>>

/** rule when output=AST and tracking for rewrite */
ruleRefTrack(rule,label,elementIndex,args,scope) ::= <<
_last = <ASTLabelType>(input.LT(1));
<super.ruleRefTrack(...)>
>>

/** x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
_last = <ASTLabelType>(input.LT(1));
<super.ruleRefTrackAndListLabel(...)>
>>

/** ^(rule ...) rewrite */
ruleRefRuleRootTrack(rule,label,elementIndex,args,scope) ::= <<
_last = <ASTLabelType>(input.LT(1));
<super.ruleRefRootTrack(...)>
>>

/** ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
_last = <ASTLabelType>(input.LT(1));
<super.ruleRefRuleRootTrackAndListLabel(...)>
>>

/** Streams for token refs are tree nodes now; override to
* change nextToken to nextNode.
*/
createRewriteNodeFromElement(token,terminalOptions,scope) ::= <<
<if(terminalOptions.node)>
new <terminalOptions.node>(stream_<token>.nextNode())
<else>
stream_<token>.nextNode()
<endif>
>>

ruleCleanup() ::= <<
<super.ruleCleanup()>
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<\n><endif>
retval.tree = <ASTLabelType>(adaptor.rulePostProcessing(root_0));
<if(backtracking)>}<endif>
<endif>
>>

```

Found in path(s):

* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/ActionScript/ASTTreeParser.stg

No license file was found, but licenses were detected in source scan.

/*

[The "BSD license"]

Copyright (c) 2008 Erik van Bilsen

Copyright (c) 2007-2008 Johannes Luber

Copyright (c) 2005-2007 Kunle Odutola

Copyright (c) 2005-2006 Terence Parr

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

/** Templates for building ASTs during tree parsing.

*

* Deal with many combinations. Dimensions are:

* Auto build or rewrite

* no label, label, list label (label/no-label handled together)

* child, root

* token, set, rule, wildcard

*

* Each combination has its own template except that label/no label

* is combined into tokenRef, ruleRef, ...

*/

```

group ASTTreeParser;

/** Add a variable to track last element matched */
ruleDeclarations() ::= <<
<super.ruleDeclarations()>
_First[0] := nil;
_Last := nil;<\n>
>>

ruleDeclarationVars() ::= <<
<super.ruleDeclarationVars()>
_First, _Save_Last: array [0..63] of I<ASTLabelType>;
_Last: I<ASTLabelType>;
>>

/** What to emit when there is no rewrite rule. For auto build
* mode, does nothing.
*/
noRewrite(rewriteBlockLevel, treeLevel) ::= <<
<if(backtracking)>if (State.Backtracking = 0) then
begin<endif>
<if(rewriteMode)>
RetVal.Tree := _First[0] as I<ASTLabelType>;
if (Adaptor.GetParent(RetVal.Tree) \<\> nil) and (Adaptor.IsNil(Adaptor.GetParent(RetVal.Tree))) then
RetVal.Tree := Adaptor.GetParent(RetVal.Tree) as I<ASTLabelType>;
<endif>
<if(backtracking)>end;<endif>
>>

/** match ^(root children) in tree parser; override here to
* add tree construction actions.
*/
tree(root, actionsAfterRoot, children, nullableChildList,
enclosingTreeLevel, treeLevel) ::= <<
_Last := Input.LT(1) as I<ASTLabelType>;
begin
_Save_Last[<treeLevel>] := _Last;
_First[<treeLevel>] := nil;
<if(!rewriteMode)>
Root[<treeLevel>] := Adaptor.GetNilNode as I<ASTLabelType>;<\n>
<endif>
<root:element()>
<if(rewriteMode)>
<if(backtracking)>if (State.Backtracking = 0) then <endif>
<if(root.el.rule)>
if (_First[<enclosingTreeLevel>] = nil) then _First[<enclosingTreeLevel>] := <root.el.label>.Tree;
<else>
if (_First[<enclosingTreeLevel>] = nil) then _First[<enclosingTreeLevel>] := <root.el.label>;

```

```

<endif>
<endif>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if (Input.LA(1) = TToken.DOWN) then
begin
  Match(Input, TToken.DOWN, nil); <checkRuleBacktrackFailure()>
  <children:element()>
  Match(Input, TToken.UP, nil); <checkRuleBacktrackFailure()>
end;
<else>
Match(Input, TToken.DOWN, nil); <checkRuleBacktrackFailure()>
<children:element()>
Match(Input, TToken.UP, nil); <checkRuleBacktrackFailure()>
<endif>
<if(!rewriteMode)>
  Adaptor.AddChild(Root[<enclosingTreeLevel>], Root[<treeLevel>]);
<endif>
  _Last := _Save_Last[<treeLevel>];
end;<\n>
>>

// TOKEN AST STUFF

/** ID! and output=AST (same as plain tokenRef) 'cept add
 * setting of _last
 */
tokenRefBang(token,label,elementIndex) ::= <<
  _Last := Input.LT(1) as I<ASTLabelType>;
  <super.tokenRef(...)>
>>

/** ID auto construct */
tokenRef(token,label,elementIndex,terminalOptions) ::= <<
  _Last := Input.LT(1) as I<ASTLabelType>;
  <super.tokenRef(...)>
  <if(!rewriteMode)>
  <if(backtracking)>
  if (State.Backtracking = 0) then
  begin<\n>
  <endif>
  <if(terminalOptions.node)>
  <label>_tree := T<terminalOptions.node>.Create(<label>);
  <else>
  <label>_tree := Adaptor.DupNode(<label>) as I<ASTLabelType>;
  <endif><\n>
  Adaptor.AddChild(Root[<treeLevel>], <label>_tree);
  <if(backtracking)>

```

```

end;
<endif>
<else> <! rewrite mode !>
<if(backtracking)>if (State.Backtracking = 0) then <endif>
if (_First[<treeLevel>] = nil) then _First[<treeLevel>] := <label>;
<endif>
>>

/** label+=TOKEN auto construct */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

/** ^(ID ...) auto construct */
tokenRefRuleRoot(token,label,elementIndex,terminalOptions) ::= <<
_Last := Input.LT(1) as I<ASTLabelType>;
<super.tokenRef(...)>
<if(!rewriteMode)>
<if(backtracking)>
if (State.Backtracking = 0) then
begin
<endif>
<if(terminalOptions.node)>
<label>_tree := T<terminalOptions.node>.Create(<label>);
<else>
<label>_tree := Adaptor.DupNode(<label>) as I<ASTLabelType>;
<endif><\n>
Root[<treeLevel>] := Adaptor.BecomeRoot(<label>_tree, Root[<treeLevel>]) as I<ASTLabelType>;
<if(backtracking)>
end;
<endif>
<endif>
>>

/** Match ^(label+=TOKEN ...) auto construct */
tokenRefRuleRootAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<tokenRefRuleRoot(...)>
<listLabel(elem=label,...)>
>>

// SET AST

matchSet(s,label,terminalOptions,elementIndex,postmatchCode) ::= <<
_Last := Input.LT(1) as I<ASTLabelType>;
<super.matchSet(..., postmatchCode={
<if(!rewriteMode)>
<if(backtracking)>if (State.Backtracking = 0) then begin <endif>

```

```

<if(terminalOptions.node)>
<label>_tree := T<terminalOptions.node>.Create(<label>);
<else>
<label>_tree := Adaptor.DupNode(<label>) as I<ASTLabelType>;
<endif><\n>
Adaptor.AddChild(Root[<treeLevel>], <label>_tree);
<if(backtracking)>end;<endif>
<endif>
}
)>
>>

matchRuleBlockSet(s,label,terminalOptions,elementIndex,postmatchCode,treeLevel="0") ::= <<
<matchSet(...)>
<noRewrite()> <! set return tree !>
>>

matchSetBang(s,label,elementIndex,postmatchCode) ::= <<
_Last := Input.LT(1) as I<ASTLabelType>;
<super.matchSet(...)>
>>

matchSetRuleRoot(s,label,terminalOptions,elementIndex,debug) ::= <<
<super.matchSet(..., postmatchCode={
<rewriteMode>
<if(backtracking)>if (State.Backtracking = 0) then begin <endif>
<if(terminalOptions.node)>
<label>_tree := T<terminalOptions.node>.Create(<label>);
<else>
<label>_tree := Adaptor.DupNode(<label>) as I<ASTLabelType>;
<endif><\n>
Root[<treeLevel>] := Adaptor.BecomeRoot(<label>_tree, Root[<treeLevel>]) as I<ASTLabelType>;
<if(backtracking)>end;<endif>
<endif>
}
)>
>>

// RULE REF AST

/** rule auto construct */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
_Last := Input.LT(1) as I<ASTLabelType>;
<super.ruleRef(...)>
<if(backtracking)>if (State.Backtracking = 0) then <endif>
<if(!rewriteMode)>
Adaptor.AddChild(Root[<treeLevel>], <label>.Tree);
<else> <! rewrite mode !>

```

```

if (_First[<treeLevel>] = nil) then _First[<treeLevel>] := <label>.Tree;
<endif>
>>

/** x+=rule auto construct */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(elem=label+".Tree",...)>
>>

/** ^(rule ...) auto construct */
ruleRefRuleRoot(rule,label,elementIndex,args,scope) ::= <<
_Last := Input.LT(1) as I<ASTLabelType>;
<super.ruleRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if (State.Backtracking = 0) then <endif>Root[<treeLevel>] :=
Adaptor.BecomeRoot(<label>.Tree, Root[<treeLevel>]) as I<ASTLabelType>;
<endif>
>>

/** ^(x+=rule ...) auto construct */
ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<listLabel(elem=label+".Tree",...)>
>>

/** rule when output=AST and tracking for rewrite */
ruleRefTrack(rule,label,elementIndex,args,scope) ::= <<
_Last := Input.LT(1) as I<ASTLabelType>;
<super.ruleRefTrack(...)>
>>

/** x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
_Last := Input.LT(1) as I<ASTLabelType>;
<super.ruleRefTrackAndListLabel(...)>
>>

/** ^(rule ...) rewrite */
ruleRefRuleRootTrack(rule,label,elementIndex,args,scope) ::= <<
_Last := Input.LT(1) as I<ASTLabelType>;
<super.ruleRefRootTrack(...)>
>>

/** ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
_Last := Input.LT(1) as I<ASTLabelType>;
<super.ruleRefRuleRootTrackAndListLabel(...)>

```

>>

```
/** Streams for token refs are tree nodes now; override to
 * change nextToken to nextNode.
 */
createRewriteNodeFromElement(token,terminalOptions,scope) ::= <<
<if(terminalOptions.node)>
T<terminalOptions.node>.Create((Locals['Stream_<token>'] as IRewriteRuleElementStream).NextNode)
<else>
(Locals['Stream_<token>'] as IRewriteRuleElementStream).NextNode
<endif>
>>
```

```
ruleCleanUp() ::= <<
<super.ruleCleanUp()>
<if(!rewriteMode)>
<if(backtracking)>
if (State.Backtracking = 0) then
begin<\n>
<endif>
RefVal.Tree := Adaptor.RulePostProcessing(Root[0]) as I<ASTLabelType>;
<if(backtracking)>
end;
<endif>
<endif>
>>
```

Found in path(s):

```
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-
jar/org/antlr/codegen/templates/Delphi/ASTTreeParser.stg
No license file was found, but licenses were detected in source scan.
```

```
/*
 * [The "BSD license"]
 * Copyright (c) 2011 Terence Parr
 * All rights reserved.
 *
 * Conversion to C#:
 * Copyright (c) 2011 Sam Harwell, Pixel Mine, Inc.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
```

- * documentation and/or other materials provided with the distribution.
- * 3. The name of the author may not be used to endorse or promote products
- * derived from this software without specific prior written permission.
- *
- * THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
- * IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
- * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
- * IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
- * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
- * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
- * DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
- * THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
- * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
- * THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
- */

/** Templates for building ASTs during tree parsing.

- *
- * Deal with many combinations. Dimensions are:
- * Auto build or rewrite
- * no label, label, list label (label/no-label handled together)
- * child, root
- * token, set, rule, wildcard
- *
- * Each combination has its own template except that label/no label
- * is combined into tokenRef, ruleRef, ...
- */

/** Add a variable to track last element matched */

```
ruleDeclarations() ::= <<
<super.ruleDeclarations()>
<if(!ruleDescriptor.isSynPred)>
<ASTLabelType> _first_0 = default(<ASTLabelType>);
<ASTLabelType> _last = default(<ASTLabelType>);
<endif>
>>
```

/** What to emit when there is no rewrite rule. For auto build

- * mode, does nothing.
- */

```
noRewrite(rewriteBlockLevel=false, treeLevel=false) ::= <<
<if(!ruleDescriptor.isSynPred)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) {<endif>
<if(rewriteMode)>
retval.Tree = (<ASTLabelType>)_first_0;
if (adaptor.GetParent(retval.Tree)!=null && adaptor.IsNil(adaptor.GetParent(retval.Tree)))
retval.Tree = (<ASTLabelType>)adaptor.GetParent(retval.Tree);
<endif>
```

```

<if(backtracking)></endif>
<endif>
>>

/** match ^(root children) in tree parser; override here to
 * add tree construction actions.
 */
tree(root, actionsAfterRoot, children, nullableChildList,
      enclosingTreeLevel, treeLevel) ::= <<
<if(!ruleDescriptor.isSynPred)>
  _last = (<ASTLabelType>)input.LT(1);
  {
<ASTLabelType> _save_last_<treeLevel> = _last;
<ASTLabelType> _first_<treeLevel> = default(<ASTLabelType>);
<if(!rewriteMode)>
<ASTLabelType> root_<treeLevel> = (<ASTLabelType>)adaptor.Nil();
<endif>
<root:element()>
<if(rewriteMode)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>)<endif>
<if(root.el.rule)>
if (_first_<enclosingTreeLevel> == null) _first_<enclosingTreeLevel> = <root.el.label>.Tree;
<else>
if (_first_<enclosingTreeLevel> == null) _first_<enclosingTreeLevel> = <root.el.label>;
<endif>
<endif>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if (input.LA(1) == TokenTypes.Down) {
  Match(input, TokenTypes.Down, null); <checkRuleBacktrackFailure()>
  <children:element()>
  Match(input, TokenTypes.Up, null); <checkRuleBacktrackFailure()>
}
<else>
Match(input, TokenTypes.Down, null); <checkRuleBacktrackFailure()>
<children:element()>
Match(input, TokenTypes.Up, null); <checkRuleBacktrackFailure()>
<endif>
<if(!rewriteMode)>
adaptor.AddChild(root_<enclosingTreeLevel>, root_<treeLevel>);
<endif>
_last = _save_last_<treeLevel>;
}
<else>
<super.tree(...)>
<endif>
>>

```

```
// TOKEN AST STUFF
```

```
/** ID! and output=AST (same as plain tokenRef) 'cept add  
* setting of _last  
*/
```

```
tokenRefBang(token,label,elementIndex,terminalOptions) ::= <<  
<if(!ruleDescriptor.isSynPred)>  
_last = (<ASTLabelType>)input.LT(1);  
<super.tokenRef(...)>  
<else>  
<super.tokenRefBang(...)>  
<endif>  
>>
```

```
/** ID auto construct */
```

```
tokenRef(token,label,elementIndex,terminalOptions) ::= <<  
<if(!ruleDescriptor.isSynPred)>  
_last = (<ASTLabelType>)input.LT(1);  
<super.tokenRef(...)>  
<if(!rewriteMode)>  
<if(backtracking)>if (<actions.(actionScope).synpredgate>) {<endif>  
<if(terminalOptions.node)>  
<label>_tree = new  
<terminalOptions.node>(<if(terminalOptions.type)><terminalOptions.type>,<endif><label><if(terminalOptions.tex  
t)>,<terminalOptions.text; format="string"><endif>);  
<else>  
<label>_tree =  
(<ASTLabelType>)adaptor.DupNode(<if(terminalOptions.type)><terminalOptions.type>,<endif><label><if(termin  
alOptions.text)>,<terminalOptions.text; format="string"><endif>);  
<endif><\n>  
adaptor.AddChild(root_<treeLevel>, <label>_tree);  
<if(backtracking)>}<endif>  
<else> <! rewrite mode !>  
<if(backtracking)>if (<actions.(actionScope).synpredgate>)<endif>  
if (_first_<treeLevel> == null) _first_<treeLevel> = <label>;  
<endif>  
<else>  
<super.tokenRef(...)>  
<endif>  
>>
```

```
/** label+=TOKEN auto construct */
```

```
tokenRefAndListLabel(token,label,elementIndex,terminalOptions) ::= <<  
<if(!ruleDescriptor.isSynPred)>  
<tokenRef(...)>  
<listLabelElem(elem=label,...)>  
<else>  
<super.tokenRefAndListLabel(...)>
```

```

<endif>
>>

/** ^(ID ...) auto construct */
tokenRefRuleRoot(token,label,elementIndex,terminalOptions) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.tokenRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) {<endif>
<if(terminalOptions.node)>
<label>_tree = new
<terminalOptions.node>(<if(terminalOptions.type)><terminalOptions.type>,<endif><label><if(terminalOptions.tex
t)>,<terminalOptions.text; format="string"><endif>);
<else>
<label>_tree =
(<ASTLabelType>)adaptor.DupNode(<if(terminalOptions.type)><terminalOptions.type>,<endif><label><if(termin
alOptions.text)>,<terminalOptions.text; format="string"><endif>);
<endif><\n>
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(<label>_tree, root_<treeLevel>);
<if(backtracking)>}<endif>
<endif>
<else>
<super.tokenRefRuleRoot(...)>
<endif>
>>

/** Match ^(label+=TOKEN ...) auto construct */
tokenRefRuleRootAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<if(!ruleDescriptor.isSynPred)>
<tokenRefRuleRoot(...)>
<listLabelElem(elem=label,...)>
<else>
<super.tokenRefRuleRootAndListLabel(...)>
<endif>
>>

/** Match . wildcard and auto dup the node/subtree */
wildcard(token,label,elementIndex,terminalOptions) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.wildcard(...)>
<if(!rewriteMode)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) {<endif>
<label>_tree = (<ASTLabelType>)adaptor.DupTree(<label>);
adaptor.AddChild(root_<treeLevel>, <label>_tree);
<if(backtracking)>}<endif>
<else> <! rewrite mode !>

```

```

<if(backtracking)>if (<actions.(actionScope).synpredgate><endif>
if (_first_<treeLevel> == null) _first_<treeLevel> = <label>;
<endif>
<else>
<super.wildcard(...)>
<endif>
>>

// SET AST

matchSet(s,label,terminalOptions,elementIndex,postmatchCode) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.matchSet(postmatchCode={
<if(!rewriteMode)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) {<endif>
<if(terminalOptions.node)>
<label>_tree = new
<terminalOptions.node><(if(terminalOptions.type)><terminalOptions.type>,<endif><label><if(terminalOptions.tex
t)>,<terminalOptions.text; format="string"><endif>);
<else>
<label>_tree =
(<ASTLabelType>)adaptor.DupNode(<if(terminalOptions.type)><terminalOptions.type>,<endif><label><if(termin
alOptions.text)>,<terminalOptions.text; format="string"><endif>);
<endif><\n>
adaptor.AddChild(root_<treeLevel>, <label>_tree);
<if(backtracking)>}\}<endif>
<endif>
}, ...
)>
<else>
<super.matchSet(...)>
<endif>
>>

matchRuleBlockSet(s,label,terminalOptions,elementIndex,postmatchCode,treeLevel="0") ::= <<
<if(!ruleDescriptor.isSynPred)>
<matchSet(...)>
<noRewrite(...)> <! set return tree !>
<else>
<super.matchRuleBlockSet(...)>
<endif>
>>

matchSetBang(s,label,terminalOptions,elementIndex,postmatchCode) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.matchSet(...)>

```

```

<else>
<super.matchSetBang(...)>
<endif>
>>

matchSetRuleRoot(s,label,terminalOptions,elementIndex,debug) ::= <<
<if(!ruleDescriptor.isSynPred)>
<super.matchSet(postmatchCode={
<if(!rewriteMode)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) {<endif>
<if(terminalOptions.node)>
<label>_tree = new
<terminalOptions.node>(<if(terminalOptions.type)><terminalOptions.type>,<endif><label><if(terminalOptions.tex
t)>,<terminalOptions.text; format="string"><endif>);
<else>
<label>_tree =
(<ASTLabelType>)adaptor.DupNode(<if(terminalOptions.type)><terminalOptions.type>,<endif><label><if(termin
alOptions.text)>,<terminalOptions.text; format="string"><endif>);
<endif><\n>
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(<label>_tree, root_<treeLevel>);
<if(backtracking)>}\}<endif>
<endif>
}, ...
)>
<else>
<super.matchSetRuleRoot(...)>
<endif>
>>

// RULE REF AST

/** rule auto construct */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRef(...)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>)<endif>
<if(!rewriteMode)>
adaptor.AddChild(root_<treeLevel>, <label>.Tree);
<else> <! rewrite mode !>
if (_first_<treeLevel> == null) _first_<treeLevel> = <label>.Tree;
<endif>
<else>
<super.ruleRef(...)>
<endif>
>>

/** x+=rule auto construct */

```

```

ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<if(!ruleDescriptor.isSynPred)>
<ruleRef(...)>
<listLabelElem(elem={<label>.Tree},...)>
<else>
<super.ruleRefAndListLabel(...)>
<endif>
>>

/** ^(rule ...) auto construct */
ruleRefRuleRoot(rule,label,elementIndex,args,scope) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) <endif>root_<treeLevel> =
(<ASTLabelType>)adaptor.BecomeRoot(<label>.Tree, root_<treeLevel>);
<endif>
<else>
<super.ruleRefRuleRoot(...)>
<endif>
>>

/** ^(x+=rule ...) auto construct */
ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<if(!ruleDescriptor.isSynPred)>
<ruleRefRuleRoot(...)>
<listLabelElem(elem={<label>.Tree},...)>
<else>
<super.ruleRefRuleRootAndListLabel(...)>
<endif>
>>

/** rule when output=AST and tracking for rewrite */
ruleRefTrack(rule,label,elementIndex,args,scope) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRefTrack(...)>
>>

/** x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRefTrackAndListLabel(...)>
<else>
<super.ruleRefTrackAndListLabel(...)>
<endif>
>>

```

```

/** ^(rule ...) rewrite */
ruleRefRuleRootTrack(rule,label,elementIndex,args,scope) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRefRuleRootTrack(...)>
<else>
<super.ruleRefRuleRootTrack(...)>
<endif>
>>

/** ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRefRuleRootTrackAndListLabel(...)>
<else>
<super.ruleRefRuleRootTrackAndListLabel(...)>
<endif>
>>

/** Streams for token refs are tree nodes now; override to
* change NextToken to NextNode.
*/
createRewriteNodeFromElement(token,terminalOptions,args) ::= <%
<if(terminalOptions.node)>
new
<terminalOptions.node><(if(terminalOptions.type)><terminalOptions.type>,<endif>stream_<token>.NextNode())
<else>
stream_<token>.NextNode()
<endif>
%>

ruleCleanup() ::= <<
<super.ruleCleanup()>
<if(!ruleDescriptor.isSynPred)>
<if(!rewriteMode)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) {<endif>
retval.Tree = (<ASTLabelType>)adaptor.RulePostProcessing(root_0);
<if(backtracking)>}<endif>
<endif>
<endif>
>>

Found in path(s):
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-
jar/org/antlr/codegen/templates/CSharp3/ASTTreeParser.stg

```

No license file was found, but licenses were detected in source scan.

/*

[The "BSD license"]

Copyright (c) 2005-2006 Terence Parr

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

/** Templates for building ASTs during tree parsing.

*

- * Deal with many combinations. Dimensions are:
- * Auto build or rewrite
- * no label, label, list label (label/no-label handled together)
- * child, root
- * token, set, rule, wildcard
- *
- * Each combination has its own template except that label/no label
- * is combined into tokenRef, ruleRef, ...

*/

```
finishedBacktracking(block) ::= <<
<if(backtracking)>
if <actions.(actionScope).synpredgate>:
  <block>
<else>
<block>
```

```

<endif>
>>

/** Add a variable to track last element matched */
ruleDeclarations() ::= <<
<super.ruleDeclarations()>
_first_0 = None
_last = None<\n>
>>

/** What to emit when there is no rewrite rule. For auto build
* mode, does nothing.
*/
noRewrite(rewriteBlockLevel, treeLevel) ::= <<
<finishedBacktracking({
<if(rewriteMode)>
retval.tree = _first_0
if self._adaptor.getParent(retval.tree) is not None and self._adaptor.isNil(self._adaptor.getParent(retval.tree)):
    retval.tree = self._adaptor.getParent(retval.tree)
<endif>
})>
>>

/** match ^(root children) in tree parser; override here to
* add tree construction actions.
*/
tree(root, actionsAfterRoot, children, nullableChildList,
    enclosingTreeLevel, treeLevel) ::= <<
_last = self.input.LT(1)
_save_last_<treeLevel> = _last
_first_<treeLevel> = None
<if(!rewriteMode)>
root_<treeLevel> = self._adaptor.nil()<\n>
<endif>
<root:element()>
<if(rewriteMode)>
<finishedBacktracking({
<if(root.el.rule)>
if _first_<enclosingTreeLevel> is None:
    _first_<enclosingTreeLevel> = <root.el.label>.tree<\n>
<else>
if _first_<enclosingTreeLevel> is None:
    _first_<enclosingTreeLevel> = <root.el.label><\n>
<endif>
})>
<endif>
<actionsAfterRoot:element()>
<if(nullableChildList)>

```

```

if self.input.LA(1) == DOWN:
    self.match(self.input, DOWN, None)
    <children:element()>
    self.match(self.input, UP, None)

<else>
self.match(self.input, DOWN, None)
<children:element()>
self.match(self.input, UP, None)<\n>
<endif>
<if(!rewriteMode)>
self._adaptor.addChild(root_<enclosingTreeLevel>, root_<treeLevel>)<\n>
<endif>
_last = _save_last_<treeLevel>

>>

// TOKEN AST STUFF

/** ID! and output=AST (same as plain tokenRef) 'cept add
 * setting of _last
 */
tokenRefBang(token,label,elementIndex,terminalOptions={ }) ::= <<
_last = self.input.LT(1)
<super.tokenRef(...)>
>>

/** ID auto construct */
tokenRef(token,label,elementIndex,terminalOptions={ }) ::= <<
_last = self.input.LT(1)
<super.tokenRef(...)>
<if(!rewriteMode)>
<finishedBacktracking({
<if(terminalOptions.node)>
<label>_tree = <terminalOptions.node>(<label>)
<else>
<label>_tree = self._adaptor.dupNode(<label>)
<endif><\n>
self._adaptor.addChild(root_<treeLevel>, <label>_tree)
})>
<else> <! rewrite mode !>
<finishedBacktracking({
if _first_<treeLevel> is None:
    _first_<treeLevel> = <label><\n>
})>
<endif>
>>

```

```

/** label+=TOKEN auto construct */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

/** ^(ID ...) auto construct */
tokenRefRuleRoot(token,label,elementIndex,terminalOptions={}) ::= <<
_last = self.input.LT(1)
<super.tokenRef(...)>
<if(!rewriteMode)>
<finishedBacktracking({
<if(terminalOptions.node)>
<label>_tree = <terminalOptions.node><(<label>)
<else>
<label>_tree = self._adaptor.dupNode(<label>)
<endif><\n>
root_<treeLevel> = self._adaptor.becomeRoot(<label>_tree, root_<treeLevel>)
})>
<endif>
>>

/** Match ^(label+=TOKEN ...) auto construct */
tokenRefRuleRootAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRefRuleRoot(...)>
<listLabel(elem=label,...)>
>>

/** Match . wildcard and auto dup the node/subtree */
wildcard(token,label,elementIndex,terminalOptions={}) ::= <<
_last = self.input.LT(1)
<super.wildcard(...)>
<if(!rewriteMode)>
<finishedBacktracking({
<label>_tree = self._adaptor.dupTree(<label>)
self._adaptor.addChild(root_<treeLevel>, <label>_tree)
})>
<else> <! rewrite mode !>
<finishedBacktracking({
if _first_<treeLevel> is None:
_first_<treeLevel> = <label>
})>
<endif>
>>

// SET AST
matchSet(s,label,elementIndex,postmatchCode,terminalOptions={}) ::= <<
_last = self.input.LT(1)

```

```

<super.matchSet(postmatchCode={
<if(!rewriteMode)>
<finishedBacktracking({
<if(terminalOptions.node)>
<label>_tree = <terminalOptions.node>(<label>)
<else>
<label>_tree = self._adaptor.dupNode(<label>)
<endif><\n>
self._adaptor.addChild(root_<treeLevel>, <label>_tree)
})>
<endif>
}, ...)>
>>

```

```

matchRuleBlockSet(s,label,elementIndex,postmatchCode,treeLevel="0",terminalOptions={}) ::= <<
<matchSet(...)>
<noRewrite(...)> <! set return tree !>
>>

```

```

matchSetBang(s,label,elementIndex,postmatchCode,terminalOptions={}) ::= <<
_last = self.input.LT(1)
<super.matchSet(...)>
>>

```

```

matchSetRuleRoot(s,label,elementIndex,debug,terminalOptions={}) ::= <<
<super.matchSet(postmatchCode={
<if(!rewriteMode)>
<finishedBacktracking({
<if(terminalOptions.node)>
<label>_tree = <terminalOptions.node>(<label>)
<else>
<label>_tree = self._adaptor.dupNode(<label>)
<endif><\n>
root_<treeLevel> = self._adaptor.becomeRoot(<label>_tree, root_<treeLevel>)
})>
<endif>
}, ...)>
>>

```

```
// RULE REF AST
```

```

/** rule auto construct */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
_last = self.input.LT(1)
<super.ruleRef(...)>
<finishedBacktracking({
<if(!rewriteMode)>
self._adaptor.addChild(root_<treeLevel>, <label>.tree)

```

```

<else> <! rewrite mode !>
if _first_<treeLevel> is None:
  _first_<treeLevel> = <label>.tree<\n>
<endif>
})>
>>

/** x+=rule auto construct */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(label, {<label>.tree})>
>>

/** ^(rule ...) auto construct */
ruleRefRuleRoot(rule,label,elementIndex,args,scope) ::= <<
_last = self.input.LT(1)
<super.ruleRef(...)>
<if(!rewriteMode)>
<finishedBacktracking({
root_<treeLevel> = self._adaptor.becomeRoot(<label>.tree, root_<treeLevel>)
})>
<endif>
>>

/** ^(x+=rule ...) auto construct */
ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<listLabel(label, {<label>.tree})>
>>

/** rule when output=AST and tracking for rewrite */
ruleRefTrack(rule,label,elementIndex,args,scope) ::= <<
_last = self.input.LT(1)
<super.ruleRefTrack(...)>
>>

/** x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
_last = self.input.LT(1)
<super.ruleRefTrackAndListLabel(...)>
>>

/** ^(rule ...) rewrite */
ruleRefRuleRootTrack(rule,label,elementIndex,args,scope) ::= <<
_last = self.input.LT(1)
<super.ruleRefRootTrack(...)>
>>

```

```

/** ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
  _last = self.input.LT(1)
  <super.ruleRefRuleRootTrackAndListLabel(...)>
>>

```

```

/** Streams for token refs are tree nodes now; override to
 * change nextToken to nextNode.
 */
createRewriteNodeFromElement(token,scope,terminalOptions={}) ::= <<
  <if(terminalOptions.node)>
  <terminalOptions.node>(stream_<token>.nextNode())
  <else>
  stream_<token>.nextNode()
  <endif>
>>

```

```

ruleCleanUp() ::= <<
  <super.ruleCleanUp()>
  <if(!rewriteMode)>
  <finishedBacktracking({
  retval.tree = self._adaptor.rulePostProcessing(root_0)
  })>
  <endif>
>>

```

Found in path(s):

```

* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-
jar/org/antlr/codegen/templates/Python/ASTTreeParser.stg

```

No license file was found, but licenses were detected in source scan.

```

/*
 * [The "BSD license"]
 * Copyright (c) 2010 Terence Parr
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * 3. The name of the author may not be used to endorse or promote products
 * derived from this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR

```

* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):

* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/misc/MultiMap.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/misc/Utils.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/tool/RuleLabelScope.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/codegen/Python3Target.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/analysis/StateCluster.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/misc/IntervalSet.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/tool/GrammarSerializerFoo.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/analysis/DFAOptimizer.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-
jar/org/antlr/tool/GrammarSyntaxMessage.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/tool/NFAFactory.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/tool/Message.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/analysis/NFAState.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/codegen/CTarget.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-
jar/org/antlr/tool/GrammarUnreachableAltsMessage.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/Tool.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/analysis/NFAContext.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/tool/Grammar.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-
jar/org/antlr/analysis/NFAToDFAConverter.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/tool/NameSpaceChecker.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/tool/GrammarReport2.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/misc/IntArrayList.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/codegen/DelphiTarget.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-
jar/org/antlr/analysis/AnalysisTimeoutException.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/codegen/Perl5Target.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/analysis/LookaheadSet.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/misc/BitSet.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/analysis/LL1Analyzer.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/analysis/NFA.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/misc/MutableInteger.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/tool/GrammarSpelunker.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/codegen/ActionScriptTarget.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-

jar/org/antlr/tool/GrammarDanglingStateMessage.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-
jar/org/antlr/tool/BuildDependencyGenerator.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/tool/CompositeGrammar.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/analysis/DecisionProbe.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/codegen/CSharp2Target.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-
jar/org/antlr/tool/CompositeGrammarTree.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/analysis/NFAConfiguration.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/tool/Interpreter.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/analysis/SemanticContext.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/codegen/JavaScriptTarget.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-
jar/org/antlr/tool/GrammarSemanticsMessage.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-
jar/org/antlr/tool/GrammarInsufficientPredicatesMessage.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-
jar/org/antlr/analysis/NFAConversionThread.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-
jar/org/antlr/analysis/RuleClosureTransition.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/tool/Rule.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/tool/Attribute.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/misc/Barrier.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/codegen/PythonTarget.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/tool/DOTGenerator.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/tool/ToolMessage.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/tool/RandomPhrase.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/tool/Strip.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/analysis/State.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-
jar/org/antlr/analysis/AnalysisRecursionOverflowException.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/analysis/Label.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-
jar/org/antlr/tool/LeftRecursionCyclesMessage.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/codegen/CppTarget.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-
jar/org/antlr/tool/AssignTokenTypesBehavior.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/tool/AttributeScope.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/misc/Interval.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-
jar/org/antlr/tool/GrammarNonDeterminismMessage.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/analysis/DFASState.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/misc/IntSet.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/analysis/Transition.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/codegen/Target.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/tool/Interp.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/analysis/LL1DFA.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-

jar/org/antlr/tool/RecursionOverflowMessage.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-
jar/org/antlr/tool/GrammarAnalysisAbortedMessage.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/tool/ANTLRErrorListener.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-
jar/org/antlr/tool/NonRegularDecisionMessage.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/tool/GrammarSanity.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/analysis/PredicateLabel.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-
jar/org/antlr/analysis/NonLLStarDecisionException.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-
jar/org/antlr/codegen/ACyclicDFACodeGenerator.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/codegen/JavaTarget.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/tool/GrammarAST.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/tool/GrammarReport.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/analysis/ActionLabel.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/misc/Graph.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/tool/ErrorHandler.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/analysis/DFA.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/tool/FASerializer.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/misc/OrderedHashSet.java
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/analysis/MachineProbe.java
No license file was found, but licenses were detected in source scan.

/*

[The "BSD license"]

Copyright (c) 2005-2012 Terence Parr

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT

(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

/** Templates for building ASTs during tree parsing.

*

* Deal with many combinations. Dimensions are:

* Auto build or rewrite

* no label, label, list label (label/no-label handled together)

* child, root

* token, set, rule, wildcard

*

* Each combination has its own template except that label/no label

* is combined into tokenRef, ruleRef, ...

*/

finishedBacktracking(block) ::= <<

<if(backtracking)>

if <actions.(actionScope).synpredgate>:

 <block>

<else>

 <block>

<endif>

>>

/** Add a variable to track last element matched */

ruleDeclarations() ::= <<

<super.ruleDeclarations()>

 _first_0 = None

 _last = None<\n>

>>

/** What to emit when there is no rewrite rule. For auto build

* mode, does nothing.

*/

noRewrite(rewriteBlockLevel, treeLevel) ::= <<

<finishedBacktracking({

 <if(rewriteMode)>

 retval.tree = _first_0

 if self._adaptor.getParent(retval.tree) is not None and self._adaptor.isNil(self._adaptor.getParent(retval.tree)):

 retval.tree = self._adaptor.getParent(retval.tree)

 <endif>

 })>

>>

/** match ^(root children) in tree parser; override here to

* add tree construction actions.

*/

```

tree(root, actionsAfterRoot, children, nullableChildList,
  enclosingTreeLevel, treeLevel) ::= <<
  _last = self.input.LT(1)
  _save_last_<treeLevel> = _last
  _first_<treeLevel> = None
  <if(!rewriteMode)>
  root_<treeLevel> = self._adaptor.nil()<\n>
  <endif>
  <root:element()>
  <if(rewriteMode)>
  <finishedBacktracking({
  <if(root.el.rule)>
  if _first_<enclosingTreeLevel> is None:
    _first_<enclosingTreeLevel> = <root.el.label>.tree<\n>
  <else>
  if _first_<enclosingTreeLevel> is None:
    _first_<enclosingTreeLevel> = <root.el.label><\n>
  <endif>
  })>
  <endif>
  <actionsAfterRoot:element()>
  <if(nullableChildList)>
  if self.input.LA(1) == DOWN:
    self.match(self.input, DOWN, None)
    <children:element()>
    self.match(self.input, UP, None)

  <else>
  self.match(self.input, DOWN, None)
  <children:element()>
  self.match(self.input, UP, None)<\n>
  <endif>
  <if(!rewriteMode)>
  self._adaptor.addChild(root_<enclosingTreeLevel>, root_<treeLevel>)<\n>
  <endif>
  _last = _save_last_<treeLevel>

  >>

// TOKEN AST STUFF

/** ID! and output=AST (same as plain tokenRef) 'cept add
 * setting of _last
 */
tokenRefBang(token,label,elementIndex,terminalOptions={ }) ::= <<
  _last = self.input.LT(1)
  <super.tokenRef(...)>
  >>

```

```

/** ID auto construct */
tokenRef(token,label,elementIndex,terminalOptions={}) ::= <<
  _last = self.input.LT(1)
  <super.tokenRef(...)>
  <if(!rewriteMode)>
  <finishedBacktracking({
  <if(terminalOptions.node)>
  <label>_tree = <terminalOptions.node>(<label>)
  <else>
  <label>_tree = self._adaptor.dupNode(<label>)
  <endif><\n>
  self._adaptor.addChild(root_<treeLevel>, <label>_tree)
  })>
  <else> <! rewrite mode !>
  <finishedBacktracking({
  if _first_<treeLevel> is None:
    _first_<treeLevel> = <label><\n>
  })>
  <endif>
  >>

/** label+=TOKEN auto construct */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
  <tokenRef(...)>
  <listLabel(elem=label,...)>
  >>

/** ^(ID ...) auto construct */
tokenRefRuleRoot(token,label,elementIndex,terminalOptions={}) ::= <<
  _last = self.input.LT(1)
  <super.tokenRef(...)>
  <if(!rewriteMode)>
  <finishedBacktracking({
  <if(terminalOptions.node)>
  <label>_tree = <terminalOptions.node>(<label>)
  <else>
  <label>_tree = self._adaptor.dupNode(<label>)
  <endif><\n>
  root_<treeLevel> = self._adaptor.becomeRoot(<label>_tree, root_<treeLevel>)
  })>
  <endif>
  >>

/** Match ^(label+=TOKEN ...) auto construct */
tokenRefRuleRootAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
  <tokenRefRuleRoot(...)>
  <listLabel(elem=label,...)>

```

```

>>

/** Match . wildcard and auto dup the node/subtree */
wildcard(token,label,elementIndex,terminalOptions={ }) ::= <<
  _last = self.input.LT(1)
  <super.wildcard(...)>
  <if(!rewriteMode)>
  <finishedBacktracking({
  <label>_tree = self._adaptor.dupTree(<label>)
  self._adaptor.addChild(root_<treeLevel>, <label>_tree)
  })>
  <else> <! rewrite mode !>
  <finishedBacktracking({
  if _first_<treeLevel> is None:
    _first_<treeLevel> = <label>
  })>
  <endif>
>>

// SET AST
matchSet(s,label,elementIndex,postmatchCode,terminalOptions={ }) ::= <<
  _last = self.input.LT(1)
  <super.matchSet(postmatchCode={
  <if(!rewriteMode)>
  <finishedBacktracking({
  <if(terminalOptions.node)>
  <label>_tree = <terminalOptions.node>(<label>)
  <else>
  <label>_tree = self._adaptor.dupNode(<label>)
  <endif><\n>
  self._adaptor.addChild(root_<treeLevel>, <label>_tree)
  })>
  <endif>
  }, ...)>
>>

matchRuleBlockSet(s,label,elementIndex,postmatchCode,treeLevel="0",terminalOptions={ }) ::= <<
  <matchSet(...)>
  <noRewrite(...)> <! set return tree !>
>>

matchSetBang(s,label,elementIndex,postmatchCode,terminalOptions={ }) ::= <<
  _last = self.input.LT(1)
  <super.matchSet(...)>
>>

matchSetRuleRoot(s,label,elementIndex,debug,terminalOptions={ }) ::= <<
  <super.matchSet(postmatchCode={

```

```

<if(!rewriteMode)>
<finishedBacktracking({
<if(terminalOptions.node)>
<label>_tree = <terminalOptions.node>(<label>)
<else>
<label>_tree = self._adaptor.dupNode(<label>)
<endif><\n>
root_<treeLevel> = self._adaptor.becomeRoot(<label>_tree, root_<treeLevel>)
})>
<endif>
}, ...)>
>>

```

```
// RULE REF AST
```

```

/** rule auto construct */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
  _last = self.input.LT(1)
  <super.ruleRef(...)>
  <finishedBacktracking({
  <if(!rewriteMode)>
  self._adaptor.addChild(root_<treeLevel>, <label>.tree)
  <else> <! rewrite mode !>
  if _first_<treeLevel> is None:
    _first_<treeLevel> = <label>.tree<\n>
  <endif>
  })>
  >>

```

```

/** x+=rule auto construct */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
  <ruleRef(...)>
  <listLabel(label, {<label>.tree})>
  >>

```

```

/** ^(rule ...) auto construct */
ruleRefRuleRoot(rule,label,elementIndex,args,scope) ::= <<
  _last = self.input.LT(1)
  <super.ruleRef(...)>
  <if(!rewriteMode)>
  <finishedBacktracking({
  root_<treeLevel> = self._adaptor.becomeRoot(<label>.tree, root_<treeLevel>)
  })>
  <endif>
  >>

```

```

/** ^(x+=rule ...) auto construct */
ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope) ::= <<

```

```

<ruleRefRuleRoot(...)>
<listLabel(label, {<label>.tree})>
>>

/** rule when output=AST and tracking for rewrite */
ruleRefTrack(rule,label,elementIndex,args,scope) ::= <<
  _last = self.input.LT(1)
  <super.ruleRefTrack(...)>
>>

/** x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
  _last = self.input.LT(1)
  <super.ruleRefTrackAndListLabel(...)>
>>

/** ^(rule ...) rewrite */
ruleRefRuleRootTrack(rule,label,elementIndex,args,scope) ::= <<
  _last = self.input.LT(1)
  <super.ruleRefRuleRootTrack(...)>
>>

/** ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
  _last = self.input.LT(1)
  <super.ruleRefRuleRootTrackAndListLabel(...)>
>>

/** Streams for token refs are tree nodes now; override to
 * change nextToken to nextNode.
 */
createRewriteNodeFromElement(token,scope,terminalOptions={}) ::= <<
<if(terminalOptions.node)>
<terminalOptions.node>(stream_<token>.nextNode())
<else>
stream_<token>.nextNode()
<endif>
>>

ruleCleanUp() ::= <<
<super.ruleCleanUp()>
<if(!rewriteMode)>
<finishedBacktracking({
retval.tree = self._adaptor.rulePostProcessing(root_0)
})>
<endif>
>>

```

Found in path(s):

```
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-  
jar/org/antlr/codegen/templates/Python3/ASTTreeParser.stg
```

No license file was found, but licenses were detected in source scan.

```
/*
```

```
[The "BSD license"]
```

```
Copyright (c) 2005-2006 Terence Parr
```

```
Copyright (c) 2008 Ronald Blaschke
```

```
All rights reserved.
```

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```
*/
```

```
/** Templates for building ASTs during tree parsing.
```

```
*
```

```
* Deal with many combinations. Dimensions are:
```

```
* Auto build or rewrite
```

```
* no label, label, list label (label/no-label handled together)
```

```
* child, root
```

```
* token, set, rule, wildcard
```

```
*
```

```
* Each combination has its own template except that label/no label
```

```
* is combined into tokenRef, ruleRef, ...
```

```
*/
```

```
group ASTTreeParser;
```

```
/** Add a variable to track last element matched */
```

```

ruleDeclarations() ::= <<
<super.ruleDeclarations()>
<ASTLabelType> _first_0 = null;
<ASTLabelType> _last = null;<\n>
>>

/** What to emit when there is no rewrite rule. For auto build
* mode, does nothing.
*/
noRewrite(rewriteBlockLevel, treeLevel) ::= <<
<if(backtracking)>if ( state.backtracking==0 ) {<endif>
<if(rewriteMode)>
retval.tree = (<ASTLabelType>)_first_0;
if ( adaptor.getParent(retval.tree)!=null && adaptor.isNil( adaptor.getParent(retval.tree) ) )
    retval.tree = (<ASTLabelType>)adaptor.getParent(retval.tree);
<endif>
<if(backtracking)>}<endif>
>>

/** match ^(root children) in tree parser; override here to
* add tree construction actions.
*/
tree(root, actionsAfterRoot, children, nullableChildList,
    enclosingTreeLevel, treeLevel) ::= <<
_last = (<ASTLabelType>)input.LT(1);
{
<ASTLabelType> _save_last_<treeLevel> = _last;
<ASTLabelType> _first_<treeLevel> = null;
<if(!rewriteMode)>
<ASTLabelType> root_<treeLevel> = (<ASTLabelType>)adaptor.nil();
<endif>
<root:element()>
<if(rewriteMode)>
<if(backtracking)>if ( state.backtracking==0 )<endif>
<if(root.el.rule)>
if ( _first_<enclosingTreeLevel>==null ) _first_<enclosingTreeLevel> = <root.el.label>.tree;
<else>
if ( _first_<enclosingTreeLevel>==null ) _first_<enclosingTreeLevel> = <root.el.label>;
<endif>
<endif>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if ( input.LA(1)==Token.DOWN ) {
    match(input, Token.DOWN, null); <checkRuleBacktrackFailure()>
    <children:element()>
    match(input, Token.UP, null); <checkRuleBacktrackFailure()>
}
<else>

```

```

match(input, Token.DOWN, null); <checkRuleBacktrackFailure()>
<children:element()>
match(input, Token.UP, null); <checkRuleBacktrackFailure()>
<endif>
<if(!rewriteMode)>
adaptor.addChild(root_<enclosingTreeLevel>, root_<treeLevel>);
<endif>
_last = _save_last_<treeLevel>;
}<\n>
>>

// TOKEN AST STUFF

/** ID! and output=AST (same as plain tokenRef) 'cept add
 * setting of _last
 */
tokenRefBang(token,label,elementIndex) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.tokenRef(...)>
>>

/** ID auto construct */
tokenRef(token,label,elementIndex,terminalOptions) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.tokenRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( state.backtracking==0 ) {<endif>
<if(terminalOptions.node)>
<label>_tree = new <terminalOptions.node>(<label>);
<else>
<label>_tree = (<ASTLabelType>)adaptor.dupNode(<label>);
<endif><\n>
adaptor.addChild(root_<treeLevel>, <label>_tree);
<if(backtracking)>}<endif>
<else> <! rewrite mode !>
<if(backtracking)>if ( state.backtracking==0 )<endif>
if ( _first_<treeLevel>==null ) _first_<treeLevel> = <label>;
<endif>
>>

/** label+=TOKEN auto construct */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

/** ^(ID ...) auto construct */
tokenRefRuleRoot(token,label,elementIndex,terminalOptions) ::= <<

```

```

_last = (<ASTLabelType>)input.LT(1);
<super.tokenRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( state.backtracking==0 ) {<endif>
<if(terminalOptions.node)>
<label>_tree = new <terminalOptions.node>(<label>);
<else>
<label>_tree = (<ASTLabelType>)adaptor.dupNode(<label>);
<endif><\n>
root_<treeLevel> = (<ASTLabelType>)adaptor.becomeRoot(<label>_tree, root_<treeLevel>);
<if(backtracking)>}<endif>
<endif>
>>

/** Match ^(label+=TOKEN ...) auto construct */
tokenRefRuleRootAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<tokenRefRuleRoot(...)>
<listLabel(elem=label,...)>
>>

// SET AST

matchSet(s,label,terminalOptions,elementIndex,postmatchCode) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.matchSet(..., postmatchCode={
<if(!rewriteMode)>
<if(backtracking)>if ( state.backtracking==0 ) {<endif>
<if(terminalOptions.node)>
<label>_tree = new <terminalOptions.node>(<label>);
<else>
<label>_tree = (<ASTLabelType>)adaptor.dupNode(<label>);
<endif><\n>
adaptor.addChild(root_<treeLevel>, <label>_tree);
<if(backtracking)>}<endif>
<endif>
}
)>
>>

matchRuleBlockSet(s,label,terminalOptions,elementIndex,postmatchCode,treeLevel="0") ::= <<
<matchSet(...)>
<noRewrite()> <! set return tree !>
>>

matchSetBang(s,label,elementIndex,postmatchCode) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.matchSet(...)>
>>

```

```

matchSetRuleRoot(s,label,terminalOptions,elementIndex,debug) ::= <<
<super.matchSet(..., postmatchCode={
<if(!rewriteMode)>
<if(backtracking)>if ( state.backtracking==0 ) {<endif>
<if(terminalOptions.node)>
<label>_tree = new <terminalOptions.node>(<label>);
<else>
<label>_tree = (<ASTLabelType>)adaptor.dupNode(<label>);
<endif><\n>
root_<treeLevel> = (<ASTLabelType>)adaptor.becomeRoot(<label>_tree, root_<treeLevel>);
<if(backtracking)>}<endif>
<endif>
}
)>
>>

```

```
// RULE REF AST
```

```

/** rule auto construct */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRef(...)>
<if(backtracking)>if ( state.backtracking==0 ) <endif>
<if(!rewriteMode)>
adaptor.addChild(root_<treeLevel>, <label>.getTree());
<else> <! rewrite mode !>
if ( _first_<treeLevel>==null ) _first_<treeLevel> = <label>.tree;
<endif>
>>

```

```

/** x+=rule auto construct */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(elem=label+".getTree()",...)>
>>

```

```

/** ^(rule ...) auto construct */
ruleRefRuleRoot(rule,label,elementIndex,args,scope) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( state.backtracking==0 ) <endif>root_<treeLevel> =
(<ASTLabelType>)adaptor.becomeRoot(<label>.getTree(), root_<treeLevel>);
<endif>
>>

```

```

/** ^(x+=rule ...) auto construct */

```

```
ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<listLabel(elem=label+".getTree()",...)>
>>
```

```
/** Streams for token refs are tree nodes now; override to
 * change nextToken to nextNode.
 */
createRewriteNodeFromElement(token,terminalOptions,scope) ::= <<
<if(terminalOptions.node)>
new <terminalOptions.node>(stream_<token>.nextNode())
<else>
stream_<token>.nextNode()
<endif>
>>
```

```
ruleCleanup() ::= <<
<super.ruleCleanup(>
<if(!rewriteMode)>
<if(backtracking)>if ( state.backtracking==0 ) {<\n><endif>
retval.tree = (<ASTLabelType>)adaptor.rulePostProcessing(root_0);
<if(backtracking)>}<endif>
<endif>
>>
```

Found in path(s):

```
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-
jar/org/antlr/codegen/templates/Perl5/ASTTreeParser.stg
```

No license file was found, but licenses were detected in source scan.

```
/*
```

```
[The "BSD license"]
```

```
Copyright (c) 2006 Kay Roepke
```

```
All rights reserved.
```

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES

OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

/*

New style messages. This file contains the actual layout of the messages emitted by ANTLR.

The text itself is coming out of the languages/*stg files, according to the chosen locale.

This file contains the format that mimicks GCC output.

*/

group gnu;

location(file, line, column) ::= "<file>:<line>:"

message(id, text) ::= "<text> (<id>)"

report(location, message, type) ::= "<location> <type>: <message>"

wantsSingleLineMessage() ::= "true"

Found in path(s):

* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-
jar/org/antlr/tool/templates/messages/formats/gnu.stg

No license file was found, but licenses were detected in source scan.

/*

[The "BSD license"]

Copyright (c) 2010 Terence Parr

All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES

OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

/*

New style messages. This file only contains the messages in English, but no
information about which file, line, or column it occurred in.

The location and message ids are taken out of the formats directory.

Kay Roepke

*/

group en_US;

// TOOL ERRORS

// file errors

CANNOT_WRITE_FILE(arg,exception,stackTrace) ::= <<

cannot write file <arg>: <exception>

<stackTrace; separator="\n">

>>

CANNOT_CLOSE_FILE(arg,exception,stackTrace) ::= "cannot close file <arg>: <exception>"

CANNOT_FIND_TOKENS_FILE(arg) ::= "cannot find tokens file <arg>"

ERROR_READING_TOKENS_FILE(arg,exception,stackTrace) ::= <<

problem reading token vocabulary file <arg>: <exception>

<stackTrace; separator="\n">

>>

DIR_NOT_FOUND(arg,exception,stackTrace) ::= "directory not found: <arg>"

OUTPUT_DIR_IS_FILE(arg,exception,stackTrace) ::= "output directory is a file: <arg>"

CANNOT_OPEN_FILE(arg,exception,stackTrace) ::= "cannot find or open file: <arg><if(exception)>; reason:
<exception><endif>"

CIRCULAR_DEPENDENCY() ::= "your grammars contain a circular dependency and cannot be sorted into a valid
build order."

INTERNAL_ERROR(arg,arg2,exception,stackTrace) ::= <<

internal error: <arg> <arg2><if(exception)>: <exception><endif>

<stackTrace; separator="\n">

>>

INTERNAL_WARNING(arg) ::= "internal warning: <arg>"

ERROR_CREATING_ARTIFICIAL_RULE(arg,exception,stackTrace) ::= <<

problems creating lexer rule listing all tokens: <exception>

<stackTrace; separator="\n">

>>

TOKENS_FILE_SYNTAX_ERROR(arg,arg2) ::=

"problems parsing token vocabulary file <arg> on line <arg2>"

CANNOT_GEN_DOT_FILE(arg,exception,stackTrace) ::=

```

"cannot write DFA DOT file <arg>: <exception>"
BAD_ACTION_AST_STRUCTURE(exception,stackTrace) ::=
"bad internal tree structure for action '<arg>': <exception>"
BAD_AST_STRUCTURE(arg,exception,stackTrace) ::= <<
bad internal tree structure '<arg>': <exception>
<stackTrace; separator="\n">
>>
FILE_AND_GRAMMAR_NAME_DIFFER(arg,arg2) ::=
"file <arg2> contains grammar <arg>; names must be identical"
FILENAME_EXTENSION_ERROR(arg) ::=
"file <arg> must end in a file extension, normally .g"

// code gen errors
MISSING_CODE_GEN_TEMPLATES(arg) ::=
"cannot find code generation templates <arg>.stg"
MISSING_CYCLIC_DFA_CODE_GEN_TEMPLATES() ::=
"cannot find code generation cyclic DFA templates for language <arg>"
CODE_GEN_TEMPLATES_INCOMPLETE(arg) ::=
"at least one code generation template missing for language <arg>"
CANNOT_CREATE_TARGET_GENERATOR(arg,exception,stackTrace) ::=
"cannot create target <arg> code generator: <exception>"
STRING_TEMPLATE_ERROR(arg,exception,stackTrace) ::=
"template error: <arg>"
CANNOT_COMPUTE_SAMPLE_INPUT_SEQ() ::=
"cannot generate a sample input sequence from lookahead DFA"

// grammar interpretation errors
/*
NO_VIABLE_DFA_ALT(arg,arg2) ::=
"no viable transition from state <arg> on <arg2> while interpreting DFA"
*/

// GRAMMAR ERRORS
SYNTAX_ERROR(arg) ::= "syntax error: <arg>"
RULE_REDEFINITION(arg) ::=
"rule <arg> redefinition"
LEXER_RULES_NOT_ALLOWED(arg) ::=
"lexer rule <arg> not allowed in parser"
PARSER_RULES_NOT_ALLOWED(arg) ::=
"parser rule <arg> not allowed in lexer"
CANNOT_FIND_ATTRIBUTE_NAME_IN_DECL(arg) ::=
"cannot find an attribute name in attribute declaration"
NO_TOKEN_DEFINITION(arg) ::=
"no lexer rule corresponding to token: <arg>"
UNDEFINED_RULE_REF(arg) ::=
"reference to undefined rule: <arg>"
LITERAL_NOT_ASSOCIATED_WITH_LEXER_RULE(arg) ::=
"literal has no associated lexer rule: <arg>"

```

CANNOT_ALIAS_TOKENS_IN_LEXER(arg) ::=
 "literals are illegal in lexer tokens{ } section: <arg>"

ATTRIBUTE_REF_NOT_IN_RULE(arg,arg2) ::=
 "reference to attribute outside of a rule: <arg><if(arg2)>.<arg2><endif>"

UNKNOWN_ATTRIBUTE_IN_SCOPE(arg,arg2) ::=
 "unknown attribute for <arg>: <arg2>"

UNKNOWN_RULE_ATTRIBUTE(arg,arg2) ::=
 "unknown attribute for rule <arg>: <arg2>"

UNKNOWN_SIMPLE_ATTRIBUTE(arg,args2) ::=
 "attribute is not a token, parameter, or return value: <arg>"

ISOLATED_RULE_SCOPE(arg) ::=
 "missing attribute access on rule scope: <arg>"

INVALID_RULE_PARAMETER_REF(arg,arg2) ::=
 "cannot access rule <arg>'s parameter: <arg2>"

INVALID_RULE_SCOPE_ATTRIBUTE_REF(arg,arg2) ::=
 "cannot access rule <arg>'s dynamically-scoped attribute: <arg2>"

SYMBOL_CONFLICTS_WITH_GLOBAL_SCOPE(arg) ::=
 "symbol <arg> conflicts with global dynamic scope with same name"

WRITE_TO_READONLY_ATTR(arg,arg2,arg3) ::=
 "cannot write to read only attribute: \$<arg><if(arg2)>.<arg2><endif>"

LABEL_CONFLICTS_WITH_RULE(arg) ::=
 "label <arg> conflicts with rule with same name"

LABEL_CONFLICTS_WITH_TOKEN(arg) ::=
 "label <arg> conflicts with token with same name"

LABEL_CONFLICTS_WITH_RULE_SCOPE_ATTRIBUTE(arg,arg2) ::=
 "label <arg> conflicts with rule <arg2>'s dynamically-scoped attribute with same name"

LABEL_CONFLICTS_WITH_RULE_ARG_RETVAL(arg,arg2) ::=
 "label <arg> conflicts with rule <arg2>'s return value or parameter with same name"

ATTRIBUTE_CONFLICTS_WITH_RULE(arg,arg2) ::=
 "rule <arg2>'s dynamically-scoped attribute <arg> conflicts with the rule name"

ATTRIBUTE_CONFLICTS_WITH_RULE_ARG_RETVAL(arg,arg2) ::=
 "rule <arg2>'s dynamically-scoped attribute <arg> conflicts with <arg2>'s return value or parameter with same name"

LABEL_TYPE_CONFLICT(arg,arg2) ::=
 "label <arg> type mismatch with previous definition: <arg2>"

ARG_RETVAL_CONFLICT(arg,arg2) ::=
 "rule <arg2>'s argument <arg> conflicts a return value with same name"

NONUNIQUE_REF(arg) ::=
 "<arg> is a non-unique reference"

FORWARD_ELEMENT_REF(arg) ::=
 "illegal forward reference: <arg>"

MISSING_RULE_ARGS(arg) ::=
 "missing parameter(s) on rule reference: <arg>"

RULE_HAS_NO_ARGS(arg) ::=
 "rule <arg> has no defined parameters"

ARGS_ON_TOKEN_REF(arg) ::=
 "token reference <arg> may not have parameters"

ILLEGAL_OPTION(arg) ::=

```

"illegal option <arg>"
LIST_LABEL_INVALID_UNLESS_RETVAL_STRUCT(arg) ::=
"rule '+' list labels are not allowed w/o output option: <arg>"
UNDEFINED_TOKEN_REF_IN_REWRITE(arg) ::=
"reference to undefined token in rewrite rule: <arg>"
REWRITE_ELEMENT_NOT_PRESENT_ON_LHS(arg) ::=
"reference to rewrite element <arg> without reference on left of ->"
UNDEFINED_LABEL_REF_IN_REWRITE(arg) ::=
"reference to undefined label in rewrite rule: $<arg>"
NO_GRAMMAR_START_RULE (arg) ::=
"grammar <arg>: no start rule (no rule can obviously be followed by EOF)"
EMPTY_COMPLEMENT(arg) ::= <<
<if(arg)>
set complement ~<arg> is empty
<else>
set complement is empty
<endif>
>>
UNKNOWN_DYNAMIC_SCOPE(arg) ::=
"unknown dynamic scope: <arg>"
UNKNOWN_DYNAMIC_SCOPE_ATTRIBUTE(arg,arg2) ::=
"unknown dynamically-scoped attribute for scope <arg>: <arg2>"
RULE_REF_AMBIG_WITH_RULE_IN_ALT(arg) ::=
"reference $<arg> is ambiguous; rule <arg> is enclosing rule and referenced in the production (assuming enclosing rule)"
ISOLATED_RULE_ATTRIBUTE(arg) ::=
"reference to locally-defined rule scope attribute without rule name: <arg>"
INVALID_ACTION_SCOPE(arg,arg2) ::=
"unknown or invalid action scope for <arg2> grammar: <arg>"
ACTION_REDEFINITION(arg) ::=
"redefinition of <arg> action"
DOUBLE_QUOTES_ILLEGAL(arg) ::=
"string literals must use single quotes (such as '\begin\'): <arg>"
INVALID_TEMPLATE_ACTION(arg) ::=
"invalid StringTemplate % shorthand syntax: '<arg>'"
MISSING_ATTRIBUTE_NAME() ::=
"missing attribute name on $ reference"
ARG_INIT_VALUES_ILLEGAL(arg) ::=
"rule parameters may not have init values: <arg>"
REWRITE_OR_OP_WITH_NO_OUTPUT_OPTION(arg) ::=
"<if(arg)>rule <arg> uses <endif>rewrite syntax or operator with no output option; setting output=AST"
AST_OP_WITH_NON_AST_OUTPUT_OPTION(arg) ::=
"AST operator with non-AST output option: <arg>"
NO_RULES(arg) ::= "grammar file <arg> has no rules"
MISSING_AST_TYPE_IN_TREE_GRAMMAR(arg) ::=
"tree grammar <arg> has no ASTLabelType option"
REWRITE_FOR_MULTI_ELEMENT_ALT(arg) ::=
"with rewrite=true, alt <arg> not simple node or obvious tree element; text attribute for rule not guaranteed to be

```

```

correct"
RULE_INVALID_SET(arg) ::=
    "Cannot complement rule <arg>; not a simple set or element"
HETERO_ILLEGAL_IN_REWRITE_ALT(arg) ::=
    "alts with rewrites can't use heterogeneous types left of ->"
NO_SUCH_GRAMMAR_SCOPE(arg,arg2) ::=
    "reference to undefined grammar in rule reference: <arg>.<arg2>"
NO_SUCH_RULE_IN_SCOPE(arg,arg2) ::=
    "rule <arg2> is not defined in grammar <arg>"
TOKEN_ALIAS_CONFLICT(arg,arg2) ::=
    "cannot alias <arg>; string already assigned to <arg2>"
TOKEN_ALIAS_REASSIGNMENT(arg,arg2) ::=
    "cannot alias <arg>; token name already assigned to <arg2>"
TOKEN_VOCAB_IN_DELEGATE(arg,arg2) ::=
    "tokenVocab option ignored in imported grammar <arg>"
INVALID_IMPORT(arg,arg2) ::=
    "<arg.grammarTypeString> grammar <arg.name> cannot import <arg2.grammarTypeString> grammar
    <arg2.name>"
IMPORTED_TOKENS_RULE_EMPTY(arg,arg2) ::=
    "no lexer rules contributed to <arg> from imported grammar <arg2>"
IMPORT_NAME_CLASH(arg,arg2) ::=
    "combined grammar <arg.name> and imported <arg2.grammarTypeString> grammar <arg2.name> both generate
    <arg2.recognizerName>; import ignored"
AST_OP_IN_ALT_WITH_REWRITE(arg,arg2) ::=
    "rule <arg> alt <arg2> uses rewrite syntax and also an AST operator"
WILDCARD_AS_ROOT(arg) ::= "Wildcard invalid as root; wildcard can itself be a tree"
CONFLICTING_OPTION_IN_TREE_FILTER(arg,arg2) ::= "option <arg>=<arg2> conflicts with tree grammar
    filter mode"
ILLEGAL_OPTION_VALUE(arg, arg2) ::= "value '<arg2>' invalid for option <arg>"
ALL_OPS_NEED_SAME_ASSOC(arg) ::= "all operators of alt <alt> of left-recursive rule must have same
    associativity"
RANGE_OP_ILLEGAL(arg) ::= "the .. range operator isn't allowed in parser rules"

// GRAMMAR WARNINGS

GRAMMAR_NONDETERMINISM(input,conflictingAlts,paths,disabled,hasPredicateBlockedByAction) ::=
<<
<if(paths)>
Decision can match input such as "<input>" using multiple alternatives:
<paths:{ it | alt <it.alt> via NFA path <it.states; separator=","><\n>}>
<else>
Decision can match input such as "<input>" using multiple alternatives: <conflictingAlts; separator=",">
<endif>
<if(disabled)><\n>As a result, alternative(s) <disabled; separator=","> were disabled for that
input<endif><if(hasPredicateBlockedByAction)><\n>Semantic predicates were present but were hidden by
actions.<endif>
>>

```

DANGLING_STATE(danglingAlts,input) ::= <<
the decision cannot distinguish between alternative(s) <danglingAlts; separator=","> for input such as "<input>"
>>

UNREACHABLE_ALTS(alts) ::= <<
The following alternatives can never be matched: <alts; separator=","><\n>
>>

INSUFFICIENT_PREDICATES(upon,altToLocations,hasPredicateBlockedByAction) ::= <<
Input such as "<upon>" is insufficiently covered with predicates at locations: <altToLocations.keys:{alt|alt <alt>:
<altToLocations.(alt):{loc| line <loc.line>:<loc.column> at <loc.text>}; separator="," >}; separator=","
><if(hasPredicateBlockedByAction)><\n>Semantic predicates were present but were hidden by actions.<endif>
>>

DUPLICATE_SET_ENTRY(arg) ::=
"duplicate token type <arg> when collapsing subrule into set"

ANALYSIS_ABORTED(enclosingRule) ::= <<
ANTLR could not analyze this decision in rule <enclosingRule>; often this is because of recursive rule references
visible from the left edge of alternatives. ANTLR will re-analyze the decision with a fixed lookahead of k=1.
Consider using "options {k=1;}" for that decision and possibly adding a syntactic predicate.
>>

RECURSION_OVERFLOW(alt,input,targetRules,callSiteStates) ::= <<
Alternative <alt>: after matching input such as <input> decision cannot predict what comes next due to recursion
overflow <targetRules,callSiteStates:{t,c|to <t> from <c:{s|<s.enclosingRule.name>};separator="," >}; separator=","
and ">
>>

LEFT_RECURSION(targetRules,alt,callSiteStates) ::= <<
Alternative <alt> discovers infinite left-recursion <targetRules,callSiteStates:{t,c|to <t> from
<c:{s|<s.enclosingRule>};separator="," >}; separator="," and ">
>>

UNREACHABLE_TOKENS(tokens) ::= <<
The following token definitions can never be matched because prior tokens match the same input: <tokens;
separator="," >
>>

TOKEN_NONDETERMINISM(input,conflictingTokens,paths,disabled,hasPredicateBlockedByAction) ::= <<
<<
<if(paths>
Decision can match input such as "<input>" using multiple alternatives:
<paths:{ it | alt <it.alt> via NFA path <it.states; separator=","><\n>}>
<else>
Multiple token rules can match input such as "<input>": <conflictingTokens; separator="," ><\n>
<endif>
<if(disabled)><\n>As a result, token(s) <disabled; separator="," > were disabled for that

```
input<endif><if(hasPredicateBlockedByAction)><\n>Semantic predicates were present but were hidden by
actions.<endif>
```

```
>>
```

```
LEFT_RECURSION_CYCLES(listOfCycles) ::= <<
```

```
The following sets of rules are mutually left-recursive <listOfCycles:{c| [<c:{r|<r.name>}; separator=", ">]};
separator=" and ">
```

```
>>
```

```
NONREGULAR_DECISION(ruleName,alts) ::= <<
```

```
[fatal] rule <ruleName> has non-LL(*) decision due to recursive rule invocations reachable from alts <alts;
separator=", ">. Resolve by left-factoring or using syntactic predicates or using backtrack=true option.
```

```
>>
```

```
/* !!on for message levels */
```

```
warning() ::= "warning"
```

```
error() ::= "error"
```

```
Found in path(s):
```

```
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-
jar/org/antlr/tool/templates/messages/languages/en.stg
```

```
No license file was found, but licenses were detected in source scan.
```

```
/*
```

```
* [The "BSD license"]
```

```
* Copyright (c) 2010 Terence Parr
```

```
* All rights reserved.
```

```
*
```

```
* Redistribution and use in source and binary forms, with or without
```

```
* modification, are permitted provided that the following conditions
```

```
* are met:
```

```
* 1. Redistributions of source code must retain the above copyright
```

```
* notice, this list of conditions and the following disclaimer.
```

```
* 2. Redistributions in binary form must reproduce the above copyright
```

```
* notice, this list of conditions and the following disclaimer in the
```

```
* documentation and/or other materials provided with the distribution.
```

```
* 3. The name of the author may not be used to endorse or promote products
```

```
* derived from this software without specific prior written permission.
```

```
*
```

```
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
```

```
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
```

```
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
```

```
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
```

```
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
```

```
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
```

```
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
```

```
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
```

```
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
```

* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

/** ANTLR's code generator.

*

* Generate recognizers derived from grammars. Language independence

* achieved through the use of STGroup objects. All output

* strings are completely encapsulated in the group files such as Java.stg.

* Some computations are done that are unused by a particular language.

* This generator just computes and sets the values into the templates;

* the templates are free to use or not use the information.

*

* To make a new code generation target, define X.stg for language X

* by copying from existing Y.stg most closely related to your language;

* e.g., to do CSharp.stg copy Java.stg. The template group file has a

* bunch of templates that are needed by the code generator. You can add

* a new target w/o even recompiling ANTLR itself. The language=X option

* in a grammar file dictates which templates get loaded/used.

*

* Some language like C need both parser files and header files. Java needs

* to have a separate file for the cyclic DFA as ANTLR generates bytcodes

* directly (which cannot be in the generated parser Java file). To facilitate

* this,

*

* cyclic can be in same file, but header, output must be searpate. recognizer

* is in outptufile.

*/

Found in path(s):

* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/codegen/CodeGenerator.java

No license file was found, but licenses were detected in source scan.

/*

[The "BSD license"]

Copyright (c) 2006 Kay Roepke

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR

IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

/*

New style messages. This file contains the actual layout of the messages emitted by ANTLR.

The text itself is coming out of the languages/*stg files, according to the chosen locale.

This file contains the default format ANTLR uses.

*/

group antlr;

location(file, line, column) ::= "<file>(<line>,<column>)"

message(id, text) ::= "error <id> : <text>"

report(location, message, type) ::= "<location> : <type> <message.id> : <message.text>"

wantsSingleLineMessage() ::= "true"

Found in path(s):

* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/tool/templates/messages/formats/vs2005.stg

No license file was found, but licenses were detected in source scan.

group Dbg;

/*

[The "BSD license"]

Copyright (c) 2005-2009 Terence Parr

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

```
@parserBody.mixins() ::= <<
include ANTLR3::<if(profile)>Profile<else>Debug<endif>::ParserEvents
>>
```

```
@parserBody.additionalMembers() ::= <<
<if(grammar.grammarIsRoot)>
RULE_NAMES = [
:invalid_rule, <grammar.allImportedRules:{rST|:<rST.name>}]; wrap="\n ", separator=", ">
].freeze
<endif>
>>
```

```
@rule.body() ::= <<
in_rule(grammar_file_name, "<ruleName>") do
@debug_listener.location(<ruleDescriptor.tree.line>, <ruleDescriptor.tree.column>)

<@super.body()>

@debug_listener.location(<ruleDescriptor.EORNode.line>, <ruleDescriptor.EORNode.column>)
end
>>
```

```
// Common debug event triggers used by region overrides below
enterSubRule() ::= <<
in_subrule(<decisionNumber>) do<\n>
>>
```

```
exitSubRule() ::= <<
end<\n>
>>
```

```
enterDecision() ::= <<
in_decision(<decisionNumber>) do<\n>
>>
```

```
exitDecision() ::= <<
end<\n>
>>
```

```
enterAlt(n) ::= <<
in_alternative(<n>)<\n>
>>
```

```
// Region overrides that tell various constructs to add debugging triggers
@block.body() ::= <<
in_subrule(<decisionNumber>) do
<@super.body()>
end
>>
```

```
// @blockBody.predecision() ::= "<enterSubRule()>"
// @blockBody.postdecision() ::= "<exitDecision()>"
// @blockBody.postbranch() ::= "<exitSubRule()>"
@blockBody.decision() ::= <<
in_decision(<decisionNumber>) do
<@super.decision()>
end
>>
```

```
@ruleBlock.decision() ::= <<
in_decision(<decisionNumber>) do
<@super.decision()>
end<\n>
>>
```

```
@ruleBlockSingleAlt.prealt() ::= "<enterAlt(n={1})>"
```

```
@blockSingleAlt.prealt() ::= "<enterAlt(n={1})>"
```

```
@positiveClosureBlock.loopBody() ::= <<
```

```
in_subrule(<decisionNumber>) do
  <@super.loopBody()>
end
>>
```

```
@positiveClosureBlockLoop.decisionBody() ::= <<
in_decision(<decisionNumber>) do
  <@super.decisionBody()>
end
>>
```

```
@positiveClosureBlockLoop.earlyExitException() ::= <<
@debug_listener.recognition_exception(eee)
>>
```

```
@closureBlock.loopBody() ::= <<
in_subrule(<decisionNumber>) do
  <@super.loopBody()>
end
>>
```

```
@closureBlockLoop.decisionBody() ::= <<
in_decision(<decisionNumber>) do
  <@super.decisionBody()>
end
>>
```

```
@altSwitchCase.preatt() ::= "<enterAlt(altNum)>" // altNum is arg of altSwitchCase
```

```
element(e) ::= <<
@debug_listener.location(<e.line>, <e.pos>) // e is arg of element
<super.element(e)>
>>
```

```
@matchSet.mismatchedSetException() ::= "@debug_listener.recognition_exception(mse)"
```

```
@dfaState.noViableAltException() ::= <<
nvae = NoViableAlternative( "<description>", <decisionNumber>, <stateNumber> )
@debug_listener.recognition_exception( nvae )
raise( nvae )<\n>
```

>>

```
@dfaStateSwitch.noViableAltException() ::= <<
nvae = NoViableAlternative( "<description>", <decisionNumber>, <stateNumber> )
@debug_listener.recognition_exception( nvae )
raise( nvae )<\n>
>>
```

```
dfaDecision(decisionNumber,description) ::= <<
begin
  @state.cyclic_decision = true
  <super.dfaDecision(...)>
rescue ANTLR3::Error::NoViableAlternative => nvae
  @debug_listener.recognition_exception(nvae)
  raise
end
>>
```

```
@cyclicDFA.errorMethod() ::= <<
def error(nvae)
  @recognizer.debug_listener.recognition_exception(nvae)
  super
end
>>
/** Force predicate validation to trigger an event */
evalPredicate(pred,description) ::= <<
predicate?("<description>") { <pred> }
>>
```

Found in path(s):

* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/Ruby/Dbg.stg
No license file was found, but licenses were detected in source scan.

/*

[The "BSD license"]
Copyright (c) 2010 Kyle Yetter
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products

derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

Found in path(s):

* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/codegen/RubyTarget.java

No license file was found, but licenses were detected in source scan.

/*

[The "BSD license"]

Copyright (c) 2005-2009 Jim Idle, Temporal Wave LLC

<http://www.temporal-wave.com>

<http://www.linkedin.com/in/jimidle>

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

```

/** Add an adaptor property that knows how to build trees */
@headerFile.members() ::= <<
/* @headerFile.members() */
pANTLR3_BASE_TREE_ADAPTOR adaptor;
pANTLR3_VECTOR_FACTORY vectors;
/* End @headerFile.members() */
>>

/** Install the tree adaptor interface pointer and anything else that
 * tree parsers and producers require.
 */
@genericParser.apifuncs() ::= <<
<if(PARSER)>
ADAPTOR = ANTLR3_TREE_ADAPTORNew(instream->tstream->tokenSource->strFactory);<\n>
<endif>
ctx->vectors = antlr3VectorFactoryNew(0);
>>

@genericParser.cleanup() ::= <<
ctx->vectors->close(ctx->vectors);
<if(PARSER)>
/* We created the adaptor so we must free it
 */
ADAPTOR->free(ADAPTOR);
<endif>
>>

@returnScope.ruleReturnMembers() ::= <<

<super.ASTLabelType()> tree;

>>

/** Add a variable to track rule's return AST */
ruleDeclarations() ::= <<
<super.ruleDeclarations()>
<ASTLabelType> root_0;<\n>
>>

ruleInitializations() ::= <<
<super.ruleInitializations()>
root_0 = NULL;<\n>
>>

ruleLabelDefs() ::= <<
<super.ruleLabelDefs()>
<ruleDescriptor.tokenLabels:{it | <ASTLabelType> <it.label.text>_tree;}; separator="\n">
<ruleDescriptor.tokenListLabels:{it | <ASTLabelType> <it.label.text>_tree;}; separator="\n">

```

```

<ruleDescriptor.allTokenRefsInAltsWithRewrites
  :{it | pANTLR3_REWRITE_RULE_<rewriteElementType>_STREAM stream_<it>;}; separator="\n">
<ruleDescriptor.allRuleRefsInAltsWithRewrites
  :{it | pANTLR3_REWRITE_RULE_SUBTREE_STREAM stream_<it>;}; separator="\n">
>>

/* Note that we defer the actual creation of any rewrite streams we need here and just initialize
 * them to NULL. This saves creating huge numbers of rewrite streams that cannot be used as only
 * one alt will be taken in a rule, but we are declaring all the streams here. So we define
 * a macro that contains the create code, then use this macro later to check if the stream
 * has been created yet. Checking for NULL is almost free in C.
 */
ruleLabelInitializations() ::= <<
<super.ruleLabelInitializations()>
<ruleDescriptor.tokenLabels:{it | <it.label.text>_tree = NULL;}; separator="\n">
<ruleDescriptor.tokenListLabels:{it | <it.label.text>_tree = NULL;}; separator="\n">

<ruleDescriptor.allTokenRefsInAltsWithRewrites
  :{it | stream_<it> = NULL;
  #define CREATE_stream_<it> if (stream_<it> == NULL) {stream_<it> =
  antlr3RewriteRule<rewriteElementType>StreamNewAE(ADAPTOR, RECOGNIZER, (pANTLR3_UINT8)"token
  <it>"); \} }; separator="\n">
<ruleDescriptor.allRuleRefsInAltsWithRewrites
  :{it | stream_<it> = NULL;
  #define CREATE_stream_<it> if (stream_<it> == NULL) {stream_<it> =
  antlr3RewriteRuleSubtreeStreamNewAE(ADAPTOR, RECOGNIZER, (pANTLR3_UINT8)"rule <it>"); \}};
  separator="\n">

<if(ruleDescriptor.hasMultipleReturnValues)>
  retval.tree = NULL;
<endif>
>>

/** a rule label including default value */
ruleLabelInitVal(label) ::= <<
<super.ruleLabelInitVal(...)>
<label.label.text>.tree = NULL;
>>

/** When doing auto AST construction, we must define some variables;
 * These should be turned off if doing rewrites. This must be a "mode"
 * as a rule could have both rewrite and AST within the same alternative
 * block.
 */
@alt.declarations() ::= <<
<if(autoAST)>
<if(outerAlt)>

```

```

<endif>
<endif>
>>

@alt.initializations() ::= <<
<if(autoAST)>
<if(outerAlt)>
<if(!rewriteMode)>
root_0 = (<ASTLabelType>)(ADAPTOR->nilNode(ADAPTOR));<\n>
<endif>
<endif>
<endif>
>>

// Tracking Rule Elements
//
/** ID but track it for use in a rewrite rule */
tokenRefTrack(token,label,elementIndex,terminalOptions) ::= <<
<tokenRefBang(...)> <! Track implies no auto AST construction!>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) { <endif>CREATE_stream_<token>;
stream_<token>->add(stream_<token>, <label>, NULL);<if(backtracking)> }<endif><\n>
>>

/** ids+=ID and track it for use in a rewrite rule; adds to ids *and*
 * to the tracking list stream_ID for use in the rewrite.
 */
tokenRefTrackAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<tokenRefTrack(...)>
<listLabel(elem=label,...)>
>>

/** ^(ID ...) track for rewrite */
tokenRefRuleRootTrack(token,label,elementIndex,terminalOptions) ::= <<
<tokenRefBang(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>CREATE_stream_<token>; stream_<token>-
>add(stream_<token>, <label>, NULL);<if(backtracking)> }<endif><\n>
>>

wildcardTrack(label,elementIndex) ::= <<
<super.wildcard(...)>
>>

/** rule when output=AST and tracking for rewrite */
ruleRefTrack(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) { <endif>CREATE_stream_<rule.name>;
stream_<rule.name>->add(stream_<rule.name>, <label>.tree, NULL);<if(backtracking)> }<endif>

```

```

>>

/** x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefTrack(...)>
<listLabelTrack(...)>
>>

/** ^(rule ...) rewrite */
ruleRefRuleRootTrack(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) { <endif>CREATE_stream_<rule.name>;
stream_<rule.name>->add(stream_<rule.name>, <label>.tree, NULL);<if(backtracking)> }<endif>
>>

/** ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRootTrack(...)>
<listLabelAST(...)>
>>

// RULE REF AST

/** Match ^(label+=TOKEN ...) track for rewrite */
tokenRefRuleRootTrackAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<tokenRefRuleRootTrack(...)>
<listLabel(elem=label,...)>
>>

/* How to accumulate lists when we are doing rewrite tracking...
*/
listLabelTrack(label) ::= <<
/* listLabelTrack(label)
*/
if (list_<label> == NULL)
{
    list_<label>=ctx->vectors->newVector(ctx->vectors);
}
list_<label>->add(list_<label>, <label>.tree, NULL);
>>

/* How to accumulate lists of rule outputs (only allowed with AST
* option but if the user is going to walk the tree, they will want
* all their custom elements from rule returns.

```

```

*
* Normally, we use inline structures (which the compiler lays down
* code to copy from heap allocations. However, here we want to accumulate copies
* of the returned structures because we are adding them to a list. This only makes sense if the
* grammar is not rewriting the tree as a tree rewrite only preserves the tree, not the object/structure
* returned from the rule. The rewrite will extract the tree pointer. However, if we are not going to
* do a tree re-write, then the user may wish to iterate the structures returned by the rule in
* action code and will expect the user defined returns[] elements to be available when they do this.
* Hence we cannot just preserve the tree that was returned. So, we must copy the local structure and provide
* a function that can free the allocated space. We cannot know how to free user allocated elements and
* presume that the user will know to do this using their own factories for the structures they allocate.
*/
listLabelAST(label) ::= <<
if (list_<label> == NULL)
{
    list_<label>=ctx->vectors->newVector(ctx->vectors);
}
{
    RETURN_TYPE_<label> * tcopy;

    tcopy = (RETURN_TYPE_<label> *)ANTLR3_MALLOC(sizeof(RETURN_TYPE_<label>)); /* Note no
memory allocation checks! */
    ANTLR3_MEMCPY((void *)tcopy, (const void *)&<label>, sizeof(RETURN_TYPE_<label>));
    list_<label>->add(list_<label>, (void *)tcopy, freeScope); /* Add whatever the return type is */<\n>
}
>>

// R e w r i t e

rewriteCode(
    alts,
    description,
    referencedElementsDeep, // ALL referenced elements to right of ->
    referencedTokenLabels,
    referencedTokenListLabels,
    referencedRuleLabels,
    referencedRuleListLabels,
    referencedWildcardLabels,
    referencedWildcardListLabels,
    rewriteBlockLevel,
    enclosingTreeLevel,
    treeLevel) ::=
<<

/* AST REWRITE
* elements      : <referencedElementsDeep; separator=", ">
* token labels  : <referencedTokenLabels; separator=", ">
* rule labels   : <referencedRuleLabels; separator=", ">

```

```

* token list labels : <referencedTokenListLabels; separator=", ">
* rule list labels : <referencedRuleListLabels; separator=", ">
*/
<if(backtracking)>
if ( <actions.(actionScope).synpredgate> ) <\n>
<endif>
{
<rewriteCodeLabelsDecl()>
<rewriteCodeLabelsInit()>
root_0 = (<ASTLabelType>)(ADAPTOR->nilNode(ADAPTOR));
<prevRuleRootRef().tree = root_0;
<alts:rewriteAlt(); separator="else ">
<if(TREE_PARSER)>
<if(rewriteMode)>
<prevRuleRootRef().tree = (<ASTLabelType>)(ADAPTOR->rulePostProcessing(ADAPTOR, root_0));
INPUT->replaceChildren(INPUT, ADAPTOR->getParent(ADAPTOR, retval.start),
ADAPTOR->getChildIndex(ADAPTOR, retval.start),
ADAPTOR->getChildIndex(ADAPTOR, _last),
retval.tree);
<endif>
<endif>
<prevRuleRootRef().tree = root_0; // set result root
<rewriteCodeLabelsFree()>
}
>>

rewriteCodeLabelsDecl() ::= <<
<referencedTokenLabels
: {it | pANTLR3_REWRITE_RULE_<rewriteElementType>_STREAM stream_<it>;};
separator="\n"
>
<referencedTokenListLabels
: {it | pANTLR3_REWRITE_RULE_<rewriteElementType>_STREAM stream_<it>;};
separator="\n"
>
<referencedRuleLabels
: {it | pANTLR3_REWRITE_RULE_SUBTREE_STREAM stream_<it>;};
separator="\n"
>
<referencedRuleListLabels
: {it | pANTLR3_REWRITE_RULE_SUBTREE_STREAM stream_<it>;};
separator="\n"
>
>>

rewriteCodeLabelsInit() ::= <<
<referencedTokenLabels

```

```

: {it | stream_<it>=antlr3RewriteRule<rewriteElementType>StreamNewAEE(ADAPTOR, RECOGNIZER,
(pANTLR3_UINT8)"token <it>", <it>);};
separator="\n"
>
<referencedTokenListLabels
: {it | stream_<it>=antlr3RewriteRule<rewriteElementType>StreamNewAEV(ADAPTOR, RECOGNIZER,
(pANTLR3_UINT8)"token <it>", list_<it>); };
separator="\n"
>
<referencedRuleLabels
: {it | stream_<it>=antlr3RewriteRuleSubtreeStreamNewAEE(ADAPTOR, RECOGNIZER,
(pANTLR3_UINT8)"token <it>", <it>.tree != NULL ? <it>.tree : NULL);};
separator="\n"
>
<referencedRuleListLabels
: {it | stream_<it>=antlr3RewriteRuleSubtreeStreamNewAEV(ADAPTOR, RECOGNIZER,
(pANTLR3_UINT8)"token <it>", list_<it>);};
separator="\n"
>
>>
rewriteCodeLabelsFree() ::= <<
<referencedTokenLabels
: {it | if (stream_<it> != NULL) stream_<it>->free(stream_<it>); };
separator="\n"
>
<referencedTokenListLabels
: {it | if (stream_<it> != NULL) stream_<it>->free(stream_<it>);};
separator="\n"
>
<referencedRuleLabels
: {it | if (stream_<it> != NULL) stream_<it>->free(stream_<it>);};
separator="\n"
>
<referencedRuleListLabels
: {it | if (stream_<it> != NULL) stream_<it>->free(stream_<it>);};
separator="\n"
>
>>

/** Generate code for an optional rewrite block; note it uses the deep ref'd element
 * list rather shallow like other blocks.
 */
rewriteOptionalBlock(
alt,
rewriteBlockLevel,
referencedElementsDeep, // all nested refs
referencedElements, // elements in immediately block; no nested blocks
description) ::=

```

```

<<
// <fileName>:<description>
{
if ( <referencedElementsDeep:{el | (stream_<el> != NULL && stream_<el>->hasNext(stream_<el>)) } ;
separator="|| "> )
{
<alt>
}
<referencedElementsDeep:{el | if ( stream_<el> != NULL) stream_<el>->reset(stream_<el>);<n>>
}<n>
}<n>
>>

rewriteClosureBlock(
alt,
rewriteBlockLevel,
referencedElementsDeep, // all nested refs
referencedElements, // elements in immediately block; no nested blocks
description) ::=
<<
// <fileName>:<description>
{
while ( <referencedElements:{el | (stream_<el> != NULL && stream_<el>->hasNext(stream_<el>)) } ; separator="||
"> )
{
<alt>
}
<referencedElements:{el | if (stream_<el> != NULL) stream_<el>->reset(stream_<el>);<n>>
}<n>
}<n>
>>
RewriteEarlyExitException() ::=
<<
CONSTRUCTEX();
EXCEPTION->type = ANTLR3_REWRITE_EARLY_EXCEPTION;
EXCEPTION->name = (void *)ANTLR3_REWRITE_EARLY_EXCEPTION_NAME;
>>
rewritePositiveClosureBlock(
alt,
rewriteBlockLevel,
referencedElementsDeep, // all nested refs
referencedElements, // elements in immediately block; no nested blocks
description) ::=
<<
if (<referencedElements:{el | (stream_<el> == NULL || !stream_<el>->hasNext(stream_<el>)) } ; separator="|| "> )
{
<RewriteEarlyExitException()>
}
else
{

```

```

while ( <referencedElements:{el | (stream_<el>->hasNext(stream_<el>)) } ; separator="|| "> ) {
  <alt>
}
<referencedElements:{el | stream_<el>->reset(stream_<el>);<\n>}>
}
>>

rewriteAlt(a) ::= <<
// <a.description>
<if(a.pred)>
if (<a.pred>)
{
  <a.alt>
}<\n>
<else>
{
  <a.alt>
}<\n>
<endif>
>>

/** For empty rewrites: "r : ... -> ;" */
rewriteEmptyAlt() ::= "root_0 = NULL; /* \<-- rewriteEmptyAlt() */"

rewriteTree(root,children,description,enclosingTreeLevel,treeLevel) ::= <<
// <fileName>:<description>
{
  <ASTLabelType> root_<treeLevel> = (<ASTLabelType>)(ADAPTOR->nilNode(ADAPTOR));
  <root:rewriteElement()>
  <children:rewriteElement()>
  ADAPTOR->addChild(ADAPTOR, root_<enclosingTreeLevel>, root_<treeLevel>);
}<\n>
>>

rewriteElementList(elements) ::= "<elements:rewriteElement()>"

rewriteElement(e) ::= <<
<@pregen()>
<e.el>
>>

/** Gen ID or ID[args] */
rewriteTokenRef(token,elementIndex,terminalOptions,args) ::= <<
ADAPTOR->addChild(ADAPTOR, root_<treeLevel>, <createRewriteNodeFromElement(...)>);<\n>
>>

/** Gen $label ... where defined via label=ID */
rewriteTokenLabelRef(label,elementIndex) ::= <<

```

```

ADAPTOR->addChild(ADAPTOR, root_<treeLevel>, stream_<label> == NULL ? NULL : stream_<label>-
>nextNode(stream_<label>));<\n>
>>

/** Gen $label ... where defined via label+=ID */
rewriteTokenListLabelRef(label,elementIndex) ::= <<
ADAPTOR->addChild(ADAPTOR, root_<treeLevel>, stream_<label> == NULL ? NULL : stream_<label>-
>nextNode(stream_<label>));<\n>
>>

/** Gen ^($label ...) */
rewriteTokenLabelRefRoot(label,elementIndex) ::= <<
root_<treeLevel> = (<ASTLabelType>)(ADAPTOR->becomeRootToken(ADAPTOR, stream_<label> == NULL ?
NULL : stream_<label>->nextToken(stream_<label>), root_<treeLevel>));<\n>
>>

/** Gen ^($label ...) where label+=... */
rewriteTokenListLabelRefRoot ::= rewriteTokenLabelRefRoot

/** Gen ^(ID ...) or ^(ID[args] ...) */
rewriteTokenRefRoot(token,elementIndex,terminalOptions,args) ::= <<
root_<treeLevel> = (<ASTLabelType>)(ADAPTOR->becomeRoot(ADAPTOR,
<createRewriteNodeFromElement(...)>, root_<treeLevel>));<\n>
>>

rewriteImaginaryTokenRef(args,token,terminalOptions,elementIndex) ::= <<
ADAPTOR->addChild(ADAPTOR, root_<treeLevel>, <createImaginaryNode(tokenType=token, ...)>);<\n>
>>

rewriteImaginaryTokenRefRoot(args,token,terminalOptions,elementIndex) ::= <<
root_<treeLevel> = (<ASTLabelType>)(ADAPTOR->becomeRoot(ADAPTOR,
<createImaginaryNode(tokenType=token, ...)>, root_<treeLevel>));<\n>
>>

/** plain -> {foo} action */
rewriteAction(action) ::= <<
root_0 = <action>;<\n>
>>

/** What is the name of the previous value of this rule's root tree? This
* let's us refer to $rule to mean previous value. I am reusing the
* variable 'tree' sitting in retval struct to hold the value of root_0 right
* before I set it during rewrites. The assign will be to retval.tree.
*/
prevRuleRootRef() ::= "retval"

rewriteRuleRef(rule,dup) ::= <<
ADAPTOR->addChild(ADAPTOR, root_<treeLevel>, stream_<rule> == NULL ? NULL : stream_<rule>-

```

```

>nextTree(stream_<rule>));<\n>
>>

rewriteRuleRefRoot(rule,dup) ::= <<
root_<treeLevel> = (<ASTLabelType>)(ADAPTOR->becomeRoot(ADAPTOR, stream_<rule> == NULL ? NULL
: stream_<rule>->nextNode(stream_<rule>), root_<treeLevel>));<\n>
>>

rewriteNodeAction(action) ::= <<
ADAPTOR->addChild(ADAPTOR, root_<treeLevel>, <action>);<\n>
>>

rewriteNodeActionRoot(action) ::= <<
root_<treeLevel> = (<ASTLabelType>)(ADAPTOR->becomeRoot(ADAPTOR, <action>, root_<treeLevel>));<\n>
>>

/** Gen $ruleLabel ... where defined via ruleLabel=rule */
rewriteRuleLabelRef(label) ::= <<
ADAPTOR->addChild(ADAPTOR, root_<treeLevel>, stream_<label> == NULL ? NULL : stream_<label>-
>nextTree(stream_<label>));<\n>
>>

/** Gen $ruleLabel ... where defined via ruleLabel+=rule */
rewriteRuleListLabelRef(label) ::= <<
ADAPTOR->addChild(ADAPTOR, root_<treeLevel>, stream_<label> == NULL ? NULL : stream_<label>-
>nextTree(stream_<label>));<\n>
>>

/** Gen ^($ruleLabel ...) where ruleLabel=rule */
rewriteRuleLabelRefRoot(label) ::= <<
root_<treeLevel> = (<ASTLabelType>)(ADAPTOR->becomeRoot(ADAPTOR, stream_<label> == NULL ? NULL
: stream_<label>->nextNode(stream_<label>), root_<treeLevel>));<\n>
>>

/** Gen ^($ruleLabel ...) where ruleLabel+=rule */
rewriteRuleListLabelRefRoot(label) ::= <<
root_<treeLevel> = (<ASTLabelType>)(ADAPTOR->becomeRoot((<ASTLabelType>)(stream_<label> == NULL
? NULL : stream_<label>->nextNode(stream_<label>), root_<treeLevel>));<\n>
>>

rewriteWildcardLabelRef(label) ::= <<
ADAPTOR->addChild(ADAPTOR, root_<treeLevel>, stream_<label> == NULL ? NULL : stream_<label>-
>nextTree(stream_<label>));<\n>
>>

createImaginaryNode(tokenType,terminalOptions,args) ::= <<
<if(terminalOptions.node)>
<! new MethodNode(IDLabel, args) !>

```

```

<terminalOptions.node>New(<tokenType><if(args)>, <args; separator=", "><endif>)
<else>
<if(args)>

#if <length(args)> == 2
(<ASTLabelType>)ADAPTOR->createTypeTokenText(ADAPTOR, <tokenType>, TOKTEXT(<args; separator=",
">))
#else
(<ASTLabelType>)ADAPTOR->createTypeText(ADAPTOR, <tokenType>, (pANTLR3_UINT8)<args;
separator=", ">)
#endif

<else>
(<ASTLabelType>)ADAPTOR->createTypeText(ADAPTOR, <tokenType>, (pANTLR3_UINT8)"<tokenType>")
<endif>
<endif>
>>

```

```

createRewriteNodeFromElement(token,terminalOptions,args) ::= <<
<if(terminalOptions.node)>
<terminalOptions.node>New(stream_<token>->nextToken(stream_<token><if(args)>, <args; separator=",
"><endif>)
<else>
<if(args)> <! must create new node from old !>

```

```

#if <length(args)> == 2
ADAPTOR->createTypeTokenText(ADAPTOR, <token>->getType(<token>, TOKTEXT(<token>, <args;
separator=", ">)) /* JIMI */
#else
ADAPTOR->createTypeToken(ADAPTOR, <token>->getType(<token>, <token>, <args; separator=", ">)
#endif

<else>
stream_<token> == NULL ? NULL : stream_<token>->nextNode(stream_<token>)
<endif>
<endif>
>>

```

Found in path(s):

* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/C/AST.stg
No license file was found, but licenses were detected in source scan.

```

/*
* [The "BSD license"]
* Copyright (c) 2007-2008 Johannes Luber
* Copyright (c) 2005-2007 Kunle Odutola
* Copyright (c) 2011 Sam Harwell
* Copyright (c) 2011 Terence Parr

```

```

* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products
* derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/
/** Template subgroup to add template rewrite output
* If debugging, then you'll also get STDbg.stg loaded.
*/

@outputFile.imports() ::= <<
<@super.imports(>
using Antlr.StringTemplate;
using Antlr.StringTemplate.Language;
<if(!backtracking)>
using Hashtable = System.Collections.Hashtable;
<endif>

>>

/** Add this to each rule's return value struct */
@returnScope.ruleReturnMembers() ::= <<
private StringTemplate _st;
public StringTemplate Template { get { return _st; } set { _st = value; } }
public override string ToString() { return (Template==null) ? string.Empty : Template.ToString(); }
>>

@genericParser.members() ::= <<
<@super.members(>
protected StringTemplateGroup templateLib = new StringTemplateGroup("<name>Templates",

```

```

typeof(AngleBracketTemplateLexer) );

public StringTemplateGroup TemplateLib
{
    get { return this.templateLib; }
    set { this.templateLib = value; }
}

/// \<summary> Allows convenient multi-value initialization:
/// "new STAttrMap().Add(...).Add(...)"
/// \</summary>
protected class STAttrMap : Hashtable
{
    public STAttrMap Add(string attrName, object value)
    {
        base.Add(attrName, value);
        return this;
    }
    public STAttrMap Add(string attrName, int value)
    {
        base.Add(attrName, value);
        return this;
    }
}
>>

/** x+=rule when output=template */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(elem=label+".Template",...)>
>>

rewriteTemplate(alts) ::= <<

// TEMPLATE REWRITE
<if(backtracking)>
if ( <actions.(actionScope).synpredgate> )
{
    <alts:rewriteTemplateAlt(); separator="else ">
    <if(rewriteMode)><replaceTextInLine()><endif>
}
<else>
<alts:rewriteTemplateAlt(); separator="else ">
<if(rewriteMode)><replaceTextInLine()><endif>
<endif>
>>

replaceTextInLine() ::= <<

```

```

<if(TREE_PARSER)>
((TokenRewriteStream)input.TokenStream).Replace(
input.TreeAdaptor.GetTokenStartIndex(retval.Start),
input.TreeAdaptor.GetTokenStopIndex(retval.Start),
retval.Template);
<else>
((TokenRewriteStream)input).Replace(
((IToken)retval.Start).TokenIndex,
input.LT(-1).TokenIndex,
retval.Template);
<endif>
>>

```

```

rewriteTemplateAlt() ::= <<
// <it.description>
<if(it.pred)>
if (<it.pred>) {
    retval.Template = <it.alt>;
}<\n>
<else>
{
    retval.Template = <it.alt>;
}<\n>
<endif>
>>

```

```

rewriteEmptyTemplate(alts) ::= <<
null;
>>

```

```

/** Invoke a template with a set of attribute name/value pairs.
 * Set the value of the rule's template *after* having set
 * the attributes because the rule's template might be used as
 * an attribute to build a bigger template; you get a self-embedded
 * template.
 */

```

```

rewriteExternalTemplate(name,args) ::= <<
templateLib.GetInstanceOf("<name>"<if(args)>,
new STAttrMap(<args:{a | .Add("<a.name>", <a.value>)}>
<endif>)
>>

```

```

/** expr is a string expression that says what template to load */
rewriteIndirectTemplate(expr,args) ::= <<
templateLib.GetInstanceOf(<expr><if(args)>,
new STAttrMap(<args:{a | .Add("<a.name>", <a.value>)}>
<endif>)
>>

```

```

/** Invoke an inline template with a set of attribute name/value pairs */
rewriteInlineTemplate(args, template) ::= <<
new StringTemplate(templateLib, "<template>"<if(args)>,
new STAttrMap(<args:{a | .Add("<a.name>", <a.value>)}>
<endif>)
>>

```

```

/** plain -> {foo} action */
rewriteAction(action) ::= <<
<action>
>>

```

```

/** An action has %st.attrName=expr; or % {st}.attrName=expr; */
actionSetAttribute(st,attrName,expr) ::= <<
(<st>).SetAttribute("<attrName>",<expr>);
>>

```

```

/** Translate % {stringExpr} */
actionStringConstructor(stringExpr) ::= <<
new StringTemplate(templateLib,<stringExpr>)
>>

```

Found in path(s):

* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-
jar/org/antlr/codegen/templates/CSharp2/ST.stg

No license file was found, but licenses were detected in source scan.

```

/*
 * [The "BSD license"]
 * Copyright (c) 2007-2008 Johannes Luber
 * Copyright (c) 2005-2007 Kunle Odutola
 * Copyright (c) 2011 Sam Harwell
 * Copyright (c) 2011 Terence Parr
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * 3. The name of the author may not be used to endorse or promote products
 * derived from this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR

```

```

* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

```

```

/** Templates for building ASTs during tree parsing.

```

```

*
* Deal with many combinations. Dimensions are:
* Auto build or rewrite
* no label, label, list label (label/no-label handled together)
* child, root
* token, set, rule, wildcard
*
* Each combination has its own template except that label/no label
* is combined into tokenRef, ruleRef, ...
*/

```

```

/** Add a variable to track last element matched */
ruleDeclarations() ::= <<
<super.ruleDeclarations()>
<if(!ruleDescriptor.isSynPred)>
<ASTLabelType> _first_0 = default(<ASTLabelType>);
<ASTLabelType> _last = default(<ASTLabelType>);
<endif>
>>

```

```

/** What to emit when there is no rewrite rule. For auto build
* mode, does nothing.
*/
noRewrite(rewriteBlockLevel=false, treeLevel=false) ::= <<
<if(!ruleDescriptor.isSynPred)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) {<endif>
<if(rewriteMode)>
retval.Tree = (<ASTLabelType>)_first_0;
if (adaptor.GetParent(retval.Tree)!=null && adaptor.IsNil(adaptor.GetParent(retval.Tree)))
retval.Tree = (<ASTLabelType>)adaptor.GetParent(retval.Tree);
<endif>
<if(backtracking)>}<endif>
<endif>
>>

```

```

/** match ^(root children) in tree parser; override here to

```

```

* add tree construction actions.
*/
tree(root, actionsAfterRoot, children, nullableChildList,
    enclosingTreeLevel, treeLevel) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
{
<ASTLabelType> _save_last_<treeLevel> = _last;
<ASTLabelType> _first_<treeLevel> = default(<ASTLabelType>);
<if(!rewriteMode)>
<ASTLabelType> root_<treeLevel> = (<ASTLabelType>)adaptor.Nil();
<endif>
<root:element()>
<if(rewriteMode)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>)<endif>
<if(root.el.rule)>
if (_first_<enclosingTreeLevel> == null) _first_<enclosingTreeLevel> = <root.el.label>.Tree;
<else>
if (_first_<enclosingTreeLevel> == null) _first_<enclosingTreeLevel> = <root.el.label>;
<endif>
<endif>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if (input.LA(1) == TokenTypes.Down) {
    Match(input, TokenTypes.Down, null); <checkRuleBacktrackFailure()>
    <children:element()>
    Match(input, TokenTypes.Up, null); <checkRuleBacktrackFailure()>
}
<else>
Match(input, TokenTypes.Down, null); <checkRuleBacktrackFailure()>
<children:element()>
Match(input, TokenTypes.Up, null); <checkRuleBacktrackFailure()>
<endif>
<if(!rewriteMode)>
adaptor.AddChild(root_<enclosingTreeLevel>, root_<treeLevel>);
<endif>
_last = _save_last_<treeLevel>;
}
<else>
<super.tree(...)>
<endif>
>>

// TOKEN AST STUFF

/** ID! and output=AST (same as plain tokenRef) 'cept add
* setting of _last
*/

```

```

tokenRefBang(token,label,elementIndex,terminalOptions) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.tokenRef(...)>
<else>
<super.tokenRefBang(...)>
<endif>
>>

/** ID auto construct */
tokenRef(token,label,elementIndex,terminalOptions) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.tokenRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) {<endif>
<if(terminalOptions.node)>
<label>_tree = new
<terminalOptions.node><(if(terminalOptions.type)><terminalOptions.type>,<endif><label><(if(terminalOptions.tex
t)>,<terminalOptions.text; format="string"><endif>);
<else>
<label>_tree =
(<ASTLabelType>)adaptor.DupNode(<(if(terminalOptions.type)><terminalOptions.type>,<endif><label><(if(termin
alOptions.text)>,<terminalOptions.text; format="string"><endif>);
<endif><\n>
adaptor.AddChild(root_<treeLevel>, <label>_tree);
<if(backtracking)>}<endif>
<else> <! rewrite mode !>
<if(backtracking)>if (<actions.(actionScope).synpredgate>)<endif>
if (_first_<treeLevel> == null) _first_<treeLevel> = <label>;
<endif>
<else>
<super.tokenRef(...)>
<endif>
>>

/** label+=TOKEN auto construct */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<if(!ruleDescriptor.isSynPred)>
<tokenRef(...)>
<listLabelElem(elem=label,...)>
<else>
<super.tokenRefAndListLabel(...)>
<endif>
>>

/** ^(ID ...) auto construct */
tokenRefRuleRoot(token,label,elementIndex,terminalOptions) ::= <<

```

```

<if(!ruleDescriptor.isSynPred)>
  _last = (<ASTLabelType>)input.LT(1);
<super.tokenRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) {<endif>
<if(terminalOptions.node)>
<label>_tree = new
<terminalOptions.node>(<if(terminalOptions.type)><terminalOptions.type>,<endif><label><if(terminalOptions.text)>,<terminalOptions.text; format="string"><endif>);
<else>
<label>_tree =
(<ASTLabelType>)adaptor.DupNode(<if(terminalOptions.type)><terminalOptions.type>,<endif><label><if(terminalOptions.text)>,<terminalOptions.text; format="string"><endif>);
<endif><\n>
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(<label>_tree, root_<treeLevel>);
<if(backtracking)>}<endif>
<endif>
<else>
<super.tokenRefRuleRoot(...)>
<endif>
>>

```

```

/** Match ^(label+=TOKEN ...) auto construct */
tokenRefRuleRootAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<if(!ruleDescriptor.isSynPred)>
<tokenRefRuleRoot(...)>
<listLabelElem(elem=label,...)>
<else>
<super.tokenRefRuleRootAndListLabel(...)>
<endif>
>>

```

```

/** Match . wildcard and auto dup the node/subtree */
wildcard(token,label,elementIndex,terminalOptions) ::= <<
<if(!ruleDescriptor.isSynPred)>
  _last = (<ASTLabelType>)input.LT(1);
<super.wildcard(...)>
<if(!rewriteMode)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) {<endif>
<label>_tree = (<ASTLabelType>)adaptor.DupTree(<label>);
adaptor.AddChild(root_<treeLevel>, <label>_tree);
<if(backtracking)>}<endif>
<else> <! rewrite mode !>
<if(backtracking)>if (<actions.(actionScope).synpredgate>)<endif>
if (_first_<treeLevel> == null) _first_<treeLevel> = <label>;
<endif>
<else>
<super.wildcard(...)>

```

```

<endif>
>>

// SET AST

matchSet(s,label,terminalOptions,elementIndex,postmatchCode) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.matchSet(postmatchCode={
<if(!rewriteMode)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) {<endif>
<if(terminalOptions.node)>
<label>_tree = new
<terminalOptions.node><(if(terminalOptions.type)><terminalOptions.type>,<endif><label><if(terminalOptions.tex
t)>,<terminalOptions.text; format="string"><endif>);
<else>
<label>_tree =
(<ASTLabelType>)adaptor.DupNode(<if(terminalOptions.type)><terminalOptions.type>,<endif><label><if(termin
alOptions.text)>,<terminalOptions.text; format="string"><endif>);
<endif><\n>
adaptor.AddChild(root_<treeLevel>, <label>_tree);
<if(backtracking)>\}<endif>
<endif>
}, ...
)>
<else>
<super.matchSet(...)>
<endif>
>>

matchRuleBlockSet(s,label,terminalOptions,elementIndex,postmatchCode,treeLevel="0") ::= <<
<if(!ruleDescriptor.isSynPred)>
<matchSet(...)>
<noRewrite(...)> <! set return tree !>
<else>
<super.matchRuleBlockSet(...)>
<endif>
>>

matchSetBang(s,label,terminalOptions,elementIndex,postmatchCode) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.matchSet(...)>
<else>
<super.matchSetBang(...)>
<endif>
>>

```

```

matchSetRuleRoot(s,label,terminalOptions,elementIndex,debug) ::= <<
<if(!ruleDescriptor.isSynPred)>
<super.matchSet(postmatchCode={
<if(!rewriteMode)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) {<endif>
<if(terminalOptions.node)>
<label>_tree = new
<terminalOptions.node><(if(terminalOptions.type)><terminalOptions.type>,<endif><label><if(terminalOptions.tex
t)>,<terminalOptions.text; format="string"><endif>);
<else>
<label>_tree =
(<ASTLabelType>)adaptor.DupNode(<if(terminalOptions.type)><terminalOptions.type>,<endif><label><if(termin
alOptions.text)>,<terminalOptions.text; format="string"><endif>);
<endif><\n>
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(<label>_tree, root_<treeLevel>);
<if(backtracking)>\}<endif>
<endif>
}, ...
)>
<else>
<super.matchSetRuleRoot(...)>
<endif>
>>

```

```
// RULE REF AST
```

```

/** rule auto construct */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRef(...)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>)<endif>
<if(!rewriteMode)>
adaptor.AddChild(root_<treeLevel>, <label>.Tree);
<else> <! rewrite mode !>
if (_first_<treeLevel> == null) _first_<treeLevel> = <label>.Tree;
<endif>
<else>
<super.ruleRef(...)>
<endif>
>>

```

```

/** x+=rule auto construct */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<if(!ruleDescriptor.isSynPred)>
<ruleRef(...)>
<listLabelElem(elem={<label>.Tree},...)>
<else>

```

```

<super.ruleRefAndListLabel(...)>
<endif>
>>

/** ^(rule ...) auto construct */
ruleRefRuleRoot(rule,label,elementIndex,args,scope) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) <endif>root_<treeLevel> =
(<ASTLabelType>)adaptor.BecomeRoot(<label>.Tree, root_<treeLevel>);
<endif>
<else>
<super.ruleRefRuleRoot(...)>
<endif>
>>

/** ^(x+=rule ...) auto construct */
ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<if(!ruleDescriptor.isSynPred)>
<ruleRefRuleRoot(...)>
<listLabelElem(elem={<label>.Tree},...)>
<else>
<super.ruleRefRuleRootAndListLabel(...)>
<endif>
>>

/** rule when output=AST and tracking for rewrite */
ruleRefTrack(rule,label,elementIndex,args,scope) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRefTrack(...)>
<else>
<super.ruleRefTrack(...)>
<endif>
>>

/** x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRefTrackAndListLabel(...)>
<else>
<super.ruleRefTrackAndListLabel(...)>
<endif>
>>

```

```

/** ^(rule ...) rewrite */
ruleRefRuleRootTrack(rule,label,elementIndex,args,scope) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRefRuleRootTrack(...)>
<else>
<super.ruleRefRuleRootTrack(...)>
<endif>
>>

/** ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRefRuleRootTrackAndListLabel(...)>
<else>
<super.ruleRefRuleRootTrackAndListLabel(...)>
<endif>
>>

/** Streams for token refs are tree nodes now; override to
 * change NextToken to NextNode.
 */
createRewriteNodeFromElement(token,terminalOptions,args) ::= <%
<if(terminalOptions.node)>
new
<terminalOptions.node><if(terminalOptions.type)><terminalOptions.type>,<endif>stream_<token>.NextNode()
<else>
stream_<token>.NextNode()
<endif>
%>

ruleCleanUp() ::= <<
<super.ruleCleanUp()>
<if(!ruleDescriptor.isSynPred)>
<if(!rewriteMode)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) {<endif>
retval.Tree = (<ASTLabelType>)adaptor.RulePostProcessing(root_0);
<if(backtracking)>}<endif>
<endif>
<endif>
>>

```

Found in path(s):

```

* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-
jar/org/antlr/codegen/templates/CSharp2/ASTTreeParser.stg

```

No license file was found, but licenses were detected in source scan.

/*

[The "BSD license"]

Copyright (c) 2005-2012 Terence Parr

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

/** The API version of the runtime that recognizers generated by this runtime

* need.

*/

apiVersion() ::= "1"

/** The overall file structure of a recognizer; stores methods for rules

* and cyclic DFAs plus support code.

*/

outputFile(LEXER,PARSER,TREE_PARSER, actionScope, actions,
docComment, recognizer,
name, tokens, tokenNames, rules, cyclicDFAs,
bitsets, buildTemplate, buildAST, rewriteMode, profile,
backtracking, synpreds, memoize, numRules,
fileName, ANTLRVersion, generatedTimestamp, trace,
scopes, superClass, literals) ::=

<<

\$ANTLR <ANTLRVersion> <fileName> <generatedTimestamp>

<@imports>

import sys

```

from antlr3 import *
<if(TREE_PARSER)>
from antlr3.tree import *<\n>
<endif>
<@end>

<actions.(actionScope).header>

<! <docComment> !>

# for convenience in actions
HIDDEN = BaseRecognizer.HIDDEN

# token types
<tokens:{it | <it.name>=<it.type>}; separator="\n">

# token names
tokenNamesMap = {
    0: "\<invalid>", 1: "\<EOR>", 2: "\<DOWN>", 3: "\<UP>",
    <tokens:{it | <it.type>: "<it.name>"}; wrap, separator=", ">
}
Token.registerTokenNamesMap(tokenNamesMap)

<recognizer>

<if(actions.(actionScope).main)>
<actions.(actionScope).main>
<else>
def main(argv, stdin=sys.stdin, stdout=sys.stdout, stderr=sys.stderr):
<if(LEXER)>
    from antlr3.main import LexerMain
    main = LexerMain(<recognizer.name>)<\n>
<endif>
<if(PARSER)>
    from antlr3.main import ParserMain
    main = ParserMain("<recognizer.grammar.name>Lexer", <recognizer.name>)<\n>
<endif>
<if(TREE_PARSER)>
    from antlr3.main import WalkerMain
    main = WalkerMain(<recognizer.name>)<\n>
<endif>
    main.stdin = stdin
    main.stdout = stdout
    main.stderr = stderr
    main.execute(argv)<\n>
<endif>

<actions.(actionScope).footer>

```

```

if __name__ == '__main__':
    main(sys.argv)

>>

lexer(grammar, name, tokens, scopes, rules, numRules, filterMode,
        labelType="CommonToken", superClass="Lexer") ::= <<
<if(grammar.directDelegates)>
# path hack to allow absolute import of related grammars.
from os.path import dirname
__path__ = [dirname(__file__)]
del dirname

<grammar.directDelegates:
{g|from .<g.recognizerName> import <g.recognizerName>; separator="\n">
<endif>

class <grammar.recognizerName>(<@superClassName><superClass><@end>):
    <scopes: {it|<if(it.isDynamicGlobalScope)><globalAttributeScope(scope=it)><endif>>

    grammarFileName = "<fileName>"
    api_version = <apiVersion()>

    def __init__(self<grammar.delegators: {g|, <g:delegateName()>>, input=None, state=None):
        if state is None:
            state = RecognizerSharedState()
            super().__init__(input, state)

<if(memoize)>
<if(grammar.grammarIsRoot)>
    self._state.ruleMemo = {}
<endif>
<endif>

    <grammar.directDelegates:
    {g|self.<g:delegateName()> = <g.recognizerName>(<trunc(g.delegators): {p|<p:delegateName()>, }>self, input,
state)); separator="\n">
    <grammar.directDelegates:
    {g|<g.delegates: {h|self.<h:delegateName()> = self.<g:delegateName()>.<h:delegateName()>};
separator="\n">}; separator="\n">
    <grammar.delegators:
    {g|self.<g:delegateName()> = <g:delegateName()>}; separator="\n">
    <last(grammar.delegators):
    {g|self.gParent = <g:delegateName()>}; separator="\n">
    self.delegates = [<grammar.delegates: {g|self.<g:delegateName()>}; separator = ", ">]

    <cyclicDFAs: {dfa | <cyclicDFAInit(dfa)>}; separator="\n">

```

```

    <actions.lexer.init>

    <actions.lexer.members>

<if(filterMode)>
    <filteringNextToken()>
<endif>
    <rules; separator="\n\n">

    <synpreds:{p | <lexerSynpred(p)>}>

    <cyclicDFAs:cyclicDFA()> <! dump tables for all DFA !>

>>

/** A override of Lexer.nextToken() that backtracks over mTokens() looking
 * for matches. No error can be generated upon error; just rewind, consume
 * a token and then try again. backtracking needs to be set as well.
 * Make rule memoization happen only at levels above 1 as we start mTokens
 * at backtracking==1.
 */
filteringNextToken() ::= <<
def nextToken(self):
    while True:
        if self.input.LA(1) == EOF:
            return self.makeEOFToken()

        self._state.token = None
        self._state.channel = DEFAULT_CHANNEL
        self._state.tokenStartCharIndex = self.input.index()
        self._state.tokenStartCharPositionInLine = self.input.charPositionInLine
        self._state.tokenStartLine = self.input.line
        self._state._text = None
        try:
            m = self.input.mark()
            try:
                # means we won't throw slow exception
                self._state.backtracking = 1
            try:
                self.mTokens()
            finally:
                self._state.backtracking = 0

        except BacktrackingFailed:

```

```

        # mTokens backtracks with synpred at backtracking==2
        # and we set the synpredgate to allow actions at level 1.
        self.input.rewind(m)
        self.input.consume() # advance one char and try again

    else:
        self.emit()
        return self._state.token

    except RecognitionException as re:
        # shouldn't happen in backtracking mode, but...
        self.reportError(re)
        self.recover(re)

def memoize(self, input, ruleIndex, ruleStartIndex, success):
    if self._state.backtracking > 1:
        # is Lexer always superclass?
        super().memoize(input, ruleIndex, ruleStartIndex, success)

def alreadyParsedRule(self, input, ruleIndex):
    if self._state.backtracking > 1:
        return super().alreadyParsedRule(input, ruleIndex)
    return False

>>

actionGate() ::= "self._state.backtracking == 0"

filteringActionGate() ::= "self._state.backtracking == 1"

/** How to generate a parser */

genericParser(grammar, name, scopes, tokens, tokenNames, rules, numRules,
              bitsets, inputStreamType, superClass, labelType, members,
              rewriteElementType, filterMode, init, ASTLabelType="Object") ::= <<
# token names
tokenNames = [
    "\<invalid>", "\<EOR>", "\<DOWN>", "\<UP>",
    <tokenNames; wrap, separator=", ">
]

<scopes:{it|<if(it.isDynamicGlobalScope)><globalAttributeScopeClass(scope=it)><endif>}>

<if(grammar.directDelegates)>
# path hack to allow absolute import of related grammars.

```

```

from os.path import dirname
__path__ = [dirname(__file__)]
del dirname

<grammar.directDelegates:
{g|from .<g.recognizerName> import <g.recognizerName>; separator="\n">
<endif>

<rules:{it|<ruleAttributeScopeClass(scope=it.ruleDescriptor.ruleScope)>>

class <grammar.recognizerName>(<@superClassName><superClass><@end>):
    grammarFileName = "<fileName>"
    api_version = <apiVersion()>
    tokenNames = tokenNames

    def __init__(self<grammar.delegators:{g|, <g.delegateName()>}>, input, state=None, *args, **kwargs):
        if state is None:
            state = RecognizerSharedState()

        <@args()>
        super().__init__(input, state, *args, **kwargs)

<if(memoize)>
<if(grammar.grammarIsRoot)>
    self._state.ruleMemo = {}
<endif>
<endif>

    <cyclicDFAs:{dfa | <cyclicDFAInit(dfa)>}; separator="\n">

    <scopes:{it | <if(it.isDynamicGlobalScope)><globalAttributeScopeStack(scope=it)><endif>>
    <rules:{it | <ruleAttributeScopeStack(scope=it.ruleDescriptor.ruleScope)>>

    <init>

    <grammar.delegators:
    {g|self.<g.delegateName()> = <g.delegateName()>}; separator="\n">
    <grammar.directDelegates:
    {g|self.<g.delegateName()> = <g.recognizerName>(<trunc(g.delegators):{p|<p.delegateName()>, }>self, input,
state)); separator="\n">
    <grammar.directDelegates:
    {g|<g.delegates:{h|self.<h.delegateName()> = self.<g.delegateName()>.<h.delegateName()>};
separator="\n">}; separator="\n">
    <last(grammar.delegators):
    {g|self.gParent = self.<g.delegateName()>}; separator="\n">
    self.delegates = [<grammar.delegates: {g|self.<g.delegateName()>}; separator = ", ">]

    <@init><@end>

```

```

<@members><@end>

<members>

<rules; separator="\n\n">

<! generate rule/method definitions for imported rules so they
  appear to be defined in this recognizer. !>
<grammar.delegatedRules:{ruleDescriptor| <delegateRule(ruleDescriptor)> }; separator="\n">

<synpreds:{p | <synpred(p)>>

<cyclicDFAs:cyclicDFA()> <! dump tables for all DFA !>

  <bitsets:{it | FOLLOW_<it.name>_in_<it.inName><it.tokenIndex> = frozenset([<it.tokenTypes:{it |
<it>};separator=", ">])<\n>>

>>

delegateRule(ruleDescriptor) ::= <<
def <ruleDescriptor.name>(self, <ruleDescriptor.parameterScope:parameterScope()>):
<\> <if(ruleDescriptor.hasReturnValue)>return
<endif>self.<ruleDescriptor.grammar:delegateName()>.<ruleDescriptor.name><(if(ruleDescriptor.parameterScope)
><ruleDescriptor.parameterScope.attributes:{a|<a.name>}; separator=", "><endif>)

>>

parser(grammar, name, scopes, tokens, tokenNames, rules, numRules, bitsets,
  ASTLabelType="Object", superClass="Parser", labelType="Token",
  members={<actions.parser.members>},
  init={<actions.parser.init>}
) ::= <<
<genericParser(grammar, name, scopes, tokens, tokenNames, rules, numRules,
  bitsets, "TokenStream", superClass,
  labelType, members, "Token",
  false, init, ASTLabelType)>

>>

/** How to generate a tree parser; same as parser except the input
 * stream is a different type.
 */
treeParser(grammar, name, scopes, tokens, tokenNames, globalAction, rules,
  numRules, bitsets, filterMode, labelType={<ASTLabelType>}, ASTLabelType="Object",
superClass={<if(filterMode)><if(buildAST)>TreeRewriter<else>TreeFilter<endif><else>TreeParser<endif>},
  members={<actions.treeparser.members>},

```

```

        init={<actions.treeparser.init>}
    ) ::= <<
<genericParser(grammar, name, scopes, tokens, tokenNames, rules, numRules,
    bitsets, "TreeNodeStream", superClass,
    labelType, members, "Node",
    filterMode, init, ASTLabelType)>
>>

/** A simpler version of a rule template that is specific to the imaginary
 * rules created for syntactic predicates. As they never have return values
 * nor parameters etc..., just give simplest possible method. Don't do
 * any of the normal memoization stuff in here either; it's a waste.
 * As predicates cannot be inlined into the invoking rule, they need to
 * be in a rule by themselves.
 */
synpredRule(ruleName, ruleDescriptor, block, description, nakedBlock) ::=
<<
# $ANTLR start "<ruleName>"
def <ruleName>_fragment(self, <ruleDescriptor.parameterScope:parameterScope(>):
    <ruleLabelDefs(>)
<if(trace)>
    self.traceIn("<ruleName>_fragment", <ruleDescriptor.index>)
    try:
        <block>

    finally:
        self.traceOut("<ruleName>_fragment", <ruleDescriptor.index>)

<else>
    <block>
<endif>
# $ANTLR end "<ruleName>"

>>

synpred(name) ::= <<
def <name>(self):
    self._state.backtracking += 1
    <@start(>)
    start = self.input.mark()
    try:
        self.<name>_fragment()
    except BacktrackingFailed:
        success = False
    else:
        success = True
    self.input.rewind(start)

```

```

    <@stop()>
    self._state.backtracking -= 1
    return success

>>

lexerSynpred(name) ::= <<
<synpred(name)>
>>

ruleMemoization(name) ::= <<
<if(memoize)>
if self._state.backtracking > 0 and self.alreadyParsedRule(self.input, <ruleDescriptor.index>):
    # for cached failed rules, alreadyParsedRule will raise an exception
    success = True
    return <ruleReturnValue()>

<endif>
>>

/** This rule has failed, exit indicating failure during backtrack */
ruleBacktrackFailure() ::= <<
<if(backtracking)>
if self._state.backtracking > 0:
    raise BacktrackingFailed

<endif>
>>

/** How to generate code for a rule. This includes any return type
 * data aggregates required for multiple return values.
 */
rule(ruleName,ruleDescriptor,block,emptyRule,description,exceptions,finally,memoize) ::= <<
<returnScope(scope=ruleDescriptor.returnScope)>

# $ANTLR start "<ruleName>"
# <fileName>:<description>
<ruleDescriptor.actions.decorate>
def <ruleName>(self, <ruleDescriptor.parameterScope:parameterScope()>):
<if(trace)>
    self.traceIn("<ruleName>", <ruleDescriptor.index>)<\n>
<endif>
    <ruleScopeSetUp()>
    <ruleDeclarations()>
    <ruleLabelDefs()>
    <ruleDescriptor.actions.init>
    <@preamble()>

```

```

<@body><ruleBody()><@end>
<@postamble()>
return <ruleReturnValue()>

# $ANTLR end "<ruleName>"
>>

ruleBody() ::= <<
<if(memoize)>
<if(backtracking)>
success = False<\n>
<endif>
<endif>
try:
  try:
    <ruleMemoization(name=ruleName)>
    <block>
    <ruleCleanUp()>
    <(ruleDescriptor.actions.after):execAction()>

<if(memoize)>
<if(backtracking)>
  success = True<\n>
<endif>
<endif>
<if(exceptions)>
  <exceptions:{e|<catch(decl=e.decl,action=e.action)><\n>}>
<else>
<if(!emptyRule)>
<if(actions.(actionScope).rulecatch)>
  <actions.(actionScope).rulecatch>
<else>
  except RecognitionException as re:
    self.reportError(re)
    self.recover(self.input, re)
  <@setErrorReturnValue()>

<endif>
<else>
  finally:
    pass

<endif>
<endif>
finally:
<if(trace)>
  self.traceOut("<ruleName>", <ruleDescriptor.index>)<\n>
<endif>

```

```

    <memoize()>
    <ruleScopeCleanUp()>
    <finally>
    pass
>>

catch(decl,action) ::= <<
except <e.decl>:
    <e.action>

>>

ruleDeclarations() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
retval = self.<ruleDescriptor.name>_return()
retval.start = self.input.LT(1)<\n>
<elseif(ruleDescriptor.returnScope)>
<ruleDescriptor.returnScope.attributes: { a |
<a.name> = <if(a.initValue)><a.initValue><else>None<endif>
}>
<endif>
<if(memoize)>
<ruleDescriptor.name>_startIndex = self.input.index()
<endif>
>>

ruleScopeSetUp() ::= <<
<ruleDescriptor.useScopes: {it | self.<it>_stack.append(<it>_scope()); separator="\n">
<ruleDescriptor.ruleScope: {it | self.<it.name>_stack.append(<it.name>_scope()); separator="\n">
>>

ruleScopeCleanUp() ::= <<
<ruleDescriptor.useScopes: {it | self.<it>_stack.pop(); separator="\n">
<ruleDescriptor.ruleScope: {it | self.<it.name>_stack.pop(); separator="\n">
>>

ruleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,ruleDescriptor.tokenListLabels,
ruleDescriptor.wildcardTreeLabels,ruleDescriptor.wildcardTreeListLabels]
: {it | <it.label.text> = None }; separator="\n"
>
<[ruleDescriptor.tokenListLabels,ruleDescriptor.ruleListLabels,
ruleDescriptor.wildcardTreeListLabels]
: {it | list_<it.label.text> = None }; separator="\n"
>
<ruleDescriptor.ruleLabels:ruleLabelDef(); separator="\n">
<ruleDescriptor.ruleListLabels: {it | <it.label.text> = None }; separator="\n">
>>

```

```

lexerRuleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,
ruleDescriptor.tokenListLabels,
ruleDescriptor.ruleLabels]
: {it | <it.label.text> = None}; separator="\n"
>
<ruleDescriptor.charLabels: {it | <it.label.text> = None}; separator="\n">
<[ruleDescriptor.tokenListLabels,
ruleDescriptor.ruleListLabels]
: {it | list_<it.label.text> = None}; separator="\n"
>
>>

```

```

ruleReturnValue() ::= <%
<if(!ruleDescriptor.isSynPred)>
<if(ruleDescriptor.hasReturnValue)>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnName>
<else>
retval
<endif>
<endif>
<endif>
%>

```

```

ruleCleanUp() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(!TREE_PARSER)>
retval.stop = self.input.LT(-1)<\n>
<endif>
<endif>
>>

```

```

memoize() ::= <<
<if(memoize)>
<if(backtracking)>
if self._state.backtracking > 0:
self.memoize(self.input, <ruleDescriptor.index>, <ruleDescriptor.name>_startIndex, success)
<endif>
<endif>
>>

```

```

/** How to generate a rule in the lexer; naked blocks are used for
* fragment rules.
*/

```

```

lexerRule(ruleName,nakedBlock,ruleDescriptor,block,memoize) ::= <<

```

```

# $ANTLR start "<ruleName>"
def m<ruleName>(self, <ruleDescriptor.parameterScope:parameterScope()>):
<if(trace)>
    self.traceIn("<ruleName>", <ruleDescriptor.index>)<\n>
<endif>
    <ruleScopeSetUp()>
    <ruleDeclarations()>
<if(memoize)>
<if(backtracking)>
    success = False<\n>
<endif>
<endif>
    try:
<if(nakedBlock)>
        <ruleMemoization(name=ruleName)>
        <lexerRuleLabelDefs()>
        <ruleDescriptor.actions.init>
        <block><\n>
<else>
        _type = <ruleName>
        _channel = DEFAULT_CHANNEL

        <ruleMemoization(name=ruleName)>
        <lexerRuleLabelDefs()>
        <ruleDescriptor.actions.init>
        <block>
        <ruleCleanUp()>
        self._state.type = _type
        self._state.channel = _channel
        <(ruleDescriptor.actions.after):execAction()>
<endif>
<if(memoize)>
<if(backtracking)>
    success = True<\n>
<endif>
<endif>

    finally:
<if(trace)>
    self.traceOut("<ruleName>", <ruleDescriptor.index>)<\n>
<endif>
    <ruleScopeCleanUp()>
    <memoize()>
    pass

# $ANTLR end "<ruleName>"

```

```

>>

/** How to generate code for the implicitly-defined lexer grammar rule
 * that chooses between lexer rules.
 */
tokensRule(ruleName,nakedBlock,args,block,ruleDescriptor) ::= <<
def mTokens(self):
  <block><\n>

>>

// S U B R U L E S

/** A (...) subrule with multiple alternatives */
block(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
# <fileName>:<description>
alt<decisionNumber> = <maxAlt>
<decls>
<@body><blockBody()><@end>
>>

blockBody() ::= <<
<@predecision()>
<@decision><decision><@end>
<@postdecision()>
<@prebranch()>
<alts:{ a | <altSwitchCase(i, a)> }; separator="\n" >
<@postbranch()>
>>

/** A rule block with multiple alternatives */
ruleBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
# <fileName>:<description>
alt<decisionNumber> = <maxAlt>
<decls>
<@predecision()>
<@decision><decision><@end>
<@postdecision()>
<alts:{ a | <altSwitchCase(i, a)> }; separator="\n" >
>>

ruleBlockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
# <fileName>:<description>
<decls>
<@prealt()>
<alts>
<@postalt()>

```

>>

/** A special case of a (...) subrule with a single alternative */

blockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<

<fileName>:<description>

<decls>

<@prealt()>

<alts>

<@postalt()>

>>

/** A (..)+ block with 1 or more alternatives */

positiveClosureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<

<fileName>:<description>

cnt<decisionNumber> = 0

<decls>

<@preloop()>

<@loopBody>

<positiveClosureBlockLoop()>

<@end>

<@postloop()>

>>

positiveClosureBlockLoop() ::= <<

while True: #loop<decisionNumber>

 alt<decisionNumber> = <maxAlt>

 <@predecision()>

 <@decisionBody><decision><@end>

 <@postdecision()>

 <alts: { a | <altSwitchCase(i, a)> }; separator="\n" >

 else:

 if cnt<decisionNumber> >= 1:

 break #loop<decisionNumber>

 <ruleBacktrackFailure()>

 eee = EarlyExitException(<decisionNumber>, self.input)

 <@earlyExitException()>

 raise eee

 cnt<decisionNumber> += 1

>>

positiveClosureBlockSingleAlt ::= positiveClosureBlock

/** A (..)* block with 1 or more alternatives */

closureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::=

<<

```

# <fileName>:<description>
<decls>
<@preloop()>
<@loopBody>
<closureBlockLoop()>
<@end>
<@postloop()>
>>

closureBlockLoop() ::= <<
while True: #loop<decisionNumber>
  alt<decisionNumber> = <maxAlt>
  <@predecision()>
  <@decisionBody><decision><@end>
  <@postdecision()>
  <alts:{ a | <altSwitchCase(i, a)>} ; separator="\nel">
  else:
    break #loop<decisionNumber>
>>

closureBlockSingleAlt ::= closureBlock

/** Optional blocks (x)? are translated to (x|) by before code generation
 * so we can just use the normal block template
 */
optionalBlock ::= block

optionalBlockSingleAlt ::= block

/** A case in a switch that jumps to an alternative given the alternative
 * number. A DFA predicts the alternative and then a simple switch
 * does the jump to the code that actually matches that alternative.
 */
altSwitchCase(altNum,alt) ::= <<
if alt<decisionNumber> == <altNum>:
  <@prealt()>
  <alt>
>>

/** An alternative is just a list of elements; at outermost level */
alt(elements,altNum,description,autoAST,outerAlt,treeLevel,rew) ::= <<
# <fileName>:<description>
pass <! so empty alternatives are a valid block !>
<@declarations()>
<elements:element()>
<rew>
<@cleanup()>
>>

```

```

/** What to emit when there is no rewrite. For auto build
 * mode, does nothing.
 */
noRewrite(rewriteBlockLevel, treeLevel) ::= ""

// E L E M E N T S

/** Dump the elements one per line */
element(e) ::= <<
<@prematch(>
<e.el><\n>
>>

/** match a token optionally with a label in front */
tokenRef(token,label,elementIndex,terminalOptions={}) ::= <<
<if(label)><label> = <endif>self.match(self.input, <token>,
self.FOLLOW_<token>_in_<ruleName><elementIndex>)
>>

/** ids+=ID */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRef(token,label,elementIndex,terminalOptions)>
<listLabel(label, label)>
>>

listLabel(label, elem) ::= <<
if list_<label> is None:
    list_<label> = []
list_<label>.append(<elem>)<\n>
>>

/** match a character */
charRef(char,label) ::= <<
<if(label)>
<label> = self.input.LA(1)<\n>
<endif>
self.match(<char>)
>>

/** match a character range */
charRangeRef(a,b,label) ::= <<
<if(label)>
<label> = self.input.LA(1)<\n>
<endif>
self.matchRange(<a>, <b>)
>>

```

```

/** For now, sets are interval tests and must be tested inline */
matchSet(s,label,elementIndex,postmatchCode="",terminalOptions={}) ::= <<
<if(label)>
<label> = self.input.LT(1)<\n>
<endif>
if <s>:
  self.input.consume()
  <postmatchCode>
<if(!LEXER)>
  self._state.errorRecovery = False<\n>
<endif>

else:
  <ruleBacktrackFailure()>
  mse = MismatchedSetException(None, self.input)
  <@mismatchedSetException()>
<if(LEXER)>
  self.recover(mse)
  raise mse
<else>
  raise mse
  <! use following code to make it recover inline; remove throw mse;
  self.recoverFromMismatchedSet(
    self.input, mse, self.FOLLOW_set_in_<ruleName><elementIndex>
  )
  !>
<endif>
<\n>
>>

matchRuleBlockSet ::= matchSet

matchSetAndListLabel(s,label,elementIndex,postmatchCode) ::= <<
<matchSet(...)>
<listLabel(label, label)>
>>

/** Match a string literal */
lexerStringRef(string,label,elementIndex="0") ::= <<
<if(label)>
<label>Start = self.getCharIndex()
self.match(<string>)
<label>StartLine<elementIndex> = self.getLine()
<label>StartCharPos<elementIndex> = self.getCharPositionInLine()
<label> = <labelType>(input=self.input, type=INVALID_TOKEN_TYPE, channel=DEFAULT_CHANNEL,
start=<label>Start, stop=self.getCharIndex()-1)
<label>.line = <label>StartLine<elementIndex>
<label>.charPositionInLine = <label>StartCharPos<elementIndex>

```

```

<else>
self.match(<string>)
<endif>
>>

wildcard(token,label,elementIndex,terminalOptions={ }) ::= <<
<if(label)>
<label> = self.input.LT(1)<\n>
<endif>
self.matchAny()
>>

wildcardAndListLabel(token,label,elementIndex,terminalOptions={ }) ::= <<
<wildcard(...)>
<listLabel(label,label)>
>>

/** Match . wildcard in lexer */
wildcardChar(label, elementIndex) ::= <<
<if(label)>
<label> = self.input.LA(1)<\n>
<endif>
self.matchAny()
>>

wildcardCharListLabel(label, elementIndex) ::= <<
<wildcardChar(label, elementIndex)>
<listLabel(label, label)>
>>

/** Match a rule reference by invoking it possibly with arguments
 * and a return value or values. The 'rule' argument was the
 * target rule name, but now is type Rule, whose toString is
 * same: the rule name. Now though you can access full rule
 * descriptor stuff.
 */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
self._state.following.append(self.FOLLOW_<rule.name>_in_<ruleName><elementIndex>)
<if(label)><label> = <endif>self.<if(scope)><scope.delegateName()>.<endif><rule.name>(<args; separator="
">)<\n>
self._state.following.pop()
>>

/** ids+=rule */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(rule,label,elementIndex,args,scope)>
<listLabel(label, label)>
>>

```

```

/** A lexer rule reference
 * The 'rule' argument was the target rule name, but now
 * is type Rule, whose toString is same: the rule name.
 * Now though you can access full rule descriptor stuff.
 */
lexerRuleRef(rule,label,args,elementIndex,scope) ::= <<
<if(label)>
<label>Start<elementIndex> = self.getCharIndex()
self.<if(scope)><scope.delegateName().<endif>m<rule.name>(<args; separator=", ">)
<label>StartLine<elementIndex> = self.getLine()
<label>StartCharPos<elementIndex> = self.getCharPositionInLine()
<label> = <labelType>(
  input=self.input,
  type=INVALID_TOKEN_TYPE,
  channel=DEFAULT_CHANNEL,
  start=<label>Start<elementIndex>,
  stop=self.getCharIndex()-1)
<label>.line = <label>StartLine<elementIndex>
<label>.charPositionInLine = <label>StartCharPos<elementIndex>
<else>
self.<if(scope)><scope.delegateName().<endif>m<rule.name>(<args; separator=", ">)
<endif>
>>

/** i+=INT in lexer */
lexerRuleRefAndListLabel(rule,label,args,elementIndex,scope) ::= <<
<lexerRuleRef(rule,label,args,elementIndex,scope)>
<listLabel(label, label)>
>>

/** EOF in the lexer */
lexerMatchEOF(label,elementIndex) ::= <<
<if(label)>
<label>Start<elementIndex> = self.getCharIndex()
<label>StartLine<elementIndex> = self.getLine()
<label>StartCharPos<elementIndex> = self.getCharPositionInLine()
self.match(EOF)
<label> = <labelType>(input=self.input, type=EOF, channel=DEFAULT_CHANNEL,
start=<label>Start<elementIndex>, stop=self.getCharIndex()-1)
<label>.line = <label>StartLine<elementIndex>
<label>.charPositionInLine = <label>StartCharPos<elementIndex>
<else>
self.match(EOF)
<endif>
>>

// used for left-recursive rules

```

```

recRuleDefArg()          ::= "<recRuleArg()>"
recRuleArg()            ::= "_p"
recRuleAltPredicate(ruleName, opPrec) ::= "<recRuleArg()> |<= <opPrec>"
recRuleSetResultAction() ::= "root_0 = $<ruleName>_primary.tree"
recRuleSetReturnAction(src, name)  ::= "$<name> = $<src>.<name>"

/** match ^(root children) in tree parser */
tree(root, actionsAfterRoot, children, nullableChildList,
    enclosingTreeLevel, treeLevel) ::= <<
<root:element()>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if self.input.LA(1) == DOWN:
    self.match(self.input, DOWN, None)
    <children:element()>
    self.match(self.input, UP, None)

<else>
self.match(self.input, DOWN, None)
<children:element()>
self.match(self.input, UP, None)
<endif>
>>

/** Every predicate is used as a validating predicate (even when it is
 * also hoisted into a prediction expression).
 */
validateSemanticPredicate(pred,description) ::= <<
if not (<evalPredicate(pred, description)>):
    <ruleBacktrackFailure()>
    raise FailedPredicateException(self.input, "<ruleName>", "<description>")

>>

// F i x e d D F A (if-then-else)

dfaState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
LA<decisionNumber>_<stateNumber> = self.input.LA(<k>)<\n>
<edges; separator="\n" >
else:
<if(eotPredictsAlt)>
    alt<decisionNumber> = <eotPredictsAlt>
<else>
    <ruleBacktrackFailure()>
    nvae = NoViableAltException("<description>", <decisionNumber>, <stateNumber>, self.input)<\n>
    <@noViableAltException()>
    raise nvae<\n>
<endif>

```

```
>>

/** Same as a normal DFA state except that we don't examine lookahead
 * for the bypass alternative. It delays error detection but this
 * is faster, smaller, and more what people expect. For (X)? people
 * expect "if ( LA(1)==X ) match(X);" and that's it.
 */
dfaOptionalBlockState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
LA<decisionNumber>_<stateNumber> = self.input.LA(<k>)<\n>
<edges; separator="\n" >
>>
```

```
/** A DFA state that is actually the loopback decision of a closure
 * loop. If end-of-token (EOT) predicts any of the targets then it
 * should act like a default clause (i.e., no error can be generated).
 * This is used only in the lexer so that for ('a)* on the end of a rule
 * anything other than 'a' predicts exiting.
 */
dfaLoopbackState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
LA<decisionNumber>_<stateNumber> = self.input.LA(<k>)<\n>
<edges; separator="\n" ><\n>
<if(eotPredictsAlt)>
<if(!edges)>
alt<decisionNumber> = <eotPredictsAlt> <! if no edges, don't gen ELSE !>
<else>
else:
  alt<decisionNumber> = <eotPredictsAlt>
<\n>
<endif>
<endif>
>>
```

```
/** An accept state indicates a unique alternative has been predicted */
dfaAcceptState(alt) ::= "alt<decisionNumber> = <alt>"
```

```
/** A simple edge with an expression. If the expression is satisfied,
 * enter to the target state. To handle gated productions, we may
 * have to evaluate some predicates for this edge.
 */
dfaEdge(labelExpr, targetState, predicates) ::= <<
if (<labelExpr>) <if(predicates)>and (<predicates>)<endif>:
  <targetState>
>>
```

```
// F i x e d D F A (switch case)
```

```
/** A DFA state where a SWITCH may be generated. The code generator
 * decides if this is possible: CodeGenerator.canGenerateSwitch().
```

```

*/
dfaStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
<!
  FIXME: this is one of the few occasion, where I miss a switch statement
  in Python. ATM this is implemented as a list of if .. elif ..
  This may be replaced by faster a dictionary lookup, when I find a solution
  for the cases when an edge is not a plain dfaAcceptState.
!>
LA<decisionNumber> = self.input.LA(<k>)
<edges; separator="\n">
else:
<if(eotPredictsAlt)>
  alt<decisionNumber> = <eotPredictsAlt>
<else>
  <ruleBacktrackFailure()>
  nvae = NoViableAltException("<description>", <decisionNumber>, <stateNumber>, self.input)<\n>
  <@noViableAltException()>
  raise nvae<\n>
<endif>

>>

dfaOptionalBlockStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
LA<decisionNumber> = self.input.LA(<k>)
<edges; separator="\n">
>>

dfaLoopbackStateSwitch(k, edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
LA<decisionNumber> = self.input.LA(<k>)
<edges; separator="\n">
<if(eotPredictsAlt)>
else:
  alt<decisionNumber> = <eotPredictsAlt>
<endif>
>>

dfaEdgeSwitch(labels, targetState) ::= <<
if LA<decisionNumber> in {<labels; separator=",">}:
  <targetState>
>>

// C y c l i c D F A

/** The code to initiate execution of a cyclic DFA; this is used
 * in the rule to predict an alt just like the fixed DFA case.
 * The <name> attribute is inherited via the parser, lexer, ...
 */
dfaDecision(decisionNumber,description) ::= <<

```

```

alt<decisionNumber> = self.dfa<decisionNumber>.predict(self.input)
>>

/* Dump DFA tables as run-length-encoded Strings of octal values.
* Can't use hex as compiler translates them before compilation.
* These strings are split into multiple, concatenated strings.
* Java puts them back together at compile time thankfully.
* Java cannot handle large static arrays, so we're stuck with this
* encode/decode approach. See analysis and runtime DFA for
* the encoding methods.
*/
cyclicDFA(dfa) ::= <<
# lookup tables for DFA #<dfa.decisionNumber>

DFA<dfa.decisionNumber>_eot = DFA.unpack(
    "<dfa.javaCompressedEOT; wrap=\"\\n \\\">"
)

DFA<dfa.decisionNumber>_eof = DFA.unpack(
    "<dfa.javaCompressedEOF; wrap=\"\\n \\\">"
)

DFA<dfa.decisionNumber>_min = DFA.unpack(
    "<dfa.javaCompressedMin; wrap=\"\\n \\\">"
)

DFA<dfa.decisionNumber>_max = DFA.unpack(
    "<dfa.javaCompressedMax; wrap=\"\\n \\\">"
)

DFA<dfa.decisionNumber>_accept = DFA.unpack(
    "<dfa.javaCompressedAccept; wrap=\"\\n \\\">"
)

DFA<dfa.decisionNumber>_special = DFA.unpack(
    "<dfa.javaCompressedSpecial; wrap=\"\\n \\\">"
)

DFA<dfa.decisionNumber>_transition = [
    <dfa.javaCompressedTransition:{s|DFA.unpack("<s; wrap=\"\\n\\\">"); separator=",\n">
}
]

# class definition for DFA #<dfa.decisionNumber>

class DFA<dfa.decisionNumber>(DFA):
    pass

```

```

<@errorMethod()>

<if(dfa.specialStateSTs)>
def specialStateTransition(self_, s, input):
    # convince pylint that my self_ magic is ok ;)
    # pylint: disable-msg=E0213

    # pretend we are a member of the recognizer
    # thus semantic predicates can be evaluated
    self = self_.recognizer

    _s = s

    <dfa.specialStateSTs:{state | if s == <i0>: <! compressed special state numbers 0..n-1 !>
<state>} ; separator="\n!>

<if(backtracking)>
    if self._state.backtracking > 0:
        raise BacktrackingFailed

<endif>
    nvae = NoViableAltException(self_.getDescription(), <dfa.decisionNumber>, _s, input)
    self_.error(nvae)
    raise nvae<\n>
<endif>

>>

cyclicDFAInit(dfa) ::= <<
self.dfa<dfa.decisionNumber> = self.DFA<dfa.decisionNumber>(
    self, <dfa.decisionNumber>,
    eot = self.DFA<dfa.decisionNumber>_eot,
    eof = self.DFA<dfa.decisionNumber>_eof,
    min = self.DFA<dfa.decisionNumber>_min,
    max = self.DFA<dfa.decisionNumber>_max,
    accept = self.DFA<dfa.decisionNumber>_accept,
    special = self.DFA<dfa.decisionNumber>_special,
    transition = self.DFA<dfa.decisionNumber>_transition
)<\n>
>>

/** A state in a cyclic DFA; it's a special state and part of a big switch on
 * state.
 */
cyclicDFAState(decisionNumber,stateNumber,edges,needErrorClause,semPredState) ::= <<
LA<decisionNumber>_<stateNumber> = input.LA(1)<\n>
<if(semPredState)> <! get next lookahead symbol to test edges, then rewind !>
index<decisionNumber>_<stateNumber> = input.index()

```

```

input.rewind()<\n>
<endif>
s = -1
<edges; separator="\n">
<if(semPredState)> <! return input cursor to state before we rewound !>
input.seek(index<decisionNumber>_<stateNumber>)<\n>
<endif>
if s >= 0:
    return s
>>

/** Just like a fixed DFA edge, test the lookahead and indicate what
 * state to jump to next if successful.
 */
cyclicDFAEdge(labelExpr, targetStateNumber, edgeNumber, predicates) ::= <<
if (<labelExpr>)<if(predicates)> and (<predicates>)<endif>:
    s = <targetStateNumber><\n>
>>

/** An edge pointing at end-of-token; essentially matches any char;
 * always jump to the target.
 */
eotDFAEdge(targetStateNumber,edgeNumber, predicates) ::= <<
se:
    s = <targetStateNumber><\n>
>>

// D F A E X P R E S S I O N S

andPredicates(left,right) ::= "((<left>) and (<right>))"

orPredicates(operands) ::= "<operands; separator=\" or \">"

notPredicate(pred) ::= "not (<evalPredicate(pred, { })>)"

evalPredicate(pred,description) ::= "<pred>"

evalSynPredicate(pred,description) ::= "self.<pred>()"

lookaheadTest(atom,k,atomAsInt) ::= "LA<decisionNumber>_<stateNumber> == <atom>"

/** Sometimes a lookahead test cannot assume that LA(k) is in a temp variable
 * somewhere. Must ask for the lookahead directly.
 */
isolatedLookaheadTest(atom,k,atomAsInt) ::= "self.input.LA(<k>) == <atom>"

lookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= <%

```

```

(<lower> \<= LA<decisionNumber>_<stateNumber> \<= <upper>)
%>

isolatedLookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= "(<lower> \<=
self.input.LA(<k>) \<= <upper>)"

lookaheadSetTest(values,k,valuesAsInt) ::= <%
LA<decisionNumber>_<stateNumber> in {<values; separator="," ">}
%>

isolatedLookaheadSetTest(values,k,valuesAsInt) ::= <%
self.input.LA(<k>) in {<values; separator="," ">}
%>

lookaheadVarName(k) ::= "LA<decisionNumber>_<stateNumber>"
isolatedLookaheadVarName(k) ::= "self.input.LA(<k>)"

setTest(ranges) ::= "<ranges; separator=\" or \">"

// A T T R I B U T E S

globalAttributeScopeClass(scope) ::= <<
<if(scope)>
<if(scope.attributes)>
class <scope.name>_scope(object):
    def __init__(self):
        <scope.attributes:{it | self.<it.decl> = None}; separator="\n">

<endif>
<endif>
>>

globalAttributeScopeStack(scope) ::= <<
<if(scope)>
<if(scope.attributes)>
self.<scope.name>_stack = []<\n>
<endif>
<endif>
>>

ruleAttributeScopeClass(scope) ::= <<
<if(scope)>
<if(scope.attributes)>
class <scope.name>_scope(object):
    def __init__(self):
        <scope.attributes:{it | self.<it.decl> = None}; separator="\n">

<endif>

```

```

<endif>
>>

ruleAttributeScopeStack(scope) ::= <<
<if(scope)>
<if(scope.attributes)>
self.<scope.name>_stack = []<\n>
<endif>
<endif>
>>

delegateName(d) ::= <<
<if(d.label)><d.label><else>g<d.name><endif>
>>

/** Define a rule label including default value */
ruleLabelDef(label) ::= <<
<label.label.text> = None
>>

returnStructName(r) ::= "<r.name>_return"

/** Define a return struct for a rule if the code needs to access its
 * start/stop tokens, tree stuff, attributes, ... Leave a hole for
 * subgroups to stick in members.
 */
returnScope(scope) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
class <ruleDescriptor:returnStructName()>(<if(TREE_PARSER)>Tree<else>Parser<endif>RuleReturnScope):
    def __init__(self):
        super().__init__()

        <if(scope)><scope.attributes:{it | self.<it.decl> = None }; separator="\n"><endif>
        <@ruleReturnInit()>

        <@ruleReturnMembers()>

<endif>
>>

parameterScope(scope) ::= <<
<if(scope)><scope.attributes:{it | <it.decl>}; separator=", "><endif>
>>

parameterAttributeRef(attr) ::= "<attr.name>"
parameterSetAttributeRef(attr,expr) ::= "<attr.name> = <expr>"

```

```

scopeAttributeRef(scope,attr,index,negIndex) ::= <%
<if(negIndex)>
self.<scope>_stack[-<negIndex>].<attr.name>
<else>
<if(index)>
self.<scope>_stack[<index>].<attr.name>
<else>
self.<scope>_stack[-1].<attr.name>
<endif>
<endif>
%>

```

/* not applying patch because of bug in action parser!

```

<if(negIndex)>
((len(self.<scope>_stack) - <negIndex> - 1) >= 0 and [self.<scope>_stack[-<negIndex>].<attr.name>] or [None])[0]
<else>
<if(index)>
((<index> \< len(self.<scope>_stack)) and [self.<scope>_stack[<index>].<attr.name>] or [None])[0]
<else>
((len(self.<scope>_stack) > 0) and [self.<scope>_stack[-1].<attr.name>] or [None])[0]
<endif>
<endif>

```

*/

```

scopeSetAttributeRef(scope,attr,expr,index,negIndex) ::= <%
<if(negIndex)>
<!--FIXME: this seems not to be used by ActionTranslator...!-->
self.<scope>_stack[-<negIndex>].<attr.name> = <expr>
<else>
<if(index)>
<!--FIXME: this seems not to be used by ActionTranslator...!-->
self.<scope>_stack[<index>].<attr.name> = <expr>
<else>
self.<scope>_stack[-1].<attr.name> = <expr>
<endif>
<endif>
%>

```

```

/** $x is either global scope or x is rule with dynamic scope; refers
* to stack itself not top of stack. This is useful for predicates
* like {$function.size()>0 && $function::name.equals("foo")}?
*/

```

```

isolatedDynamicScopeRef(scope) ::= "self.<scope>_stack"

```

```

/** reference an attribute of rule; might only have single return value */

```

```

ruleLabelRef(referencedRule,scope,attr) ::= <%

```

```

<if(referencedRule.hasMultipleReturnValues)>
((<scope> is not None) and [<scope>.<attr.name>] or [None])[0]
<else>
<scope>
<endif>
%>

returnAttributeRef(ruleDescriptor,attr) ::= <%
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name>
<else>
<attr.name>
<endif>
%>

returnSetAttributeRef(ruleDescriptor,attr,expr) ::= <%
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name> = <expr>
<else>
<attr.name> = <expr>
<endif>
%>

/** How to translate $tokenLabel */
tokenLabelRef(label) ::= "<label>"

/** ids+=ID {$ids} or e+=expr {$e} */
listLabelRef(label) ::= "list_<label>"

// not sure the next are the right approach; and they are evaluated early;
// they cannot see TREE_PARSER or PARSER attributes for example. :(

tokenLabelPropertyRef_text(scope,attr) ::= "<scope>.text"
tokenLabelPropertyRef_type(scope,attr) ::= "<scope>.type"
tokenLabelPropertyRef_line(scope,attr) ::= "<scope>.line"
tokenLabelPropertyRef_pos(scope,attr) ::= "<scope>.charPositionInLine"
tokenLabelPropertyRef_channel(scope,attr) ::= "<scope>.channel"
tokenLabelPropertyRef_index(scope,attr) ::= "<scope>.index"
tokenLabelPropertyRef_tree(scope,attr) ::= "<scope>_tree"

ruleLabelPropertyRef_start(scope,attr) ::= "<scope>.start"
ruleLabelPropertyRef_stop(scope,attr) ::= "<scope>.stop"
ruleLabelPropertyRef_tree(scope,attr) ::= "<scope>.tree"
ruleLabelPropertyRef_text(scope,attr) ::= <%
<if(TREE_PARSER)>
((<scope> is not None) and [self.input.getTokenStream().toString(
self.input.getTreeAdaptor().getTokenStartIndex(<scope>.start),

```

```

    self.input.getTreeAdaptor().getTokenStopIndex(<scope>.start)
    ]) or [None])[0]
<else>
((<scope> is not None) and [self.input.toString(<scope>.start,<scope>.stop)] or [None])[0]
<endif>
%>
ruleLabelPropertyRef_st(scope,attr) ::= "((<scope> is not None) and [<scope>.st] or [None])[0]"

/** Isolated $RULE ref ok in lexer as it's a Token */
lexerRuleLabel(label) ::= "<label>"

lexerRuleLabelPropertyRef_type(scope,attr) ::= "((<scope> is not None) and [<scope>.type] or [0])[0]"
lexerRuleLabelPropertyRef_line(scope,attr) ::= "((<scope> is not None) and [<scope>.line] or [0])[0]"
lexerRuleLabelPropertyRef_pos(scope,attr) ::= "((<scope> is not None) and [<scope>.charPositionInLine] or [0])[0]"
lexerRuleLabelPropertyRef_channel(scope,attr) ::= "((<scope> is not None) and [<scope>.channel] or [0])[0]"
lexerRuleLabelPropertyRef_index(scope,attr) ::= "((<scope> is not None) and [<scope>.index] or [0])[0]"
lexerRuleLabelPropertyRef_text(scope,attr) ::= "((<scope> is not None) and [<scope>.text] or [None])[0]"
lexerRuleLabelPropertyRef_int(scope,attr) ::= "((<scope> is not None) and [int(<scope>.text)] or [0])[0]"

// Somebody may ref $template or $tree or $stop within a rule:
rulePropertyRef_start(scope,attr) ::= "retval.start"
rulePropertyRef_stop(scope,attr) ::= "retval.stop" //mmm... or input.LT(-1)??
rulePropertyRef_tree(scope,attr) ::= "retval.tree"
rulePropertyRef_text(scope,attr) ::= "self.input.toString(retval.start, self.input.LT(-1))"
rulePropertyRef_st(scope,attr) ::= "retval.st"

lexerRulePropertyRef_text(scope,attr) ::= "self.text"
lexerRulePropertyRef_type(scope,attr) ::= "_type"
lexerRulePropertyRef_line(scope,attr) ::= "self._state.tokenStartLine"
lexerRulePropertyRef_pos(scope,attr) ::= "self._state.tokenStartCharPositionInLine"
lexerRulePropertyRef_index(scope,attr) ::= "-1" // undefined token index in lexer
lexerRulePropertyRef_channel(scope,attr) ::= "_channel"
lexerRulePropertyRef_start(scope,attr) ::= "self._state.tokenStartCharIndex"
lexerRulePropertyRef_stop(scope,attr) ::= "(self.getCharIndex()-1)"
lexerRulePropertyRef_int(scope,attr) ::= "int(<scope>.text)"

// setting $st and $tree is allowed in local rule. everything else
// is flagged as error
ruleSetPropertyRef_tree(scope,attr,expr) ::= "retval.tree =<expr>"
ruleSetPropertyRef_st(scope,attr,expr) ::= "retval.st =<expr>"

/** How to execute an action (only when not backtracking) */
execAction(action) ::= <<
<if(backtracking)>
<if(actions.(actionScope).synpredgate)>
if <actions.(actionScope).synpredgate>:

```

```

    pass
    <action>

<else>
if <actions.(actionScope).synpredgate>:
    pass
    <action>

<endif>
<else>
#action start
<action>
#action end
<endif>
>>

/** How to always execute an action even when backtracking */
execForcedAction(action) ::= "<action>"

// M I S C (properties, etc...)

codeFileExtension() ::= ".py"

true_value() ::= "True"
false_value() ::= "False"

Found in path(s):
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-
jar/org/antlr/codegen/templates/Python3/Python3.stg
No license file was found, but licenses were detected in source scan.

/*
* [The "BSD license"]
* Copyright (c) 2011 Terence Parr
* All rights reserved.
*
* Conversion to C#:
* Copyright (c) 2011 Sam Harwell, Pixel Mine, Inc.
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the

```

- * documentation and/or other materials provided with the distribution.
- * 3. The name of the author may not be used to endorse or promote products
- * derived from this software without specific prior written permission.
- *
- * THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
- * IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
- * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
- * IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
- * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
- * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
- * DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
- * THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
- * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
- * THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
- */

/** Templates for building ASTs during normal parsing.

- *
- * Deal with many combinations. Dimensions are:
- * Auto build or rewrite
- * no label, label, list label (label/no-label handled together)
- * child, root
- * token, set, rule, wildcard
- *
- * The situation is not too bad as rewrite (->) usage makes ^ and !
- * invalid. There is no huge explosion of combinations.
- */

```
@rule.setErrorReturnValue() ::= <<
retval.Tree = (<ASTLabelType>)adaptor.ErrorNode(input, retval.Start, input.LT(-1), re);
<! System.out.WriteLine("<ruleName> returns "+((CommonTree)retval.tree).toStringTree()); !>
>>
```

// TOKEN AST STUFF

```
/** ID and output=AST */
tokenRef(token,label,elementIndex,terminalOptions={}) ::= <%
<super.tokenRef(...)>
<if(!ruleDescriptor.isSynPred)>
<if(backtracking)><\n>if (state.backtracking == 0) {<endif>
<\n><label>_tree = <createNodeFromToken(...)>;
<\n>adaptor.AddChild(root_0, <label>_tree);
<if(backtracking)><\n>}<endif>
<endif>
%>
```

```
/** ID! and output=AST (same as plain tokenRef) */
tokenRefBang(token,label,elementIndex,terminalOptions={}) ::= "<super.tokenRef(...)>"
```

```

/** ID^ and output=AST */
tokenRefRuleRoot(token,label,elementIndex,terminalOptions={ }) ::= <%
<super.tokenRef(...)>
<if(!ruleDescriptor.isSynPred)>
<if(backtracking)><\n>if (<actions.(actionScope).synpredgate>) {<endif>
<\n><label>_tree = <createNodeFromToken(...)>;
<\n>root_0 = (<ASTLabelType>)adaptor.BecomeRoot(<label>_tree, root_0);
<if(backtracking)><\n>}<endif>
<endif>
%>

/** ids+=ID! and output=AST */
tokenRefBangAndListLabel(token,label,elementIndex,terminalOptions={ }) ::= <<
<tokenRefBang(...)>
<listLabelElem(elem=label,elemType=labelType,...)>
>>

/** label+=TOKEN when output=AST but not rewrite alt */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions={ }) ::= <<
<tokenRef(...)>
<listLabelElem(elem=label,elemType=labelType,...)>
>>

/** Match label+=TOKEN^ when output=AST but not rewrite alt */
tokenRefRuleRootAndListLabel(token,label,elementIndex,terminalOptions={ }) ::= <<
<tokenRefRuleRoot(...)>
<listLabelElem(elem=label,elemType=labelType,...)>
>>

// SET AST

// the match set stuff is interesting in that it uses an argument list
// to pass code to the default matchSet; another possible way to alter
// inherited code. I don't use the region stuff because I need to pass
// different chunks depending on the operator. I don't like making
// the template name have the operator as the number of templates gets
// large but this is the most flexible--this is as opposed to having
// the code generator call matchSet then add root code or ruleroot code
// plus list label plus ... The combinations might require complicated
// rather than just added on code. Investigate that refactoring when
// I have more time.

matchSet(s,label,elementIndex,postmatchCode,terminalOptions={ }) ::= <<
<super.matchSet(postmatchCode={<if(!ruleDescriptor.isSynPred)><if(backtracking)>if
(<actions.(actionScope).synpredgate>) <endif>adaptor.AddChild(root_0, <createNodeFromToken(...)>);<endif>},
...)>
>>

```

```

matchRuleBlockSet(s,label,elementIndex,postmatchCode,treeLevel="0",terminalOptions={}) ::= <<
<matchSet(...)>
>>

matchSetBang(s,label,elementIndex,postmatchCode,terminalOptions={}) ::= "<super.matchSet(...)>"

// note there is no matchSetTrack because -> rewrites force sets to be
// plain old blocks of alts: (A|B|...|C)

matchSetRuleRoot(s,label,elementIndex,debug,terminalOptions={}) ::= <<
<if(label)>
<label>=(<labelType>)input.LT(1);
<endif>
<super.matchSet(postmatchCode={ <if(!ruleDescriptor.isSynPred)><if(backtracking)>if
(<actions.(actionScope).synpredgate>) <endif>root_0 =
(<ASTLabelType>)adaptor.BecomeRoot(<createNodeFromToken(...)>, root_0);<endif>}, ...)>
>>

// RULE REF AST

/** rule when output=AST */
ruleRef(rule,label,elementIndex,args,scope) ::= <%
<super.ruleRef(...)>
<if(!ruleDescriptor.isSynPred)>
<\n><if(backtracking)>if (<actions.(actionScope).synpredgate>) <endif>adaptor.AddChild(root_0, <label>.Tree);
<endif>
%>

/** rule! is same as normal rule ref */
ruleRefBang(rule,label,elementIndex,args,scope) ::= "<super.ruleRef(...)>"

/** rule^ */
ruleRefRuleRoot(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) <endif>root_0 =
(<ASTLabelType>)adaptor.BecomeRoot(<label>.Tree, root_0);
>>

/** x+=rule when output=AST */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabelElem(elem={<label>.Tree},elemType=ASTLabelType,...)>
>>

/** x+=rule! when output=AST is a rule ref with list addition */
ruleRefBangAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefBang(...)>

```

```

<listLabelElem(elem={<label>.Tree},elemType=ASTLabelType,...)>
>>

/** x+=rule^ */
ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<listLabelElem(elem={<label>.Tree},elemType=ASTLabelType,...)>
>>

// WILDCARD AST

wildcard(token,label,elementIndex,terminalOptions={}) ::= <<
<super.wildcard(...)>
<if(!ruleDescriptor.isSynPred)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) {<endif>
<label>_tree = (<ASTLabelType>)adaptor.Create(<label>);
adaptor.AddChild(root_0, <label>_tree);
<if(backtracking)>}<endif>
<endif>
>>

wildcardBang(label,elementIndex) ::= "<super.wildcard(token=[],...)>"

wildcardRuleRoot(token,label,elementIndex,terminalOptions={}) ::= <<
<super.wildcard(...)>
<if(!ruleDescriptor.isSynPred)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) {<endif>
<label>_tree = (<ASTLabelType>)adaptor.Create(<label>);
root_0 = (<ASTLabelType>)adaptor.BecomeRoot(<label>_tree, root_0);
<if(backtracking)>}<endif>
<endif>
>>

createNodeFromToken(label,terminalOptions={}) ::= <%
<if(terminalOptions.node)>
new
<terminalOptions.node><(if(terminalOptions.type)><terminalOptions.type>,<endif><label><(if(terminalOptions.text)>,<terminalOptions.text; format="string"><endif>)
<else>
(<ASTLabelType>)adaptor.Create(<(if(terminalOptions.type)><terminalOptions.type>,<endif><label><(if(terminalOptions.text)>,<terminalOptions.text; format="string"><endif>)
<endif>
%>

ruleCleanup() ::= <<
<super.ruleCleanup()>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) {<endif>
retval.Tree = (<ASTLabelType>)adaptor.RulePostProcessing(root_0);

```

```
adaptor.SetTokenBoundaries(retval.Tree, retval.Start, retval.Stop);
<if(backtracking)><endif>
>>
```

Found in path(s):

```
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-
jar/org/antlr/codegen/templates/CSharp3/ASTParser.stg
No license file was found, but licenses were detected in source scan.
```

```
/*
[The "BSD license"]
Copyright (c) 2005-2009 Jim Idle, Temporal Wave LLC
http://www.temporal-wave.com
http://www.linkedin.com/in/jimidle
```

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```
*/
```

```
/*
* This code generating template and the associated C runtime was produced by:
* Jim Idle jim|hereisanat|idle|dotgoeshere|ws.
* If it causes the destruction of the Universe, it will be pretty cool so long as
* I am in a different one at the time.
```

```
*/
```

```
cTypeInitMap ::= [
    "int" : "0", // Integers start out being 0
```

```

"long"   : "0",           // Longs   start out being 0
"float"  : "0.0",        // Floats  start out being 0
"double" : "0.0",        // Doubles start out being 0
"ANTLR3_BOOLEAN" : "ANTLR3_FALSE", // Booleans start out being Antlr C for false
"byte"   : "0",           // Bytes   start out being 0
"short"  : "0",           // Shorts  start out being 0
"char"   : "0"           // Chars   start out being 0
]

```

```
leadIn(type) ::=
```

```
<<
```

```
/** \file
```

```
* This <type> file was generated by $ANTLR version <ANTLRVersion>
```

```
*
```

```
* - From the grammar source file : <fileName>
```

```
* - On : <generatedTimestamp>
```

```
<if(LEXER)>
```

```
* - for the lexer : <name>Lexer
```

```
<endif>
```

```
<if(PARSER)>
```

```
* - for the parser : <name>Parser
```

```
<endif>
```

```
<if(TREE_PARSER)>
```

```
* - for the tree parser : <name>TreeParser
```

```
<endif>
```

```
*
```

```
* Editing it, at least manually, is not wise.
```

```
*
```

```
* C language generator and runtime by Jim Idle, jim@hereisanat[idle|dotgoeshere]ws.
```

```
*
```

```
*
```

```
>>
```

```
/** The overall file structure of a recognizer; stores methods for rules
```

```
* and cyclic DFAs plus support code.
```

```
*/
```

```
outputFile( LEXER,
```

```
    PARSER,
```

```
    TREE_PARSER,
```

```
    actionScope,
```

```
    actions,
```

```
    docComment,
```

```
    recognizer,
```

```
    name,
```

```
    tokens,
```

```
    tokenNames,
```

```
    rules,
```

```
    cyclicDFAs,
```

```

    bitsets,
    buildTemplate,
    buildAST,
    rewriteMode,
    profile,
    backtracking,
    synpreds,
    memoize,
    numRules,
    fileName,
    ANTLRVersion,
    generatedTimestamp,
    trace,
    scopes,
    superClass,
    literals
) ::=
<<
<leadIn("C source")>
*/
// [The "BSD license"]
// Copyright (c) 2005-2009 Jim Idle, Temporal Wave LLC
// http://www.temporal-wave.com
// http://www.linkedin.com/in/jimidle
//
// All rights reserved.
//
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions
// are met:
// 1. Redistributions of source code must retain the above copyright
//    notice, this list of conditions and the following disclaimer.
// 2. Redistributions in binary form must reproduce the above copyright
//    notice, this list of conditions and the following disclaimer in the
//    documentation and/or other materials provided with the distribution.
// 3. The name of the author may not be used to endorse or promote products
//    derived from this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
// IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
// OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
// IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
// INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
// NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
// THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```

```

<if(actions.(actionScope).header)>

/* =====
* This is what the grammar programmer asked us to put at the top of every file.
*/
<actions.(actionScope).header>
/* End of Header action.
* =====
*/
<endif>

/* -----
* Include the ANTLR3 generated header file.
*/
#include "<name>.h"
<actions.(actionScope).postinclude>
/* ----- */

<docComment>

<if(literals)>
/** String literals used by <name> that we must do things like MATCHS() with.
* C will normally just lay down 8 bit characters, and you can use L"xxx" to
* get wchar_t, but wchar_t is 16 bits on Windows, which is not UTF32 and so
* we perform this little trick of defining the literals as arrays of UINT32
* and passing in the address of these.
*/
<literals:{it | static ANTLR3_UCHAR lit_<i>[] = <it>;}; separator="\n">

<endif>

/* MACROS that hide the C interface implementations from the
* generated code, which makes it a little more understandable to the human eye.
* I am very much against using C pre-processor macros for function calls and bits
* of code as you cannot see what is happening when single stepping in debuggers
* and so on. The exception (in my book at least) is for generated code, where you are
* not maintaining it, but may wish to read and understand it. If you single step it, you know that input()
* hides some indirect calls, but is always referring to the input stream. This is
* probably more readable than ctx->input->istream->input(snarfle0->blarg) and allows me to rejig
* the runtime interfaces without changing the generated code too often, without
* confusing the reader of the generated output, who may not wish to know the gory
* details of the interface inheritance.
*/

```

```

#define CTX ctx

/* Aids in accessing scopes for grammar programmers
*/
#undef SCOPE_TYPE
#undef SCOPE_STACK
#undef SCOPE_TOP
#define SCOPE_TYPE(scope) p<name>_##scope##_SCOPE
#define SCOPE_STACK(scope) p<name>_##scope##_Stack
#define SCOPE_TOP(scope) ctx->p<name>_##scope##_Top
#define SCOPE_SIZE(scope) ctx->p<name>_##scope##_Stack_limit
#define SCOPE_INSTANCE(scope, i) (ctx->SCOPE_STACK(scope)->get(ctx->SCOPE_STACK(scope),i))

<if(LEXER)>

/* Macros for accessing things in a lexer
*/
#undef LEXER
#undef RECOGNIZER
#undef RULEMEMO
#undef GETCHARINDEX
#undef GETLINE
#undef GETCHARPOSITIONINLINE
#undef EMIT
#undef EMITNEW
#undef MATCHC
#undef MATCHS
#undef MATCHRANGE
#undef LTOKEN
#undef HASFAILED
#undef FAILEDFLAG
#undef INPUT
#undef STRSTREAM
#undef LA
#undef HASEXCEPTION
#undef EXCEPTION
#undef CONSTRUCTEX
#undef CONSUME
#undef LRECOVER
#undef MARK
#undef REWIND
#undef REWINDLAST
#undef BACKTRACKING
#undef MATCHANY
#undef MEMOIZE
#undef HAVEPARSEDRULE
#undef GETTEXT
#undef INDEX

```

```

#undef SEEK
#undef PUSHSTREAM
#undef POPSTREAM
#undef SETTEXT
#undef SETTEXT8

#define LEXER ctx->pLexer
#define RECOGNIZER LEXER->rec
#define LEXSTATE RECOGNIZER->state
#define TOKSOURCE LEXSTATE->tokSource
#define GETCHARINDEX() LEXER->getCharIndex(LEXER)
#define GETLINE() LEXER->getLine(LEXER)
#define GETTEXT() LEXER->getText(LEXER)
#define GETCHARPOSITIONINLINE() LEXER->getCharPositionInLine(LEXER)
#define EMIT() LEXSTATE->type = _type; LEXER->emit(LEXER)
#define EMITNEW(t) LEXER->emitNew(LEXER, t)
#define MATCHC(c) LEXER->matchc(LEXER, c)
#define MATCHS(s) LEXER->matchs(LEXER, s)
#define MATCHRANGE(c1,c2) LEXER->matchRange(LEXER, c1, c2)
#define MATCHANY() LEXER->matchAny(LEXER)
#define LTOKEN LEXSTATE->token
#define HASFAILED() (LEXSTATE->failed == ANTLR3_TRUE)
#define BACKTRACKING LEXSTATE->backtracking
#define FAILEDFLAG LEXSTATE->failed
#define INPUT LEXER->input
#define STRSTREAM INPUT
#define ISTREAM INPUT->istream
#define INDEX() ISTREAM->index(ISTREAM)
#define SEEK(n) ISTREAM->seek(ISTREAM, n)
#define EOF_TOKEN &(LEXSTATE->tokSource->eofToken)
#define HASEXCEPTION() (LEXSTATE->error == ANTLR3_TRUE)
#define EXCEPTION LEXSTATE->exception
#define CONSTRUCTEX() RECOGNIZER->exConstruct(RECOGNIZER)
#define LRECOVER() LEXER->recover(LEXER)
#define MARK() ISTREAM->mark(ISTREAM)
#define REWIND(m) ISTREAM->rewind(ISTREAM, m)
#define REWINDLAST() ISTREAM->rewindLast(ISTREAM)
#define MEMOIZE(ri,si) RECOGNIZER->memoize(RECOGNIZER, ri, si)
#define HAVEPARSEDRULE(r) RECOGNIZER->alreadyParsedRule(RECOGNIZER, r)
#define PUSHSTREAM(str) LEXER->pushCharStream(LEXER, str)
#define POPSTREAM() LEXER->popCharStream(LEXER)
#define SETTEXT(str) LEXSTATE->text = str
#define SKIP() LEXSTATE->token = &(TOKSOURCE->skipToken)
#define USER1 LEXSTATE->user1
#define USER2 LEXSTATE->user2
#define USER3 LEXSTATE->user3
#define CUSTOM LEXSTATE->custom
#define RULEMEMO LEXSTATE->ruleMemo

```

```

#define DBG RECOGNIZER->debugger

/* If we have been told we can rely on the standard 8 bit or UTF16 input
 * stream, then we can define our macros to use the direct pointers
 * in the input object, which is much faster than indirect calls. This
 * is really only significant to lexers with a lot of fragment rules (which
 * do not place LA(1) in a temporary at the moment) and even then
 * only if there is a lot of input (order of say 1M or so).
 */
#if defined(ANTLR3_INLINE_INPUT_8BIT) || defined(ANTLR3_INLINE_INPUT_UTF16)

# ifdef ANTLR3_INLINE_INPUT_8BIT

/* 8 bit character set */

# define NEXTCHAR ((pANTLR3_UINT8)(INPUT->nextChar))
# define DATAP ((pANTLR3_UINT8)(INPUT->data))

# else

# define NEXTCHAR ((pANTLR3_UINT16)(INPUT->nextChar))
# define DATAP ((pANTLR3_UINT16)(INPUT->data))

# endif

# define LA(n) ((NEXTCHAR + n) > (DATAP + INPUT->sizeBuf) ? ANTLR3_CHARSTREAM_EOF :
(ANTLR3_UCHAR)*(NEXTCHAR + n - 1))
# define CONSUME() \
{ \
    if (NEXTCHAR < (DATAP + INPUT->sizeBuf)) \
    { \
        INPUT->charPositionInLine++; \
        if ((ANTLR3_UCHAR)*NEXTCHAR == INPUT->newlineChar) \
        { \
            INPUT->line++; \
            INPUT->charPositionInLine = 0; \
            INPUT->currentLine = (void *) (NEXTCHAR + 1); \
        } \
        INPUT->nextChar = (void *) (NEXTCHAR + 1); \
    } \
}

# else

// Pick up the input character by calling the input stream implementation.
//
# define CONSUME() INPUT->istream->consume(INPUT->istream)
# define LA(n) INPUT->istream->_LA(INPUT->istream, n)

```

```

#endif
<endif>

<if(PARSER)>
/* Macros for accessing things in the parser
*/

#undef  PARSE
#undef  RECOGNIZER
#undef  HAVEPARSEDRULE
#undef  MEMOIZE
#undef  INPUT
#undef  STRSTREAM
#undef  HASEXCEPTION
#undef  EXCEPTION
#undef  MATCHT
#undef  MATCHANYT
#undef  FOLLOWSTACK
#undef  FOLLOWPUSH
#undef  FOLLOWPOP
#undef  PRECOVER
#undef  PREPORTERROR
#undef  LA
#undef  LT
#undef  CONSTRUCTEX
#undef  CONSUME
#undef  MARK
#undef  REWIND
#undef  REWINDLAST
#undef  PERRORRECOVERY
#undef  HASFAILED
#undef  FAILEDFLAG
#undef  RECOVERFROMMISMATCHEDSET
#undef  RECOVERFROMMISMATCHEDELEMENT
#undef  INDEX
#undef  ADAPTOR
#undef  SEEK
#undef  RULEMEMO
#undef  DBG

#define  PARSE  ctx->pParser
#define  RECOGNIZER  PARSE->rec
#define  PSRSTATE  RECOGNIZER->state
#define  HAVEPARSEDRULE(r)  RECOGNIZER->alreadyParsedRule(RECOGNIZER, r)
#define  MEMOIZE(ri,si)  RECOGNIZER->memoize(RECOGNIZER, ri, si)
#define  INPUT  PARSE->tstream
#define  STRSTREAM  INPUT

```

```

#define ISTREAM INPUT->istream
#define INDEX() ISTREAM->index(INPUT->istream)
#define HASEXCEPTION() (PSRSTATE->error == ANTLR3_TRUE)
#define EXCEPTION PSRSTATE->exception
#define MATCHT(t, fs) RECOGNIZER->match(RECOGNIZER, t, fs)
#define MATCHANYT() RECOGNIZER->matchAny(RECOGNIZER)
#define FOLLOWSTACK PSRSTATE->following
#ifdef SKIP_FOLLOW_SETS
#define FOLLOWPUSH(x)
#define FOLLOWPOP()
#else
#define FOLLOWPUSH(x) FOLLOWSTACK->push(FOLLOWSTACK, ((void *)&(x)), NULL)
#define FOLLOWPOP() FOLLOWSTACK->pop(FOLLOWSTACK)
#endif
#define PRECOVER() RECOGNIZER->recover(RECOGNIZER)
#define PREPORTERROR() RECOGNIZER->reportError(RECOGNIZER)
#define LA(n) INPUT->istream->_LA(ISTREAM, n)
#define LT(n) INPUT->_LT(INPUT, n)
#define CONSTRUCTEX() RECOGNIZER->exConstruct(RECOGNIZER)
#define CONSUME() ISTREAM->consume(ISTREAM)
#define MARK() ISTREAM->mark(ISTREAM)
#define REWIND(m) ISTREAM->rewind(ISTREAM, m)
#define REWINDLAST() ISTREAM->rewindLast(ISTREAM)
#define SEEK(n) ISTREAM->seek(ISTREAM, n)
#define PERORRECOVERY PSRSTATE->errorRecovery
#define FAILEDFLAG PSRSTATE->failed
#define HASFAILED() (FAILEDFLAG == ANTLR3_TRUE)
#define BACKTRACKING PSRSTATE->backtracking
#define RECOVERFROMMISMATCHEDSET(s) RECOGNIZER->recoverFromMismatchedSet(RECOGNIZER,
s)
#define RECOVERFROMMISMATCHEDELEMENT(e) RECOGNIZER-
>recoverFromMismatchedElement(RECOGNIZER, s)
#define ADAPTOR ctx->adaptor
#define RULEMEMO PSRSTATE->ruleMemo
#define DBG RECOGNIZER->debugger

<endif>

<if(TREE_PARSER)>
/* Macros for accessing things in the parser
*/

#undef PARSE
#undef RECOGNIZER
#undef HAVEPARSEDRULE
#undef INPUT
#undef STRSTREAM
#undef HASEXCEPTION

```

```

#undef EXCEPTION
#undef MATCHT
#undef MATCHANYT
#undef FOLLOWSTACK
#undef FOLLOWPUSH
#undef FOLLOWPOP
#undef PRECOVER
#undef PREPORTERROR
#undef LA
#undef LT
#undef CONSTRUCTEX
#undef CONSUME
#undef MARK
#undef REWIND
#undef REWINDLAST
#undef PERORRECOVERY
#undef HASFAILED
#undef FAILEDFLAG
#undef RECOVERFROMMISMATCHEDSET
#undef RECOVERFROMMISMATCHEDELEMENT
#undef BACKTRACKING
#undef ADAPTOR
#undef RULEMEMO
#undef SEEK
#undef INDEX
#undef DBG

#define PARSER    ctx->pTreeParser
#define RECOGNIZER  PARSER->rec
#define PSRSTATE   RECOGNIZER->state
#define HAVEPARSEDRULE(r)  RECOGNIZER->alreadyParsedRule(RECOGNIZER, r)
#define INPUT     PARSER->ctnstream
#define ISTREAM   INPUT->tnstream->istream
#define STRSTREAM INPUT->tnstream
#define HASEXCEPTION()  (PSRSTATE->error == ANTLR3_TRUE)
#define EXCEPTION    PSRSTATE->exception
#define MATCHT(t, fs)  RECOGNIZER->match(RECOGNIZER, t, fs)
#define MATCHANYT()   RECOGNIZER->matchAny(RECOGNIZER)
#define FOLLOWSTACK   PSRSTATE->following
#define FOLLOWPUSH(x) FOLLOWSTACK->push(FOLLOWSTACK, ((void *)&(x)), NULL)
#define FOLLOWPOP()   FOLLOWSTACK->pop(FOLLOWSTACK)
#define PRECOVER()    RECOGNIZER->recover(RECOGNIZER)
#define PREPORTERROR() RECOGNIZER->reportError(RECOGNIZER)
#define LA(n)         ISTREAM->_LA(ISTREAM, n)
#define LT(n)         INPUT->tnstream->_LT(INPUT->tnstream, n)
#define CONSTRUCTEX() RECOGNIZER->exConstruct(RECOGNIZER)
#define CONSUME()     ISTREAM->consume(ISTREAM)
#define MARK()        ISTREAM->mark(ISTREAM)

```

```

#define REWIND(m) ISTREAM->rewind(ISTREAM, m)
#define REWINDLAST() ISTREAM->rewindLast(ISTREAM)
#define PERRORRECOVERY PSRSTATE->errorRecovery
#define FAILEDFLAG PSRSTATE->failed
#define HASFAILED() (FAILEDFLAG == ANTLR3_TRUE)
#define BACKTRACKING PSRSTATE->backtracking
#define RECOVERFROMMISMATCHEDSET(s) RECOGNIZER->recoverFromMismatchedSet(RECOGNIZER,
s)
#define RECOVERFROMMISMATCHEDELEMENT(e) RECOGNIZER-
>recoverFromMismatchedElement(RECOGNIZER, s)
#define ADAPTOR INPUT->adaptor
#define RULEMEMO PSRSTATE->ruleMemo
#define SEEK(n) ISTREAM->seek(ISTREAM, n)
#define INDEX() ISTREAM->index(ISTREAM)
#define DBG RECOGNIZER->debugger

```

```
<endif>
```

```
#define TOKTEXT(tok, txt) tok, (pANTLR3_UINT8)txt
```

```

/* The 4 tokens defined below may well clash with your own #defines or token types. If so
* then for the present you must use different names for your defines as these are hard coded
* in the code generator. It would be better not to use such names internally, and maybe
* we can change this in a forthcoming release. I deliberately do not #undef these
* here as this will at least give you a redefined error somewhere if they clash.
*/

```

```

#define UP ANTLR3_TOKEN_UP
#define DOWN ANTLR3_TOKEN_DOWN
#define EOR ANTLR3_TOKEN_EOR
#define INVALID ANTLR3_TOKEN_INVALID

```

```
/* =====
```

```

* Functions to create and destroy scopes. First come the rule scopes, followed
* by the global declared scopes.
*/

```

```

<rules: {r |<if(r.ruleDescriptor.ruleScope)>
<ruleAttributeScopeFuncDecl(scope=r.ruleDescriptor.ruleScope)>
<ruleAttributeScopeFuncs(scope=r.ruleDescriptor.ruleScope)>
<endif>}>

```

```

<recognizer.scopes: {it | <if(it.isDynamicGlobalScope)>
<globalAttributeScopeFuncDecl(it)>
<globalAttributeScopeFuncs(it)>
<endif>}>

```

```

/* ===== */

/* =====
* Start of recognizer
*/

<recognizer>

/* End of code
* =====
*/

>>
headerFileExtension() ::= ".h"

headerFile( LEXER,
            PARSER,
            TREE_PARSER,
            actionScope,
            actions,
            docComment,
            recognizer,
            name,
            tokens,
            tokenNames,
            rules,
            cyclicDFAs,
            bitsets,
            buildTemplate,
            buildAST,
            rewriteMode,
            profile,
            backtracking,
            synpreds,
            memoize,
            numRules,
            fileName,
            ANTLRVersion,
            generatedTimestamp,
            trace,
            scopes,
            superClass,
            literals
            ) ::=

<<
<leadIn("C header")>
<if(PARSER)>
* The parser <mainName()>

```

```

<endif>
<if(LEXER)>
* The lexer <mainName()>
<endif>
<if(TREE_PARSER)>
* The tree parser <mainName()>
<endif>
has the callable functions (rules) shown below,
* which will invoke the code for the associated rule in the source grammar
* assuming that the input stream is pointing to a token/text stream that could begin
* this rule.
*
* For instance if you call the first (topmost) rule in a parser grammar, you will
* get the results of a full parse, but calling a rule half way through the grammar will
* allow you to pass part of a full token stream to the parser, such as for syntax checking
* in editors and so on.
*
* The parser entry points are called indirectly (by function pointer to function) via
* a parser context typedef p<name>, which is returned from a call to <name>New().
*
<if(LEXER)>
* As this is a generated lexer, it is unlikely you will call it 'manually'. However
* the methods are provided anyway.
*
<endif>
* The methods in p<name> are as follows:
*
* <rules:{r | <if(!r.ruleDescriptor.isSynPred)> - <headerReturnType(ruleDescriptor=r.ruleDescriptor,...)>
p<name>-><r.ruleDescriptor.name>(p<name><endif>}; separator="\n * ">
*
* The return type for any particular rule is of course determined by the source
* grammar file.
*/
// [The "BSD license"]
// Copyright (c) 2005-2009 Jim Idle, Temporal Wave LLC
// http://www.temporal-wave.com
// http://www.linkedin.com/in/jimidle
//
// All rights reserved.
//
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions
// are met:
// 1. Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// 2. Redistributions in binary form must reproduce the above copyright
// notice, this list of conditions and the following disclaimer in the
// documentation and/or other materials provided with the distribution.

```

```
// 3. The name of the author may not be used to endorse or promote products
// derived from this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
// IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
// OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
// IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
// INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
// NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
// THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

```
#ifndef _<name>_H
#define _<name>_H
<actions.(actionScope).preincludes>
/* =====
* Standard antlr3 C runtime definitions
*/
#include <antlr3.h>

/* End of standard antlr 3 runtime definitions
* =====
*/
<actions.(actionScope).includes>

#ifdef __cplusplus
extern "C" {
#endif

// Forward declare the context typedef so that we can use it before it is
// properly defined. Delegates and delegates (from import statements) are
// interdependent and their context structures contain pointers to each other
// C only allows such things to be declared if you pre-declare the typedef.
//
typedef struct <name>_Ctx_struct <name>, * p<name>;

<if(recognizer.grammar.delegates)>
// Include delegate definition header files
//
<recognizer.grammar.delegates: {g|#include \<<g.recognizerName>.h>}; separator="\n">

<endif>

<actions.(actionScope).header>
```

```

#ifdef ANTLR3_WINDOWS
// Disable: Unreferenced parameter, - Rules with parameters that are not used
// constant conditional, - ANTLR realizes that a prediction is always true (synpred usually)
// initialized but unused variable - tree rewrite variables declared but not needed
// Unreferenced local variable - lexer rule declares but does not always use _type
// potentially uninitialized variable used - retval always returned from a rule
// unreferenced local function has been removed - susually getTokenNames or freeScope, they can go without
warnigns
//
// These are only really displayed at warning level /W4 but that is the code ideal I am aiming at
// and the codegen must generate some of these warnings by necessity, apart from 4100, which is
// usually generated when a parser rule is given a parameter that it does not use. Mostly though
// this is a matter of orthogonality hence I disable that one.
//
#pragma warning( disable : 4100 )
#pragma warning( disable : 4101 )
#pragma warning( disable : 4127 )
#pragma warning( disable : 4189 )
#pragma warning( disable : 4505 )
#pragma warning( disable : 4701 )
#endif
<if(backtracking)>

/* =====
* BACKTRACKING IS ENABLED
* =====
*/
<endif>

<rules:{r |<headerReturnScope(ruleDescriptor=r.ruleDescriptor,...)>}>

<scopes:{it | <if(it.isDynamicGlobalScope)><globalAttributeScopeDecl(it)><endif>}>
<rules:{r |<ruleAttributeScopeDecl(scope=r.ruleDescriptor.ruleScope)>}>
<if(recognizer.grammar.delegators)>
// Include delegator definition header files
//
<recognizer.grammar.delegators: {g##include \<<g.recognizerName>.h}; separator="\n">

<endif>

/** Context tracking structure for <mainName()>
*/
struct <name>_Ctx_struct
{
/** Built in ANTLR3 context tracker contains all the generic elements
* required for context tracking.
*/
<if(PARSER)>

```

```

    pANTLR3_PARSER pParser;
<endif>
<if(LEXER)>
    pANTLR3_LEXER pLexer;
<endif>
<if(TREE_PARSER)>
    pANTLR3_TREE_PARSER pTreeParser;
<endif>

<if(recognizer.grammar.delegates)>
    <recognizer.grammar.delegates:
        {g|p<g.recognizerName> <g.delegateName(>}; separator="\n">
<endif>
<if(recognizer.grammar.delegators)>
    <recognizer.grammar.delegators:
        {g|p<g.recognizerName> <g.delegateName(>}; separator="\n">
<endif>
<scopes:{it | <if(it.isDynamicGlobalScope)>
    <globalAttributeScopeDef(it)>
<endif>} ; separator="\n\n">
<rules: {r |<if(r.ruleDescriptor.ruleScope)>
    <ruleAttributeScopeDef(scope=r.ruleDescriptor.ruleScope)>
<endif>}>

<if(LEXER)>
    <rules:{r | <if(!r.ruleDescriptor.isSynPred)><headerReturnType(ruleDescriptor=r.ruleDescriptor)>
(*m<r.ruleDescriptor.name>) (struct <name>_Ctx_struct * ctx<if(r.ruleDescriptor.parameterScope)>,
<endif><r.ruleDescriptor.parameterScope:parameterScope(>>);<endif>} ; separator="\n">
<endif>
<if(!LEXER)>
    <rules:{r | <headerReturnType(ruleDescriptor=r.ruleDescriptor)> (*<r.ruleDescriptor.name>) (struct
<name>_Ctx_struct * ctx<if(r.ruleDescriptor.parameterScope)>,
<endif><r.ruleDescriptor.parameterScope:parameterScope(>>); ; separator="\n">
<! generate rule/method definitions for imported rules so they
appear to be defined in this recognizer. !>
    // Delegated rules
<recognizer.grammar.delegatedRules:{ruleDescriptor|
    <headerReturnType(ruleDescriptor)> (*<ruleDescriptor.name>)(struct <name>_Ctx_struct *
ctx<if(ruleDescriptor.parameterScope)>, <endif><ruleDescriptor.parameterScope:parameterScope(>>); ;
separator="\n">
<endif>

const char * (*getGrammarFileName());
void (*reset) (struct <name>_Ctx_struct * ctx);
void (*free) (struct <name>_Ctx_struct * ctx);
<@members>
<@end>
<actions.(actionScope).context>

```

```

};

// Function prototypes for the constructor functions that external translation units
// such as delegators and delegates may wish to call.
//
ANTLR3_API p<name> <name>New      (<inputType()> instream<recognizer.grammar.delegators:{g|,
p<g.recognizerName> <g:delegateName()>>>);
ANTLR3_API p<name> <name>NewSSD   (<inputType()> instream,
pANTLR3_RECOGNIZER_SHARED_STATE state<recognizer.grammar.delegators:{g|, p<g.recognizerName>
<g:delegateName()>>>);
<if(!recognizer.grammar.grammarIsRoot)>
extern pANTLR3_UINT8 <recognizer.grammar.composite.rootGrammar.recognizerName>TokenNames[];
<endif>

/** Symbolic definitions of all the tokens that the <grammarType()> will work with.
* \{
*
* Antlr will define EOF, but we can't use that as it is too common in
* in C header files and that would be confusing. There is no way to filter this out at the moment
* so we just undef it here for now. That isn't the value we get back from C recognizers
* anyway. We are looking for ANTLR3_TOKEN_EOF.
*/
#ifdef EOF
#undef EOF
#endif
#ifdef Tokens
#undef Tokens
#endif
<tokens:{it | #define <it.name>    <it.type>}; separator="\n">
#ifdef EOF
#undef EOF
#define EOF ANTLR3_TOKEN_EOF
#endif

#ifdef TOKENSOURCE
#define TOKENSOURCE(lxr) lxr->pLexer->rec->state->tokSource
#endif

/* End of token definitions for <name>
* =====
*/
/** \} */

#ifdef __cplusplus
}
#endif

```

```

#endif

/* END - Note:Keep extra line feed to satisfy UNIX systems */

>>

inputType() ::= <<
<if(LEXER)>
pANTLR3_INPUT_STREAM
<endif>
<if(PARSER)>
pANTLR3_COMMON_TOKEN_STREAM
<endif>
<if(TREE_PARSER)>
pANTLR3_COMMON_TREE_NODE_STREAM
<endif>
>>

grammarType() ::= <<
<if(PARSER)>
parser
<endif>
<if(LEXER)>
lexer
<endif>
<if(TREE_PARSER)>
tree parser
<endif>
>>

mainName() ::= <<
<if(PARSER)>
<name>
<endif>
<if(LEXER)>
<name>
<endif>
<if(TREE_PARSER)>
<name>
<endif>
>>

headerReturnScope(ruleDescriptor) ::= "<returnScope(...)>"

headerReturnType(ruleDescriptor) ::= <<
<if(LEXER)>
<if(!ruleDescriptor.isSynPred)>
void

```

```

<else>
<returnType()>
<endif>
<else>
<returnType()>
<endif>
>>

// Produce the lexer output
//
lexer( grammar,
      name,
        tokens,
        scopes,
        rules,
        numRules,
        filterMode,
        superClass,
        labelType="pANTLR3_COMMON_TOKEN") ::= <<

<if(filterMode)>
/* Forward declare implementation function for ANTLR3_TOKEN_SOURCE interface when
* this is a filter mode lexer.
*/
static pANTLR3_COMMON_TOKEN <name>NextToken (pANTLR3_TOKEN_SOURCE toksource);

/* Override the normal MEMOIZE and HAVEALREADYPARSED macros as this is a filtering
* lexer. In filter mode, the memoizing and backtracking are gated at BACKTRACKING > 1 rather
* than just BACKTRACKING. In some cases this might generate code akin to:
* if (BACKTRACKING) if (BACKTRACKING > 1) memoize.
* However, I assume that the C compilers/optimizers are smart enough to work this one out
* these days - Jim
*/
#undef MEMOIZE
#define MEMOIZE(ri,si) if (BACKTRACKING>1) { RECOGNIZER->memoize(RECOGNIZER, ri, si) }
#undef HAVEPARSEDRULE
#define HAVEPARSEDRULE(r) if (BACKTRACKING>1) { RECOGNIZER->alreadyParsedRule(RECOGNIZER,
r) }
<endif>

/* Forward declare the locally static matching functions we have generated and any predicate functions.
*/
<rules:{r | static ANTLR3_INLINE <headerReturnType(ruleDescriptor=r.ruleDescriptor)>
<if(!r.ruleDescriptor.isSynPred)>m<endif><r.ruleDescriptor.name> (p<name>
ctx<if(r.ruleDescriptor.parameterScope)>, <endif><r.ruleDescriptor.parameterScope:parameterScope()>);};
separator="\n">
static void <name>Free(p<name> ctx);

```

```

/* =====
 * Lexer matching rules end.
 * =====
 */

```

```

<scopes:{it |<if(it.isDynamicGlobalScope)><globalAttributeScope(it)><endif>}>

```

```

<actions.lexer.members>

```

```

static void
<name>Free (p<name> ctx)
{
<if(memoize)>
if (RULEMEMO != NULL)
{
RULEMEMO->free(RULEMEMO);
RULEMEMO = NULL;
}
<endif>
<if(grammar.directDelegates)>
// Free the lexers that we delegated to
// functions to. NULL the state so we only free it once.
//
<grammar.directDelegates:
    {g|ctx-><g:delegateName()->pLexer->rec->state = NULL;
    ctx-><g:delegateName()->free(ctx-><g:delegateName()->);}; separator="\n">
<endif>
    LEXER->free(LEXER);

    ANTLR3_FREE(ctx);
}

```

```

static void
<name>Reset (p<name> ctx)
{
    RECOGNIZER->reset(RECOGNIZER);
}

```

```

/** \brief Name of the grammar file that generated this code
 */

```

```

static const char fileName[] = "<fileName>";

```

```

/** \brief Return the name of the grammar file that generated this code.
 */

```

```

static const char * getGrammarFileName()
{
    return fileName;
}

```

```

<if(filterMode)>
  <filteringNextToken()>
<endif>

/** \brief Create a new lexer called <name>
 *
 * \param[in] instream Pointer to an initialized input stream
 * \return
 * - Success p<name> initialized for the lex start
 * - Fail NULL
 */
ANTLR3_API p<name> <name>New
(<inputType()> instream<grammar.delegators:{g|, p<g.recognizerName> <g.delegateName()>>>)
{
  // See if we can create a new lexer with the standard constructor
  //
  return <name>NewSSD(instream, NULL<grammar.delegators:{g|, <g.delegateName()>>>);
}

/** \brief Create a new lexer called <name>
 *
 * \param[in] instream Pointer to an initialized input stream
 * \param[state] state Previously created shared recognizer stat
 * \return
 * - Success p<name> initialized for the lex start
 * - Fail NULL
 */
ANTLR3_API p<name> <name>NewSSD
(pANTLR3_INPUT_STREAM instream, pANTLR3_RECOGNIZER_SHARED_STATE
state<grammar.delegators:{g|, p<g.recognizerName> <g.delegateName()>>>)
{
  p<name> ctx; // Context structure we will build and return

  ctx = (p<name>) ANTLR3_CALLOC(1, sizeof(<name>));

  if (ctx == NULL)
  {
    // Failed to allocate memory for lexer context
    return NULL;
  }

  /* -----
   * Memory for basic structure is allocated, now to fill in
   * in base ANTLR3 structures. We initialize the function pointers
   * for the standard ANTLR3 lexer function set, but upon return
   * from here, the programmer may set the pointers to provide custom
   * implementations of each function.

```

```

*
* We don't use the macros defined in <name>.h here so you can get a sense
* of what goes where.
*/

/* Create a base lexer, using the supplied input stream
*/
ctx->pLexer = antlr3LexerNewStream(ANTLR3_SIZE_HINT, instream, state);

/* Check that we allocated the memory correctly
*/
if (ctx->pLexer == NULL)
{
    ANTLR3_FREE(ctx);
    return NULL;
}
<if(memoize)>
<if(grammar.grammarIsRoot)>
    // Create a LIST for recording rule memos.
    //
    ctx->pLexer->rec->ruleMemo = antlr3IntTrieNew(15); /* 16 bit depth is enough for 32768 rules! */
<endif>
<endif>

/* Install the implementation of our <name> interface
*/
<rules:{r | <if(!r.ruleDescriptor.isSynPred)>ctx->m<r.ruleDescriptor.name> =
m<r.ruleDescriptor.name>;<endif>} separator="\n">

/** When the nextToken() call is made to this lexer's pANTLR3_TOKEN_SOURCE
* it will call mTokens() in this generated code, and will pass it the ctx
* pointer of this lexer, not the context of the base lexer, so store that now.
*/
ctx->pLexer->ctx = ctx;

/**Install the token matching function
*/
ctx->pLexer->mTokens = (void (*)(void *))(mTokens);

ctx->getGrammarFileName = getGrammarFileName;
ctx->free = <name>Free;
ctx->reset = <name>Reset;

<if(grammar.directDelegates)>
// Initialize the lexers that we are going to delegate some
// functions to.
//
<grammar.directDelegates:

```

```

    {g|ctx-><g:delegateName()> = <g.recognizerName>NewSSD(instream, ctx->pLexer->rec->state,
ctx<grammar.delegators:{g|, <g:delegateName()>>}); separator="\n">
<endif>
<if(grammar.delegators)>
// Install the pointers back to lexers that will delegate us to perform certain functions
// for them.
//
<grammar.delegators:
    {g|ctx-><g:delegateName()> = <g:delegateName()>;}; separator="\n">
<endif>
<if(filterMode)>
/* We have filter mode turned on, so install the filtering nextToken function
*/
ctx->pLexer->rec->state->tokSource->nextToken = <name>NextToken;
<endif>
<actions.lexer.apifuncs>

/* Return the newly built lexer to the caller
*/
return ctx;
}
<if(cyclicDFAs)>

/* =====
* DFA tables for the lexer
*/
<cyclicDFAs:cyclicDFA()> <! dump tables for all DFA !>
/* =====
* End of DFA tables for the lexer
*/
<endif>

/* =====
* Functions to match the lexer grammar defined tokens from the input stream
*/

<rules; separator="\n\n">

/* =====
* Lexer matching rules end.
* =====
*/
<if(synpreds)>

/* =====
* Lexer syntactic predicates
*/
<synpreds:{p | <lexerSynpred(predname=p)>>}>

```

```

/* =====
* Lexer syntactic predicates end.
* =====
*/
<endif>

/* End of Lexer code
* =====
* =====
*/

>>

```

```

filteringNextToken() ::= <<

```

```

/** An override of the lexer's nextToken() method that backtracks over mTokens() looking
* for matches in lexer filterMode. No error can be generated upon error; just rewind, consume
* a token and then try again. BACKTRACKING needs to be set as well.
* Make rule memoization happen only at levels above 1 as we start mTokens
* at BACKTRACKING==1.
*/

```

```

static pANTLR3_COMMON_TOKEN
<name>NextToken(pANTLR3_TOKEN_SOURCE toksource)
{
    pANTLR3_LEXER lexer;
    pANTLR3_RECOGNIZER_SHARED_STATE state;

```

```

    lexer = (pANTLR3_LEXER)(toksource->super);
    state = lexer->rec->state;

```

```

    /* Get rid of any previous token (token factory takes care of
    * any deallocation when this token is finally used up.
    */
    state ->token = NULL;
    state ->error = ANTLR3_FALSE; /* Start out without an exception */
    state ->failed = ANTLR3_FALSE;

```

```

    /* Record the start of the token in our input stream.
    */
    state->tokenStartCharIndex = lexer->input->istream->index(lexer->input->istream);
    state->tokenStartCharPositionInLine = lexer->input->getCharPositionInLine(lexer->input);
    state->tokenStartLine = lexer->input->getLine(lexer->input);
    state->text = NULL;

```

```

    /* Now call the matching rules and see if we can generate a new token
    */
    for (;;)
    {

```



```

/** How to generate a parser */
genericParser( grammar,
    name,
        scopes,
        tokens,
        tokenNames,
        rules,
        numRules,
        bitsets,
        inputStreamType,
        superClass,
        labelType,
    members,
    rewriteElementType, filterMode,
        ASTLabelType="pANTLR3_BASE_TREE"
    ) ::= <<

<if(grammar.grammarIsRoot)>
/** \brief Table of all token names in symbolic order, mainly used for
*   error reporting.
*/
pANTLR3_UINT8 <name>TokenNames[<length(tokenNames)>+4]
= {
    (pANTLR3_UINT8) "\<invalid>", /* String to print to indicate an invalid token */
    (pANTLR3_UINT8) "\<EOR>",
    (pANTLR3_UINT8) "\<DOWN>",
    (pANTLR3_UINT8) "\<UP>",
    <tokenNames:{it |(pANTLR3_UINT8) <it>; separator=",\n">
    };
<endif>

<@members>

<@end>
<rules:{r |<ruleAttributeScopeFuncMacro(scope=r.ruleDescriptor.ruleScope)>}>
<scopes:{it |<if(it.isDynamicGlobalScope)><globalAttributeScopeFuncMacro(it)><endif>}>

// Forward declare the locally static matching functions we have generated.
//
<rules:{r | static <headerReturnType(ruleDescriptor=r.ruleDescriptor)> <r.ruleDescriptor.name> (p<name>
ctx<if(r.ruleDescriptor.parameterScope)>, <endif><r.ruleDescriptor.parameterScope:parameterScope()>);};
separator="\n">
static void <name>Free(p<name> ctx);
static void <name>Reset (p<name> ctx);

<if(!LEXER)>

```

```

<! generate rule/method definitions for imported rules so they
  appear to be defined in this recognizer. !>
<if(recognizer.grammar.delegatedRules)>
// Delegated rules
//
<recognizer.grammar.delegatedRules:{ruleDescriptor|static <headerReturnType(ruleDescriptor)>
<ruleDescriptor.name>(p<name> ctx<if(ruleDescriptor.parameterScope)>,
<endif><ruleDescriptor.parameterScope:parameterScope(>>); separator="\n">

<endif>
<endif>

/* For use in tree output where we are accumulating rule labels via label += ruleRef
 * we need a function that knows how to free a return scope when the list is destroyed.
 * We cannot just use ANTLR3_FREE because in debug tracking mode, this is a macro.
 */
static void ANTLR3_CDECL freeScope(void * scope)
{
    ANTLR3_FREE(scope);
}

/** \brief Name of the grammar file that generated this code
 */
static const char fileName[] = "<fileName>";

/** \brief Return the name of the grammar file that generated this code.
 */
static const char * getGrammarFileName()
{
    return fileName;
}

/** \brief Create a new <name> parser and return a context for it.
 *
 * \param[in] instream Pointer to an input stream interface.
 *
 * \return Pointer to new parser context upon success.
 */
ANTLR3_API p<name>
<name>New (<inputStreamType> instream<grammar.delegators:{g|, p<g.recognizerName>
<g:delegateName(>>)>
{
    // See if we can create a new parser with the standard constructor
    //
    return <name>NewSSD(instream, NULL<grammar.delegators:{g|, <g:delegateName(>>)>);
}

/** \brief Create a new <name> parser and return a context for it.
 */

```

```

* \param[in] instream Pointer to an input stream interface.
*
* \return Pointer to new parser context upon success.
*/
ANTLR3_API p<name>
<name>NewSSD (<inputStreamType> instream, pANTLR3_RECOGNIZER_SHARED_STATE
state<grammar.delegators:{g|, p<g.recognizerName> <g.delegateName()>}>)
{
    p<name> ctx;    /* Context structure we will build and return */

    ctx = (p<name>) ANTLR3_CALLOC(1, sizeof(<name>));

    if (ctx == NULL)
    {
        // Failed to allocate memory for parser context
        //
        return NULL;
    }

    /* -----
    * Memory for basic structure is allocated, now to fill in
    * the base ANTLR3 structures. We initialize the function pointers
    * for the standard ANTLR3 parser function set, but upon return
    * from here, the programmer may set the pointers to provide custom
    * implementations of each function.
    *
    * We don't use the macros defined in <name>.h here, in order that you can get a sense
    * of what goes where.
    */

<if(PARSER)>
    /* Create a base parser/recognizer, using the supplied token stream
    */
    ctx->pParser = antlr3ParserNewStream(ANTLR3_SIZE_HINT, instream->tstream, state);
<endif>
<if(TREE_PARSER)>
    /* Create a base Tree parser/recognizer, using the supplied tree node stream
    */
    ctx->pTreeParser = antlr3TreeParserNewStream(ANTLR3_SIZE_HINT, instream, state);
<endif>

    /* Install the implementation of our <name> interface
    */
    <rules:{r | ctx-><r.ruleDescriptor.name> = <r.ruleDescriptor.name>;}; separator="\n">
<if(grammar.delegatedRules)>
    // Install the delegated methods so that they appear to be a part of this
    // parser
    //

```

```

    <grammar.delegatedRules:{ruleDescriptor | ctx-><ruleDescriptor.name> = <ruleDescriptor.name>;};
separator="\n">
<endif>

ctx->free = <name>Free;
ctx->reset = <name>Reset;
ctx->getGrammarFileName = getGrammarFileName;

/* Install the scope pushing methods.
*/
<rules: {r |<if(r.ruleDescriptor.ruleScope)>
<ruleAttributeScope(scope=r.ruleDescriptor.ruleScope)><\n>
<endif>}>
<recognizer.scopes:{it |<if(it.isDynamicGlobalScope)>
<globalAttributeScope(it)><\n>
<endif>}>
<@apifuncs>

<@end>
<if(grammar.directDelegates)>
// Initialize the parsers that we are going to delegate some
// functions to.
//
<grammar.directDelegates:
    {g|ctx-><g.delegateName()> = <g.recognizerName>NewSSD(instream, PSRSTATE,
ctx-><grammar.delegators:{g|, <g.delegateName()>}>);}; separator="\n">
<endif>
<if(grammar.delegators)>
// Install the pointers back to parsers that will delegate us to perform certain functions
// for them.
//
<grammar.delegators:
    {g|ctx-><g.delegateName()> = <g.delegateName()>;}; separator="\n">
<endif>
<actions.parser.apifuncs>
<actions.treeparser.apifuncs>
<if(memoize)>
<if(grammar.grammarIsRoot)>
/* Create a LIST for recording rule memos.
*/
RULEMEMO = antlr3IntTrieNew(15); /* 16 bit depth is enough for 32768 rules! */<\n>
<endif>
<endif>
/* Install the token table
*/
PSRSTATE->tokenNames = <grammar.composite.rootGrammar.recognizerName>TokenNames;

<@debugStuff()>

```

```

    /* Return the newly built parser to the caller
    */
    return ctx;
}

static void
<name>Reset (p<name> ctx)
{
    RECOGNIZER->reset(RECOGNIZER);
}

/** Free the parser resources
*/
static void
<name>Free(p<name> ctx)
{
    /* Free any scope memory
    */
    <rules: {r
|<if(r.ruleDescriptor.ruleScope)><ruleAttributeScopeFree(scope=r.ruleDescriptor.ruleScope)><\n><endif> }>
    <recognizer.scopes: {it |<if(it.isDynamicGlobalScope)><globalAttributeScopeFree(it)><\n><endif> }>

    <@cleanup>
    <@end>
    <if(grammar.directDelegates)>
    // Free the parsers that we delegated to
    // functions to.NULL the state so we only free it once.
    //
    <grammar.directDelegates:
        {g| ctx-><g:delegateName()-><if(TREE_PARSER)>pTreeParser<else>pParser<endif>->rec->state = NULL;
        ctx-><g:delegateName()->free(ctx-><g:delegateName()->);}; separator="\n">
    <endif>
    <if(memoize)>
    <if(grammar.grammarIsRoot)>
    if (RULEMEMO != NULL)
    {
        RULEMEMO->free(RULEMEMO);
        RULEMEMO = NULL;
    }
    <endif>
    <endif>
    // Free this parser
    //
    <if(TREE_PARSER)>
        ctx->pTreeParser->free(ctx->pTreeParser);<\n>
    <else>
        ctx->pParser->free(ctx->pParser);<\n>

```

```

<endif>

    ANTLR3_FREE(ctx);

    /* Everything is released, so we can return
    */
    return;
}

/** Return token names used by this <grammarType()>
*
* The returned pointer is used as an index into the token names table (using the token
* number as the index).
*
* \return Pointer to first char * in the table.
*/
static pANTLR3_UINT8 *getTokenNames()
{
    return <grammar.composite.rootGrammar.recognizerName>TokenNames;
}

<members>

/* Declare the bitsets
*/
<bitsets:{it | <bitsetDeclare(name={FOLLOW_<it.name>_in_<it.inName><it.tokenIndex>},
    words64=it.bits)>}>

<if(cyclicDFAs)>

/* =====
* DFA tables for the parser
*/
<cyclicDFAs:cyclicDFA()> <! dump tables for all DFA !>
/* =====
* End of DFA tables for the parser
*/
<endif>

/* =====
* Parsing rules
*/
<rules; separator="\n\n">
<if(grammar.delegatedRules)>
// Delegated methods that appear to be a part of this
// parser
//

```

```

<grammar.delegatedRules:{ruleDescriptor|
  <returnType()> <ruleDescriptor.name>(p<name> ctx<if(ruleDescriptor.parameterScope.attributes)>,
<endif><ruleDescriptor.parameterScope:parameterScope()>
  \{
    <if(ruleDescriptor.hasReturnValue)>return <endif>ctx-><ruleDescriptor.grammar:delegateName()>-
  ><ruleDescriptor.name>(ctx-
  ><ruleDescriptor.grammar:delegateName()><if(ruleDescriptor.parameterScope.attributes)>,
  <endif><ruleDescriptor.parameterScope.attributes:{ a|<a.name>}; separator=", ">);
  \}); separator="\n">

<endif>
/* End of parsing rules
* =====
*/

/* =====
* Syntactic predicates
*/
<synpreds:{ p | <synpred(predname=p)>>
/* End of syntactic predicates
* =====
*/

>>

parser( grammar,
  name,
  scopes,
  tokens,
  tokenNames,
  rules,
  numRules,
  bitsets,
  ASTLabelType,
  superClass="Parser",
  labelType="pANTLR3_COMMON_TOKEN",
  members={ <actions.parser.members> }
) ::= <<
<genericParser(inputStreamType="pANTLR3_COMMON_TOKEN_STREAM", rewriteElementType="TOKEN",
...)>
>>

/** How to generate a tree parser; same as parser except the input
* stream is a different type.

```

```

*/
treeParser( grammar,
    name,
    scopes,
    tokens,
    tokenNames,
    globalAction,
    rules,
    numRules,
    bitsets,
    filterMode,
    labelType={<ASTLabelType>},
    ASTLabelType="pANTLR3_BASE_TREE",
    superClass="TreeParser",
    members={<actions.treeparser.members>}
) ::= <<
<genericParser(inputStreamType="pANTLR3_COMMON_TREE_NODE_STREAM",
rewriteElementType="NODE", ...)>
>>

/** A simpler version of a rule template that is specific to the imaginary
 * rules created for syntactic predicates. As they never have return values
 * nor parameters etc..., just give simplest possible method. Don't do
 * any of the normal memoization stuff in here either; it's a waste.
 * As predicates cannot be inlined into the invoking rule, they need to
 * be in a rule by themselves.
 */
synpredRule(ruleName, ruleDescriptor, block, description, nakedBlock) ::=
<<
// $ANTLR start <ruleName>
static void <ruleName>_fragment(p<name> ctx <ruleDescriptor.parameterScope:parameterScope(>
{
<ruleLabelDefs(>
<ruleLabelInitializations(>
<if(trace)>
    ANTLR3_PRINTF("enter <ruleName> %d failed = %d, backtracking = %d\n",LT(1),failed,BACKTRACKING);
    <block>
    ANTLR3_PRINTF("exit <ruleName> %d, failed = %d, backtracking = %d\n",LT(1),failed,BACKTRACKING);

<else>
    <block>
<endif>
<ruleCleanUp(>
}
// $ANTLR end <ruleName>
>>

synpred(predname) ::= <<

```

```

static ANTLR3_BOOLEAN <predname>(p<name> ctx)
{
    ANTLR3_MARKER start;
    ANTLR3_BOOLEAN success;

    BACKTRACKING++;
    <@start()>
    start = MARK();
    <predname>_fragment(ctx); // can never throw exception
    success = !(FAILEDFLAG);
    REWIND(start);
    <@stop()>
    BACKTRACKING--;
    FAILEDFLAG = ANTLR3_FALSE;
    return success;
}<\n>
>>

lexerSynpred(predname) ::= <<
<synpred(predname)>
>>

ruleMemoization(rname) ::= <<
<if(memoize)>
if ( (BACKTRACKING>0) && (HAVEPARSEDRULE(<ruleDescriptor.index>)) )
{
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(!ruleDescriptor.isSynPred)>
retval.start = 0;<\n>
<endif>
<endif>
<(ruleDescriptor.actions.after):execAfter()>
<finalCode(finalBlock=finally)>
<if(!ruleDescriptor.isSynPred)>
<scopeClean()><\n>
<endif>
return <ruleReturnValue()>;
}
<endif>
>>

/** How to test for failure and return from rule */
checkRuleBacktrackFailure() ::= <<
if (HASEXCEPTION())
{
goto rule<ruleDescriptor.name>Ex;
}
<if(backtracking)>

```

```

if (HASFAILED())
{
    <scopeClean()>
    <@debugClean()>
    return <ruleReturnValue()>;
}
<endif>
>>

/** This rule has failed, exit indicating failure during backtrack */
ruleBacktrackFailure() ::= <<
<if(backtracking)>
if (BACKTRACKING>0)
{
    FAILEDFLAG = <>true_value()>;
    <scopeClean()>
    return <ruleReturnValue()>;
}
<endif>
>>

/** How to generate code for a rule. This includes any return type
 * data aggregates required for multiple return values.
 */
rule(ruleName,ruleDescriptor,block,emptyRule,description,exceptions,finally,memoize) ::= <<
/**
 * $ANTLR start <ruleName>
 * <fileName>:<description>
 */
static <returnType()>
<ruleName>(p<name> ctx<if(ruleDescriptor.parameterScope)>,
<endif><ruleDescriptor.parameterScope:parameterScope()>)
{
    <if(trace)>ANTLR3_PRINTF("enter <ruleName> %s failed=%d, backtracking=%d\n", LT(1),
BACKTRACKING);<endif>
    <ruleDeclarations()>
    <ruleDescriptor.actions.declarations>
    <ruleLabelDefs()>
    <ruleInitializations()>
    <ruleDescriptor.actions.init>
    <ruleMemoization(rname=ruleName)>
    <ruleLabelInitializations()>
    <@preamble()>
    {
        <block>
    }

    <ruleCleanUp()>

```

```

<if(exceptions)>
  if (HASEXCEPTION())
  {
<exceptions: {e|<catch(decl=e.decl,action=e.action)><\n> }>
  }
  else
  {
<(ruleDescriptor.actions.after):execAfter()>
  }
<else>
  <if(!emptyRule)>
    <if(actions.(actionScope).rulecatch)>
      <actions.(actionScope).rulecatch>
    <else>
      if (HASEXCEPTION())
      {
        PREPORTERROR();
        PRECOVER();
        <@setErrorReturnValue()>
      }
      <if(ruleDescriptor.actions.after)>
      else
      {
        <(ruleDescriptor.actions.after):execAfter()>
      }<\n>
      <endif>
    <endif>
  <endif>
<endif>

  <if(trace)>ANTLR3_PRINTF("exit <ruleName> %d failed=%s backtracking=%s\n", LT(1), failed,
BACKTRACKING);<endif>
  <memoize()>
<if(finally)>
  <finalCode(finalBlock=finally)>
<endif>
  <scopeClean()>
  <@postamble()>
  return <ruleReturnValue()>;
}
/* $ANTLR end <ruleName> */
>>

finalCode(finalBlock) ::= <<
{
  <finalBlock>
}

```

>>

```
catch(decl,action) ::= <<
/* catch(decl,action)
*/
{
  <e.action>
}

```

>>

```
ruleDeclarations() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<returnType()> retval;<\n>
<else>
<ruleDescriptor.returnScope.attributes:{ a |
<a.type> <a.name>;
}>
<endif>
<if(memoize)>
ANTLR3_MARKER <ruleDescriptor.name>_startIndex;
<endif>
>>

```

```
ruleInitializations() ::= <<
/* Initialize rule variables
*/
<if(ruleDescriptor.hasMultipleReturnValues)>
<ruleDescriptor.returnScope.attributes:{ a |
<if(a.initValue)>retval.<a.name> = <a.initValue>;<endif>
}>
<else>
<ruleDescriptor.returnScope.attributes:{ a |
<if(a.initValue)><a.name> = <a.initValue>;<endif>
}>
<endif>
<if(memoize)>
<ruleDescriptor.name>_startIndex = INDEX();<\n>
<endif>
<ruleDescriptor.useScopes:{it |<scopeTop(it)> = <scopePush(it)>;}; separator="\n">
<ruleDescriptor.ruleScope:{it |<scopeTop(it.name)> = <scopePush(it.name)>;}; separator="\n">
>>

```

```
ruleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,ruleDescriptor.tokenListLabels]
: {it |<labelType> <it.label.text>;}; separator="\n"
>
<[ruleDescriptor.tokenListLabels,ruleDescriptor.ruleListLabels]

```

```

: {it |pANTLR3_VECTOR list_<it.label.text>;}; separator="\n"
>
<[ruleDescriptor.ruleLabels,ruleDescriptor.ruleListLabels]
:ruleLabelDef(); separator="\n"
>
>
>>

ruleLabelInitializations() ::= <<
<[ruleDescriptor.tokenLabels,ruleDescriptor.tokenListLabels]
: {it |<it.label.text> = NULL;}; separator="\n"
>
<[ruleDescriptor.tokenListLabels,ruleDescriptor.ruleListLabels]
: {it |list_<it.label.text> = NULL;}; separator="\n"
>
<[ruleDescriptor.ruleLabels,ruleDescriptor.ruleListLabels]
:ruleLabelInitVal(); separator="\n"
>
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(!ruleDescriptor.isSynPred)>
retval.start = LT(1); retval.stop = retval.start;<\n>
<endif>
<endif>
>>

lexerRuleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,
ruleDescriptor.tokenListLabels,
ruleDescriptor.ruleLabels]
: {it |<labelType> <it.label.text>;}; separator="\n"
>
<ruleDescriptor.charLabels: {it |ANTLR3_UINT32 <it.label.text>;}; separator="\n">
<[ruleDescriptor.tokenListLabels,
ruleDescriptor.ruleListLabels,
ruleDescriptor.ruleListLabels]
: {it |pANTLR3_INT_TRIE list_<it.label.text>;}; separator="\n"
>
>>

lexerRuleLabelInit() ::= <<
<[ruleDescriptor.tokenLabels,
ruleDescriptor.tokenListLabels,
ruleDescriptor.ruleLabels]
: {it |<it.label.text> = NULL;}; separator="\n"
>
<[ruleDescriptor.tokenListLabels,
ruleDescriptor.ruleListLabels,
ruleDescriptor.ruleListLabels]
: {it |list_<it.label.text> = antlr3IntTrieNew(31);}; separator="\n"

```

```
>
>>
```

```
lexerRuleLabelFree() ::= <<
<[ruleDescriptor.tokenLabels,
ruleDescriptor.tokenListLabels,
ruleDescriptor.ruleLabels]
: {it | <it.label.text> = NULL; }; separator="\n"
>
<[ruleDescriptor.tokenListLabels,
ruleDescriptor.ruleListLabels,
ruleDescriptor.ruleListLabels]
: {it | list_<it.label.text>->free(list_<it.label.text>); }; separator="\n"
>
>>
```

```
ruleReturnValue() ::= <%
<if(!ruleDescriptor.isSynPred)>
<if(ruleDescriptor.hasReturnValue)>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnName>
<else>
retval
<endif>
<endif>
<endif>
%>
```

```
memoize() ::= <<
<if(memoize)>
<if(backtracking)>
if ( BACKTRACKING>0 ) { MEMOIZE(<ruleDescriptor.index>, <ruleDescriptor.name>_startIndex); }
<endif>
<endif>
>>
```

```
ruleCleanup() ::= <<

// This is where rules clean up and exit
//
goto rule<ruleDescriptor.name>Ex; /* Prevent compiler warnings */
rule<ruleDescriptor.name>Ex; ;
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(!TREE_PARSER)>
<if(!ruleDescriptor.isSynPred)>
retval.stop = LT(-1);<\n>
<endif>
<endif>
```

```

<endif>
>>

scopeClean() ::= <<
<ruleDescriptor.useScopes:{it |<scopePop(it)>}; separator="\n">
<ruleDescriptor.ruleScope:{it |<scopePop(it.name)>}; separator="\n">

>>
/** How to generate a rule in the lexer; naked blocks are used for
 * fragment rules, which do not produce tokens.
 */
lexerRule(ruleName,nakedBlock,ruleDescriptor,block,memoize) ::= <<
// Comes from: <block.description>
/** \brief Lexer rule generated by ANTLR3
 *
 * $ANTLR start <ruleName>
 *
 * Looks to match the characters the constitute the token <ruleName>
 * from the attached input stream.
 *
 *
 * \remark
 * - lexer->error == ANTLR3_TRUE if an exception was thrown.
 */
static ANTLR3_INLINE
void m<ruleName>(p<name> ctx<if(ruleDescriptor.parameterScope)>,
<endif><ruleDescriptor.parameterScope:parameterScope()>)
{
  ANTLR3_UINT32 _type;
  <ruleDeclarations()>
  <ruleDescriptor.actions.declarations>
  <lexerRuleLabelDefs()>
  <if(trace)>System.out.println("enter <ruleName> "+(char)LA(1)+"
line="+GETLINE()+": "+GETCHARPOSITIONINLINE()+ " failed="+failed+"
backtracking="+BACKTRACKING);<endif>

<if(nakedBlock)>
  <ruleMemoization(rname=ruleName)>
  <lexerRuleLabelInit()>
  <ruleDescriptor.actions.init>

  <block><\n>
<else>
  <ruleMemoization(rname=ruleName)>
  <lexerRuleLabelInit()>
  _type = <ruleName>;

  <ruleDescriptor.actions.init>

```

```

    <block>
    LEXSTATE->type = _type;
<endif>
    <if(trace)> ANTLR3_FPRINTF(stderr, "exit <ruleName> '%c' line=%d:%d failed = %d, backtracking
    =%d\n",LA(1),GETLINE(),GETCHARPOSITIONINLINE(),failed,BACKTRACKING);<endif>
    <ruleCleanUp()>
    <lexerRuleLabelFree()>
    <(ruleDescriptor.actions.after):execAfter()>
    <memoize>
}
// $ANTLR end <ruleName>
>>

/** How to generate code for the implicitly-defined lexer grammar rule
 * that chooses between lexer rules.
 */
tokensRule(ruleName,nakedBlock,args,block,ruleDescriptor) ::= <<
/** This is the entry point in to the lexer from an object that
 * wants to generate the next token, such as a pCOMMON_TOKEN_STREAM
 */
static void
mTokens(p<name> ctx)
{
    <block><\n>

    goto ruleTokensEx; /* Prevent compiler warnings */
ruleTokensEx: ;
}
>>

// S U B R U L E S

/** A (...) subrule with multiple alternatives */
block(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<

// <fileName>:<description>
{
    int alt<decisionNumber>=<maxAlt>;
    <decls>
    <@predecision()>
    <decision>
    <@postdecision()>
    <@prebranch()>
    switch (alt<decisionNumber>)
    {
    <alts:{a | <altSwitchCase(i,a)>}>
    }
}

```

```

    <@postbranch()>
  }
>>

/** A rule block with multiple alternatives */
ruleBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
{
  // <fileName>:<description>

  ANTLR3_UINT32 alt<decisionNumber>;

  alt<decisionNumber>=<maxAlt>;

  <decls>
  <@predecision()>
  <decision>
  <@postdecision()>
  switch (alt<decisionNumber>)
  {
    <alts:{ a | <altSwitchCase(i,a)> }>
  }
}
>>

ruleBlockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
// <fileName>:<description>
<decls>
<@prealt()>
<alts>
<@postalt()>
>>

/** A special case of a (...) subrule with a single alternative */
blockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
// <fileName>:<description>
<decls>
<@prealt()>
<alts>
<@postalt()>
>>

/** A (..)+ block with 1 or more alternatives */
positiveClosureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
// <fileName>:<description>
{
  int cnt<decisionNumber>=0;
  <decls>

```

```

<@preloop()>

for (;;)
{
    int alt<decisionNumber>=<maxAlt>;
<@predecision()>
<decision>
<@postdecision()>
switch (alt<decisionNumber>)
{
    <alts:{ a | <altSwitchCase(i,a)>}>
    default:

if ( cnt<decisionNumber> >= 1 )
{
    goto loop<decisionNumber>;
}
<ruleBacktrackFailure()>
<earlyExitEx()>
<@earlyExitException()>
goto rule<ruleDescriptor.name>Ex;
}
cnt<decisionNumber>++;
}
loop<decisionNumber>; /* Jump to here if this rule does not match */
<@postloop()>
}
>>

earlyExitEx() ::= <<
/* mismatchedSetEx()
*/
CONSTRUCTEX();
EXCEPTION->type = ANTLR3_EARLY_EXIT_EXCEPTION;
EXCEPTION->name = (void *)ANTLR3_EARLY_EXIT_NAME;
<\n>
>>
positiveClosureBlockSingleAlt ::= positiveClosureBlock

/** A (..)* block with 1 or more alternatives */
closureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::=
<<

// <fileName>:<description>
<decls>

<@preloop()>
for (;;)

```

```

{
  int alt<decisionNumber>=<maxAlt>;
  <@predecision()>
  <decision>
  <@postdecision()>
  switch (alt<decisionNumber>)
  {
    <alts:{ a | <altSwitchCase(i,a)>}>
  default:
    goto loop<decisionNumber>; /* break out of the loop */
    break;
  }
}
loop<decisionNumber>; /* Jump out to here if this rule does not match */
<@postloop()>
>>

closureBlockSingleAlt ::= closureBlock

/** Optional blocks (x)? are translated to (x|) by antlr before code generation
 * so we can just use the normal block template
 */
optionalBlock ::= block

optionalBlockSingleAlt ::= block

/** A case in a switch that jumps to an alternative given the alternative
 * number. A DFA predicts the alternative and then a simple switch
 * does the jump to the code that actually matches that alternative.
 */
altSwitchCase(altNum,alt) ::= <<
case <altNum>:
  <@prealt()>
  <alt>
  break;<\n>
>>

/** An alternative is just a list of elements; at outermost level */
alt(elements,altNum,description,autoAST,outerAlt,treeLevel,rew) ::= <<
// <fileName>:<description>
{
  <@declarations()>
  <@initializations()>
  <elements:element()>
  <rew>
  <@cleanup()>
}
>>

```

```

// ELEMENTS
/** What to emit when there is no rewrite. For auto build
 * mode, does nothing.
 */
noRewrite(rewriteBlockLevel, treeLevel) ::= ""

/** Dump the elements one per line */
element(e) ::= <<
<@prematch()>
<e.el><\n>
>>

/** match a token optionally with a label in front */
tokenRef(token,label,elementIndex,terminalOptions) ::= <<
<if(label)><label> = (<labelType>)<endif> MATCHT(<token>,
&FOLLOW_<token>_in_<ruleName><elementIndex>);
<checkRuleBacktrackFailure()>
>>

/** ids+=ID */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

listLabel(label,elem) ::= <<
if (list_<label> == NULL)
{
list_<label>=ctx->vectors->newVector(ctx->vectors);
}
list_<label>->add(list_<label>, <elem>, NULL);
>>

/** match a character */
charRef(char,label) ::= <<
<if(label)>
<label> = LA(1);<\n>
<endif>
MATCHC(<char>);
<checkRuleBacktrackFailure()>
>>

/** match a character range */
charRangeRef(a,b,label) ::= <<
<if(label)>
<label> = LA(1);<\n>

```

```

<endif>
MATCHRANGE(<a>, <b>);
<checkRuleBacktrackFailure()>
>>

/** For now, sets are interval tests and must be tested inline */
matchSet(s,label,elementIndex,terminalOptions,postmatchCode="") ::= <<
<if(label)>
<if(LEXER)>
<label>= LA(1);<\n>
<else>
<label>=(<labelType>)LT(1);<\n>
<endif>
<endif>
if ( <s> )
{
    CONSUME();
    <postmatchCode>
<if(!LEXER)>
    PERRORRECOVERY=ANTLR3_FALSE;
<endif>
    <if(backtracking)>FAILEDFLAG=ANTLR3_FALSE;<\n><endif>
}
else
{
    <ruleBacktrackFailure()>
    <mismatchedSetEx()>
    <@mismatchedSetException()>
<if(LEXER)>
    LRECOVER();
<else>
    RECOVERFROMMISMATCHEDSET(&FOLLOW_set_in_<ruleName><elementIndex>);
<endif>
    goto rule<ruleDescriptor.name>Ex;
}<\n>
>>

mismatchedSetEx() ::= <<
CONSTRUCTEX();
EXCEPTION->type      = ANTLR3_MISMATCHED_SET_EXCEPTION;
EXCEPTION->name      = (void *)ANTLR3_MISMATCHED_SET_NAME;
<if(PARSER)>
EXCEPTION->expectingSet = &FOLLOW_set_in_<ruleName><elementIndex>;
<endif>
>>

matchRuleBlockSet ::= matchSet

```

```

matchSetAndListLabel(s,label,elementIndex,postmatchCode) ::= <<
<matchSet(...)>
<listLabel(elem=label,...)>
>>

/** Match a string literal */
lexerStringRef(string,label,elementIndex) ::= <<
<if(label)>
<label>Start = GETCHARINDEX();
MATCHS(<string>);
<checkRuleBacktrackFailure()>
<label> = LEXSTATE->tokFactory->newToken(LEXSTATE->tokFactory);
<label>->setType(<label>, ANTLR3_TOKEN_INVALID);
<label>->setStartIndex(<label>, <label>Start);
<label>->setStopIndex(<label>, GETCHARINDEX()-1);
<label>->input = INPUT->tstream->istream;
<else>
MATCHS(<string>);
<checkRuleBacktrackFailure()><\n>
<endif>
>>

wildcard(token,label,elementIndex,terminalOptions) ::= <<
<if(label)>
<label>=(<labelType>)LT(1);<\n>
<endif>
MATCHANYT();
<checkRuleBacktrackFailure()>
>>

wildcardAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<wildcard(...)>
<listLabel(elem=label,...)>
>>

/** Match . wildcard in lexer */
wildcardChar(label, elementIndex) ::= <<
<if(label)>
<label> = LA(1);<\n>
<endif>
MATCHANY();
<checkRuleBacktrackFailure()>
>>

wildcardCharListLabel(label, elementIndex) ::= <<
<wildcardChar(...)>
<listLabel(elem=label,...)>
>>

```

```

/** Match a rule reference by invoking it possibly with arguments
 * and a return value or values. The 'rule' argument was the
 * target rule name, but now is type Rule, whose toString is
 * same: the rule name. Now though you can access full rule
 * descriptor stuff.
 */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
FOLLOWPUSH(FOLLOW_rule.name_in_ruleName<elementIndex>);
<if(label)><label>=<endif><if(scope)>ctx-><scope:delegateName()-><endif><rule.name>(ctx<if(scope)>-
><scope:delegateName()-><endif><if(args)>, <args; separator=","><endif>);<n>
FOLLOWPOP();
<checkRuleBacktrackFailure()>
>>

/** ids+=r */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(elem=label,...)>
>>

/** A lexer rule reference
 * The 'rule' argument was the target rule name, but now
 * is type Rule, whose toString is same: the rule name.
 * Now though you can access full rule descriptor stuff.
 */
lexerRuleRef(rule,label,args,elementIndex,scope) ::= <<
/* <description> */
<if(label)>
{
    ANTLR3_MARKER <label>Start<elementIndex> = GETCHARINDEX();
    <if(scope)>ctx-><scope:delegateName()-><endif>m<rule.name>(ctx<if(scope)>-
><scope:delegateName()-><endif> <if(args)>, <endif><args; separator=",">);
    <checkRuleBacktrackFailure()>
    <label> = LEXSTATE->tokFactory->newToken(LEXSTATE->tokFactory);
    <label>->setType(<label>, ANTLR3_TOKEN_INVALID);
    <label>->setStartIndex(<label>, <label>Start<elementIndex>);
    <label>->setStopIndex(<label>, GETCHARINDEX()-1);
    <label>->input = INPUT;
}
<else>
<if(scope)>ctx-><scope:delegateName()-><endif>m<rule.name>(ctx<if(scope)>-
><scope:delegateName()-><endif> <if(args)>, <endif><args; separator=",">);
<checkRuleBacktrackFailure()>
<endif>
>>

/** i+=INT in lexer */

```

```

lexerRuleRefAndListLabel(rule,label,args,elementIndex,scope) ::= <<
<lexerRuleRef(...)>
<listLabel(elem=label,...)>
>>

/** EOF in the lexer */
lexerMatchEOF(label,elementIndex) ::= <<
<if(label)>
{
  ANTLR3_UINT32 <label>Start<elementIndex>;
  <labelType> <label>;
  <label>Start<elementIndex> = GETCHARINDEX();
  MATCHC(ANTLR3_CHARSTREAM_EOF);
  <checkRuleBacktrackFailure()>
  <label> = LEXSTATE->tokFactory->newToken(LEXSTATE->tokFactory);
  <label>->setType(<label>, ANTLR3_TOKEN_EOF);
  <label>->setStartIndex(<label>, <label>Start<elementIndex>);
  <label>->setStopIndex(<label>, GETCHARINDEX()-1);
  <label>->input = INPUT->tstream->istream;
}
<else>
  MATCHC(ANTLR3_CHARSTREAM_EOF);
  <checkRuleBacktrackFailure()>
  <endif>
>>

// used for left-recursive rules
recRuleDefArg()          ::= "int <recRuleArg()>"
recRuleArg()             ::= "_p"
recRuleAltPredicate(ruleName,opPrec) ::= "<recRuleArg()> \<= <opPrec>"
recRuleSetResultAction() ::= "root_0=$<ruleName>_primary.tree;"
recRuleSetReturnAction(src,name)  ::= "$<name>=$<src>.<name>;"

/** match ^(root children) in tree parser */
tree(root, actionsAfterRoot, children, nullableChildList, enclosingTreeLevel, treeLevel) ::= <<
<root:element()>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if ( LA(1)==ANTLR3_TOKEN_DOWN ) {
  MATCHT(ANTLR3_TOKEN_DOWN, NULL);
  <checkRuleBacktrackFailure()>
  <children:element()>
  MATCHT(ANTLR3_TOKEN_UP, NULL);
  <checkRuleBacktrackFailure()>
}
<else>
MATCHT(ANTLR3_TOKEN_DOWN, NULL);
<checkRuleBacktrackFailure()>

```

```

<children:element()>
MATCHT(ANTLR3_TOKEN_UP, NULL);
<checkRuleBacktrackFailure()>
<endif>
>>

/** Every predicate is used as a validating predicate (even when it is
 * also hoisted into a prediction expression).
 */
validateSemanticPredicate(pred,description) ::= <<
if ( !(<evalPredicate(...)> )
{
  <ruleBacktrackFailure()>
  <newFPE(...)>
}
>>

newFPE() ::= <<
  CONSTRUCTEX();
  EXCEPTION->type      = ANTLR3_FAILED_PREDICATE_EXCEPTION;
  EXCEPTION->message   = (void *)"<description>";
  EXCEPTION->ruleName = (void *)"<ruleName>";
  <\n>
>>

// F i x e d D F A (if-then-else)

dfaState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<

{
  int LA<decisionNumber>_<stateNumber> = LA(<k>);
  <edges; separator="\nelse ">
  else
  {
    <if(eotPredictsAlt)>
      alt<decisionNumber>=<eotPredictsAlt>;
    <else>
      <ruleBacktrackFailure()>

      <newNVException()>
      goto rule<ruleDescriptor.name>Ex;

    <endif>
  }
}
>>

newNVException() ::= <<

```

```

CONSTRUCTEX();
EXCEPTION->type      = ANTLR3_NO_VIABLE_ALT_EXCEPTION;
EXCEPTION->message    = (void *)"<description>";
EXCEPTION->decisionNum = <decisionNumber>;
EXCEPTION->state      = <stateNumber>;
<@noViableAltException()>
<\n>
>>

/** Same as a normal DFA state except that we don't examine lookahead
 * for the bypass alternative. It delays error detection but this
 * is faster, smaller, and more what people expect. For (X)? people
 * expect "if ( LA(1)==X ) match(X);" and that's it.
 */
dfaOptionalBlockState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
{
  int LA<decisionNumber>_<stateNumber> = LA(<k>);
  <edges; separator="\nelse ">
}
>>

/** A DFA state that is actually the loopback decision of a closure
 * loop. If end-of-token (EOT) predicts any of the targets then it
 * should act like a default clause (i.e., no error can be generated).
 * This is used only in the lexer so that for ('a')* on the end of a rule
 * anything other than 'a' predicts exiting.
 */

dfaLoopbackStateDecls() ::= <<
ANTLR3_UINT32 LA<decisionNumber>_<stateNumber>;
>>
dfaLoopbackState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
{
  /* dfaLoopbackState(k,edges,eotPredictsAlt,description,stateNumber,semPredState)
  */
  int LA<decisionNumber>_<stateNumber> = LA(<k>);
  <edges; separator="\nelse "><\n>
  <if(eotPredictsAlt)>
  <if(!edges)>
  alt<decisionNumber>=<eotPredictsAlt>; <! if no edges, don't gen ELSE !>
  <else>
  else
  {
  alt<decisionNumber>=<eotPredictsAlt>;
  }<\n>
  <endif>
  <endif>
}

```

```

>>

/** An accept state indicates a unique alternative has been predicted */
dfaAcceptState(alt) ::= "alt<decisionNumber>=<alt>";

/** A simple edge with an expression. If the expression is satisfied,
 * enter to the target state. To handle gated productions, we may
 * have to evaluate some predicates for this edge.
 */
dfaEdge(labelExpr, targetState, predicates) ::= <<
if ( <if(predicates)><predicates> && <endif><labelExpr>))
{
    <targetState>
}
>>

// F i x e d D F A (switch case)

/** A DFA state where a SWITCH may be generated. The code generator
 * decides if this is possible: CodeGenerator.canGenerateSwitch().
 */
dfaStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ( LA(<k> )
{
    <edges; separator="\n">

default:
<if(eotPredictsAlt)>
    alt<decisionNumber>=<eotPredictsAlt>;
<else>
    <ruleBacktrackFailure()>
    <newNVEException()>
    goto rule<ruleDescriptor.name>Ex;<\n>
<endif>
}<\n>
>>

dfaOptionalBlockStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ( LA(<k> )
{
    <edges; separator="\n">
}<\n>
>>

dfaLoopbackStateSwitch(k, edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ( LA(<k> )
{
    <edges; separator="\n"><\n>

```

```

<if(eotPredictsAlt)>
default:
    alt<decisionNumber>=<eotPredictsAlt>;
    break;<\n>
<endif>
}<\n>
>>

dfaEdgeSwitch(labels, targetState) ::= <<
<labels:{it |case <it>:}; separator="\n">
{
    <targetState>
}
    break;
>>

// C y c l i c D F A

/** The code to initiate execution of a cyclic DFA; this is used
 * in the rule to predict an alt just like the fixed DFA case.
 * The <name> attribute is inherited via the parser, lexer, ...
 */
dfaDecision(decisionNumber,description) ::= <<
alt<decisionNumber> = cdfa<decisionNumber>.predict(ctx, RECOGNIZER, ISTREAM,
&cdfa<decisionNumber>);
<checkRuleBacktrackFailure()>
>>

/* Dump DFA tables as static initialized arrays of shorts(16 bits)/characters(8 bits)
 * which are then used to statically initialize the dfa structure, which means that there
 * is no runtime initialization whatsoever, other than anything the C compiler might
 * need to generate. In general the C compiler will lay out memory such that there is no
 * runtime code required.
 */
cyclicDFA(dfa) ::= <<
/** Static dfa state tables for Cyclic dfa:
 * <dfa.description>
 */
static const ANTLR3_INT32 dfa<dfa.decisionNumber>_eot[<dfa.numberofStates>] =
{
    <dfa.eot; wrap="\n", separator=", ", null="-1">
};
static const ANTLR3_INT32 dfa<dfa.decisionNumber>_eof[<dfa.numberofStates>] =
{
    <dfa.eof; wrap="\n", separator=", ", null="-1">
};
static const ANTLR3_INT32 dfa<dfa.decisionNumber>_min[<dfa.numberofStates>] =
{

```

```

<dfa.min; wrap="\n", separator=", ", null="-1">
};
static const ANTLR3_INT32 dfa<dfa.decisionNumber>_max[<dfa.numberofStates>] =
{
<dfa.max; wrap="\n", separator=", ", null="-1">
};
static const ANTLR3_INT32 dfa<dfa.decisionNumber>_accept[<dfa.numberofStates>] =
{
<dfa.accept; wrap="\n", separator=", ", null="-1">
};
static const ANTLR3_INT32 dfa<dfa.decisionNumber>_special[<dfa.numberofStates>] =
{
<dfa.special; wrap="\n", separator=", ", null="-1">
};

/** Used when there is no transition table entry for a particular state */
#define dfa<dfa.decisionNumber>_T_empty NULL

<dfa.edgeTransitionClassMap.keys:{ table |
static const ANTLR3_INT32 dfa<dfa.decisionNumber>_T<i0>[] =
{
<table; separator=", ", wrap="\n", null="-1">
\};}; null = "">

/* Transition tables are a table of sub tables, with some tables
* reused for efficiency.
*/
static const ANTLR3_INT32 * const dfa<dfa.decisionNumber>_transitions[] =
{
<dfa.transitionEdgeTables:{ xref[dfa<dfa.decisionNumber>_T<xref>]; separator=", ", wrap="\n", null="_empty">
};

<if(dfa.specialStateSTs)>
static ANTLR3_INT32 dfa<dfa.decisionNumber>_sst(p<name> ctx, pANTLR3_BASE_RECOGNIZER recognizer,
pANTLR3_INT_STREAM is, pANTLR3_CYCLIC_DFA dfa, ANTLR3_INT32 s)
{
ANTLR3_INT32 _s;

_s = s;
switch (s)
{
<dfa.specialStateSTs:{ state |
case <i0>:

<state>}; separator="\n">
}
<if(backtracking)>
if (BACKTRACKING > 0)

```

```

{
FAILEDFLAG = ANTLR3_TRUE;
return -1;
}
<endif>

CONSTRUCTEX();
EXCEPTION->type      = ANTLR3_NO_VIABLE_ALT_EXCEPTION;
EXCEPTION->message    = (void *)"<dfa.description>";
EXCEPTION->decisionNum = <dfa.decisionNumber>;
EXCEPTION->state      = _s;
<@noViableAltException()>
return -1;
}
<endif>

<@errorMethod()>

/* Declare tracking structure for Cyclic DFA <dfa.decisionNumber>
*/
static
ANTLR3_CYCLIC_DFA cdfa<dfa.decisionNumber>
= {
    <dfa.decisionNumber>, /* Decision number of this dfa */
    /* Which decision this represents: */
    (const pANTLR3_UCHAR)"<dfa.description>",
<if(dfa.specialStateSTs)>
    (C DFA_SPECIAL_FUNC) dfa<dfa.decisionNumber>_sst,
<else>
    (C DFA_SPECIAL_FUNC) antlr3dfaspecialStateTransition, /* Default special state transition function */
<endif>

    antlr3dfaspecialTransition, /* DFA specialTransition is currently just a default function in the runtime */
    antlr3dfapredict, /* DFA simulator function is in the runtime */
    dfa<dfa.decisionNumber>_eot, /* EOT table */
    dfa<dfa.decisionNumber>_eof, /* EOF table */
    dfa<dfa.decisionNumber>_min, /* Minimum tokens for each state */
    dfa<dfa.decisionNumber>_max, /* Maximum tokens for each state */
    dfa<dfa.decisionNumber>_accept, /* Accept table */
    dfa<dfa.decisionNumber>_special, /* Special transition states */
    dfa<dfa.decisionNumber>_transitions /* Table of transition tables */

};
/* End of Cyclic DFA <dfa.decisionNumber>
* -----
*/
>>

```

```

/** A state in a cyclic DFA; it's a special state and part of a big switch on
 * state.
 */
cyclicDFAState(decisionNumber,stateNumber,edges,needErrorClause,semPredState) ::= <<
{
  ANTLR3_UINT32 LA<decisionNumber>_<stateNumber>;<\n>
  ANTLR3_MARKER index<decisionNumber>_<stateNumber>;<\n>

  LA<decisionNumber>_<stateNumber> = LA(1);<\n>
  <if(semPredState)> <! get next lookahead symbol to test edges, then rewind !>
  index<decisionNumber>_<stateNumber> = INDEX();<\n>
  REWINDLAST();<\n>
  <endif>
  s = -1;
  <edges; separator="\nelse ">
  <if(semPredState)> <! return input cursor to state before we rewound !>
  SEEK(index<decisionNumber>_<stateNumber>);<\n>
  <endif>
  if ( s>=0 )
  {
  return s;
  }
}
break;
>>

/** Just like a fixed DFA edge, test the lookahead and indicate what
 * state to jump to next if successful.
 */
cyclicDFAEdge(labelExpr, targetStateNumber, edgeNumber, predicates) ::= <<
if ( <if(predicates)><predicates> && <endif><labelExpr> )
{
  s = <targetStateNumber>;
}<\n>
>>

/** An edge pointing at end-of-token; essentially matches any char;
 * always jump to the target.
 */
eotDFAEdge(targetStateNumber,edgeNumber, predicates) ::= <<
s = <targetStateNumber>;<\n>
>>

// D F A E X P R E S S I O N S

andPredicates(left,right) ::= "( (<left>) && (<right> )"

```

```

orPredicates(operands) ::= "<operands:{o|(<o>)}; separator=\\|\\>"

notPredicate(pred) ::= "!( <evalPredicate(pred,{ })>)"

evalPredicate(pred,description) ::= "<pred>"

evalSynPredicate(pred,description) ::= "<pred>(ctx)"

lookaheadTest(atom,k,atomAsInt) ::= "LA<decisionNumber>_<stateNumber> == <atom>"

/** Sometimes a lookahead test cannot assume that LA(k) is in a temp variable
 * somewhere. Must ask for the lookahead directly.
 */
isolatedLookaheadTest(atom,k,atomAsInt) ::= "LA(<k>) == <atom>"

lookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= <%
((LA<decisionNumber>_<stateNumber> >= <lower>) && (LA<decisionNumber>_<stateNumber> \<= <upper>))
%>

isolatedLookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= "((LA(<k>) >= <lower>) &&
(LA(<k>) \<= <upper>))"

setTest(ranges) ::= "<ranges; separator=\\ || \\>"

// A T T R I B U T E S

makeScopeSet() ::= <<
/* makeScopeSet()
 */
/** Definition of the <scope.name> scope variable tracking
 * structure. An instance of this structure is created by calling
 * <name>_<scope.name>Push().
 */
typedef struct <scopeStruct(sname=scope.name,...)>_struct
{
    /** Function that the user may provide to be called when the
     * scope is destroyed (so you can free pANTLR3_HASH_TABLES and so on)
     *
     * \param POinter to an instance of this typedef/struct
     */
    void (ANTLR3_CDECL *free) (struct <scopeStruct(sname=scope.name,...)>_struct * frame);

    /** =====
     * Programmer defined variables...
     */
    <scope.attributes:{it |<it.decl>;}; separator="\n">

    /** End of programmer defined variables

```

```

* =====
*/
}
<scopeStruct(sname=scope.name,...)>, * <scopeType(sname=scope.name,...)>;

>>

globalAttributeScopeDecl(scope) ::= <<
<if(scope.attributes)>
/* globalAttributeScopeDecl(scope)
*/
<makeScopeSet(...)>
<endif>
>>

ruleAttributeScopeDecl(scope) ::= <<
<if(scope.attributes)>
/* ruleAttributeScopeDecl(scope)
*/
<makeScopeSet(...)>
<endif>
>>

globalAttributeScopeFuncDecl(scope) ::=
<<
/* globalAttributeScopeFuncDecl(scope)
*/
<if(scope.attributes)>
/* -----
* Function declaration for creating a <name>_<scope.name> scope set
*/
static <scopeType(sname=scope.name,...)> <scopePushName(sname=scope.name,...)>(p<name> ctx);
static void ANTLR3_CDECL <scope.name>Free(<scopeType(sname=scope.name)> scope);
/* ----- */

<endif>
>>

globalAttributeScopeFuncMacro(scope) ::= <<
<if(scope.attributes)>
/* globalAttributeScopeFuncMacro(scope)
*/
/** Function for popping the top value from a <scopeStack(sname=scope.name)>
*/
void
<scopePopName(sname=scope.name,...)>(p<name> ctx)
{
// First see if the user defined a function they want to be called when a

```

```

    // scope is popped/freed.
    //
    // If the user supplied the scope entries with a free function,then call it first
    //
    if (SCOPE_TOP(<scope.name>->free != NULL)
    {
        SCOPE_TOP(<scope.name>->free(SCOPE_TOP(<scope.name>));
    }

    // Now we decrement the scope's upper limit bound. We do not actually pop the scope as
    // we want to reuse scope entries if we do continuous push and pops. Most scopes don't
    // next too far so we don't want to keep freeing and allocating them
    //
    ctx-><scopeStack(sname=scope.name,...)>_limit--;
    SCOPE_TOP(<scope.name>) = (<scopeType(sname=scope.name)>)(ctx-><scopeStack(sname=scope.name,...)>-
>get(ctx-><scopeStack(sname=scope.name,...)>, ctx-><scopeStack(sname=scope.name,...)>_limit - 1));
    }
<endif>
>>

ruleAttributeScopeFuncDecl(scope) ::= <<
<if(scope.attributes)>
/* ruleAttributeScopeFuncDecl(scope)
*/
/* -----
* Function declarations for creating a <name>_<scope.name> scope set
*/
static <scopeType(sname=scope.name,...)> <scopePushName(sname=scope.name,...)>(p<name> ctx);
static void ANTLR3_CDECL <scope.name>Free(<scopeType(sname=scope.name)> scope);
/* ----- */

<endif>
>>

ruleAttributeScopeFuncMacro(scope) ::= <<
<if(scope.attributes)>
/* ruleAttributeScopeFuncMacro(scope)
*/
/** Function for popping the top value from a <scopeStack(sname=scope.name,...)>
*/
void
<scopePopName(sname=scope.name,...)>(p<name> ctx)
{
    // First see if the user defined a function they want to be called when a
    // scope is popped/freed.
    //
    // If the user supplied the scope entries with a free function,then call it first
    //

```

```

if (SCOPE_TOP(<scope.name>->free != NULL)
{
    SCOPE_TOP(<scope.name>->free(SCOPE_TOP(<scope.name>));
}

// Now we decrement the scope's upper limit bound. We do not actually pop the scope as
// we want to reuse scope entries if we do continuous push and pops. Most scopes don't
// next too far so we don't want to keep freeing and allocating them
//
ctx-><scopeStack(sname=scope.name,...)>_limit--;
SCOPE_TOP(<scope.name>) = (<scopeType(sname=scope.name)>)(ctx-><scopeStack(sname=scope.name,...)>-
>get(ctx-><scopeStack(sname=scope.name,...)>, ctx-><scopeStack(sname=scope.name,...)>_limit - 1));
}

<endif>
>>

globalAttributeScopeDef(scope) ::=
<<
/* globalAttributeScopeDef(scope)
*/
<if(scope.attributes)>
/** Pointer to the <scope.name> stack for use by <scopePushName(sname=scope.name)>()
* and <scopePopName(sname=scope.name,...)>()
*/
pANTLR3_STACK <scopeStack(sname=scope.name)>;
ANTLR3_UINT32 <scopeStack(sname=scope.name)>_limit;
/** Pointer to the top of the stack for the global scope <scopeStack(sname=scope.name)>
*/
<scopeType(sname=scope.name,...)> (*<scopePushName(sname=scope.name,...)>)(struct <name>_Ctx_struct *
ctx);
<scopeType(sname=scope.name,...)> <scopeTopDecl(sname=scope.name,...)>;

<endif>
>>

ruleAttributeScopeDef(scope) ::= <<
<if(scope.attributes)>
/* ruleAttributeScopeDef(scope)
*/
/** Pointer to the <scope.name> stack for use by <scopePushName(sname=scope.name)>()
* and <scopePopName(sname=scope.name,...)>()
*/
pANTLR3_STACK <scopeStack(sname=scope.name,...)>;
ANTLR3_UINT32 <scopeStack(sname=scope.name,...)>_limit;
<scopeType(sname=scope.name,...)> (*<scopePushName(sname=scope.name,...)>)(struct <name>_Ctx_struct *
ctx);
<scopeType(sname=scope.name,...)> <scopeTopDecl(sname=scope.name,...)>;

```

```

<endif>
>>

globalAttributeScopeFuncs(scope) ::= <<
<if(scope.attributes)>
/* globalAttributeScopeFuncs(scope)
*/
<attributeFuncs(scope)>
<endif>
>>

ruleAttributeScopeFuncs(scope) ::= <<
<if(scope.attributes)>
/* ruleAttributeScopeFuncs(scope)
*/
<attributeFuncs(scope)>
<endif>
>>

globalAttributeScope(scope) ::= <<
<if(scope.attributes)>
/* globalAttributeScope(scope)
*/
ctx-><scopePushName(sname=scope.name,...)> = <scopePushName(sname=scope.name,...)>;
ctx-><scopeStack(sname=scope.name,...)> = antlr3StackNew(0);
ctx-><scopeStack(sname=scope.name,...)>_limit = 0;
<scopeTop(sname=scope.name,...)> = NULL;
<endif>
>>

ruleAttributeScope(scope) ::=
<<
<if(scope.attributes)>
/* ruleAttributeScope(scope)
*/
ctx-><scopePushName(sname=scope.name,...)> = <scopePushName(sname=scope.name,...)>;
ctx-><scopeStack(sname=scope.name,...)> = antlr3StackNew(0);
ctx-><scopeStack(sname=scope.name,...)>_limit = 0;
<scopeTop(sname=scope.name,...)> = NULL;
<endif>
>>

globalAttributeScopeFree(scope) ::= <<
<if(scope.attributes)>
/* globalAttributeScope(scope)
*/
ctx-><scopeStack(sname=scope.name,...)>->free(ctx-><scopeStack(sname=scope.name,...)>);
<endif>

```

```

>>

ruleAttributeScopeFree(scope) ::=
<<
<if(scope.attributes)>
/* ruleAttributeScope(scope)
*/
ctx-><scopeStack(sname=scope.name,...)>->free(ctx-><scopeStack(sname=scope.name,...)>);
<endif>
>>

scopeTopDecl(sname) ::= <<
p<name>_<sname>Top
>>

scopeTop(sname) ::= <<
ctx-><scopeTopDecl(sname=sname,...)>
>>

scopePop(sname) ::= <<
<scopePopName(sname=sname,...)>(ctx);
>>

scopePush(sname) ::= <<
p<name>_<sname>Push(ctx)
>>

scopePopName(sname) ::= <<
p<name>_<sname>Pop
>>

scopePushName(sname) ::= <<
p<name>_<sname>Push
>>

scopeType(sname) ::= <<
p<name>_<sname>_SCOPE
>>

scopeStruct(sname) ::= <<
<name>_<sname>_SCOPE
>>

scopeStack(sname) ::= <<
p<name>_<sname>Stack
>>

attributeFuncs(scope) ::= <<

```

```

<if(scope.attributes)>
/* attributeFuncs(scope)
*/

static void ANTLR3_CDECL <scope.name>Free(<scopeType(sname=scope.name)> scope)
{
    ANTLR3_FREE(scope);
}

/** \brief Allocate initial memory for a <name> <scope.name> scope variable stack entry and
 *     add it to the top of the stack.
 *
 * \remark
 * By default the structure is freed with ANTLR_FREE(), but you can use the
 * the \@init action to install a pointer to a custom free() routine by
 * adding the code:
 * \code
 * <scopeTop(sname=scope.name)>->free = myroutine;
 * \endcode
 *
 * With lots of comments of course! The routine should be declared in
 * \@members { } as:
 * \code
 * void ANTLR3_CDECL myfunc( <scopeType(sname=scope.name)> ptr).
 * \endcode
 *
 * It should perform any custom freeing stuff that you need (call ANTLR_FREE3, not free())
 * NB: It should not free the pointer it is given, which is the scope stack entry itself
 * and will be freed by the function that calls your custom free routine.
 *
 */
static <scopeType(sname=scope.name)>
<scopePushName(sname=scope.name)>(p<name> ctx)
{
    /* Pointer used to create a new set of attributes
    */
    <scopeType(sname=scope.name)>    newAttributes;

    /* Allocate the memory for a new structure if we need one.
    */
    if (ctx-><scopeStack(sname=scope.name)>->size(ctx-><scopeStack(sname=scope.name)>) > ctx-
    ><scopeStack(sname=scope.name)>_limit)
    {
        // The current limit value was less than the number of scopes available on the stack so
        // we can just reuse one. Our limit tracks the stack count, so the index of the entry we want
        // is one less than that, or conveniently, the current value of limit.
        //
        newAttributes = (<scopeType(sname=scope.name)>)ctx-><scopeStack(sname=scope.name)>->get(ctx-

```

```

<<scopeStack(sname=scope.name)>, ctx-><scopeStack(sname=scope.name)>_limit);
    }
    else
    {
        // Need a new allocation
        //
        newAttributes = (<scopeType(sname=scope.name)>)
        ANTLR3_MALLOC(sizeof(<scopeStruct(sname=scope.name)>));
        if (newAttributes != NULL)
        {
            /* Standard ANTLR3 library implementation
            */
            ctx-><scopeStack(sname=scope.name)>->push(ctx-><scopeStack(sname=scope.name)>, newAttributes,
            (void (*)(void *))<scope.name>Free);
        }
    }

    // Blank out any previous free pointer, the user might or might install a new one.
    //
    newAttributes->free = NULL;

    // Indicate the position in the available stack that the current level is at
    //
    ctx-><scopeStack(sname=scope.name)>_limit++;

    /* Return value is the pointer to the new entry, which may be used locally
    * without de-referencing via the context.
    */
    return newAttributes;
}<\n>

<endif>
>>
returnStructName(r) ::= "<r.name>_return"

returnType() ::= <%
<if(!ruleDescriptor.isSynPred)>
<if(ruleDescriptor.hasMultipleReturnValues)>
<ruleDescriptor.grammar.recognizerName>_<ruleDescriptor:returnStructName()>
<else>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnType>
<else>
void
<endif>
<endif>
<else>
ANTLR3_BOOLEAN

```

```

<endif>
%>

/** Generate the C type associated with a single or multiple return
 * value(s).
 */
ruleLabelType(referencedRule) ::= <%
<if(referencedRule.hasMultipleReturnValues)>
<referencedRule.grammar.recognizerName>_<referencedRule.name>_return
<else>
<if(referencedRule.hasSingleReturnValue)>
<referencedRule.singleValueType>
<else>
void
<endif>
<endif>
%>

delegateName(d) ::= <<
<if(d.label)><d.label><else>g<it.name><endif>
>>

/** Using a type to init value map, try to init a type; if not in table
 * must be an object, default value is "0".
 */
initValue(typeName) ::= <<
= <cTypeInitMap.(typeName)>
>>

/** Define a rule label */
ruleLabelDef(label) ::= <<
<ruleLabelType(referencedRule=label.referencedRule)> <label.label.text>;
#undef RETURN_TYPE_<label.label.text>
#define RETURN_TYPE_<label.label.text> <ruleLabelType(referencedRule=label.referencedRule)><\n>
>>

/** Rule label default value */
ruleLabelInitVal(label) ::= <<
>>

ASTLabelType() ::=
"<if(recognizer.ASTLabelType)><recognizer.ASTLabelType><else>pANTLR3_BASE_TREE<endif>"

/** Define a return struct for a rule if the code needs to access its
 * start/stop tokens, tree stuff, attributes, ... Leave a hole for
 * subgroups to stick in members.
 */
returnScope(scope) ::= <<
<if(!ruleDescriptor.isSynPred)>

```

```

<if(ruleDescriptor.hasMultipleReturnValues)>
typedef struct <ruleDescriptor.grammar.recognizerName>_<ruleDescriptor:returnStructName()>_struct
{
<if(!TREE_PARSER)>
  /** Generic return elements for ANTLR3 rules that are not in tree parsers or returning trees
  */
  pANTLR3_COMMON_TOKEN  start;
  pANTLR3_COMMON_TOKEN  stop;
<else>
  <recognizer.ASTLabelType>  start;
  <recognizer.ASTLabelType>  stop;
<endif>
<@ruleReturnMembers()>
  <ruleDescriptor.returnScope.attributes:{it |<it.type> <it.name>;}; separator="\n">
}
  <ruleDescriptor.grammar.recognizerName>_<ruleDescriptor:returnStructName()>;<\n><\n>
<endif>
<endif>
>>

parameterScope(scope) ::= <<
<scope.attributes:{it |<it.decl>}; separator=", ">
>>

parameterAttributeRef(attr) ::= "<attr.name>"
parameterSetAttributeRef(attr,expr) ::= "<attr.name>=<expr>";

/** Note that the scopeAttributeRef does not have access to the
 * grammar name directly
 */
scopeAttributeRef(scope,attr,index,negIndex) ::= <%
<if(negIndex)>
  ((SCOPE_TYPE(<scope>))(ctx->SCOPE_STACK(<scope>)->get( ctx->SCOPE_STACK(<scope>), ctx-
>SCOPE_STACK(<scope>)->size(ctx->SCOPE_STACK(<scope>)) - <negIndex> - 1 ))-><attr.name>
<else>
<if(index)>
  ((SCOPE_TYPE(<scope>))(ctx->SCOPE_STACK(<scope>)->get(ctx->SCOPE_STACK(<scope>),
(ANTLR3_UINT32)<index> ) ))-><attr.name>
<else>
  (SCOPE_TOP(<scope>))-><attr.name>
<endif>
<endif>
%>

scopeSetAttributeRef(scope,attr,expr,index,negIndex) ::= <%
<if(negIndex)>
  ((SCOPE_TYPE(<scope>))(ctx->SCOPE_STACK(<scope>)->get( ctx->SCOPE_STACK(<scope>), ctx-
>SCOPE_STACK(<scope>)->size(ctx->SCOPE_STACK(<scope>)) - <negIndex> - 1 ))-><attr.name> = <expr>;

```

```

<else>
<if(index)>
((SCOPE_TYPE(<scope>))(ctx->SCOPE_STACK(<scope>)->get(ctx->SCOPE_STACK(<scope>),
(ANTLR3_UINT32)<index> ) )-><attr.name> = <expr>;
<else>
(SCOPE_TOP(<scope>))-><attr.name>=<expr>;
<endif>
<endif>
%>

```

```

/** $x is either global scope or x is rule with dynamic scope; refers
* to stack itself not top of stack. This is useful for predicates
* like {$function.size(>0 && $function::name.equals("foo"))?
*/
isolatedDynamicScopeRef(scope) ::= "ctx->SCOPE_STACK(<scope>)"

```

```

/** reference an attribute of rule; might only have single return value */
ruleLabelRef(referencedRule,scope,attr) ::= <<
<if(referencedRule.hasMultipleReturnValues)>
<scope>.<attr.name>
<else>
<scope>
<endif>
>>

```

```

returnAttributeRef(ruleDescriptor,attr) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name>
<else>
<attr.name>
<endif>
>>

```

```

returnSetAttributeRef(ruleDescriptor,attr,expr) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name>=<expr>;
<else>
<attr.name>=<expr>;
<endif>
>>

```

```

/** How to translate $tokenLabel */
tokenLabelRef(label) ::= "<label>"

```

```

/** ids+=ID {$ids} or e+=expr {$e} */
listLabelRef(label) ::= "list_<label>"

```

```

// not sure the next are the right approach
//
tokenLabelPropertyRef_text(scope,attr) ::= "<scope>->getText(<scope>)"
tokenLabelPropertyRef_type(scope,attr) ::= "<scope>->getType(<scope>)"
tokenLabelPropertyRef_line(scope,attr) ::= "<scope>->getLine(<scope>)"
tokenLabelPropertyRef_pos(scope,attr) ::= "<scope>->getCharPositionInLine(<scope>)"
tokenLabelPropertyRef_channel(scope,attr) ::= "<scope>->getChannel(<scope>)"
tokenLabelPropertyRef_index(scope,attr) ::= "<scope>->getTokenIndex(<scope>)"
tokenLabelPropertyRef_tree(scope,attr) ::= "<scope>->tree"
tokenLabelPropertyRef_int(scope,attr) ::= "<scope>->getText(<scope>->toInt32(<scope>->getText(<scope>)))"

ruleLabelPropertyRef_start(scope,attr) ::= "<scope>.start"
ruleLabelPropertyRef_stop(scope,attr) ::= "<scope>.stop"
ruleLabelPropertyRef_tree(scope,attr) ::= "<scope>.tree"
ruleLabelPropertyRef_text(scope,attr) ::= <<
<if(TREE_PARSER)>
(STRSTREAM->toStringSS(STRSTREAM, <scope>.start, <scope>.start))
<else>
(STRSTREAM->toStringTT(STRSTREAM, <scope>.start, <scope>.stop))
<endif>
>>

ruleLabelPropertyRef_st(scope,attr) ::= "<scope>.st"

/** Isolated $RULE ref ok in lexer as it's a Token */
lexerRuleLabel(label) ::= "<label>"

lexerRuleLabelPropertyRef_type(scope,attr) ::= "<scope>->getType(<scope>)"
lexerRuleLabelPropertyRef_line(scope,attr) ::= "<scope>->getLine(<scope>)"
lexerRuleLabelPropertyRef_pos(scope,attr) ::= "<scope>->getCharPositionInLine(<scope>)"
lexerRuleLabelPropertyRef_channel(scope,attr) ::= "<scope>->getChannel(<scope>)"
lexerRuleLabelPropertyRef_index(scope,attr) ::= "<scope>->getTokenIndex(<scope>)"
lexerRuleLabelPropertyRef_text(scope,attr) ::= "<scope>->getText(<scope>)"

// Somebody may ref $template or $tree or $stop within a rule:
rulePropertyRef_start(scope,attr) ::= "retval.start"
rulePropertyRef_stop(scope,attr) ::= "retval.stop"
rulePropertyRef_tree(scope,attr) ::= "retval.tree"
rulePropertyRef_text(scope,attr) ::= <<
<if(TREE_PARSER)>
INPUT->toStringSS(INPUT, ADAPTOR->getTokenStartIndex(ADAPTOR, retval.start), ADAPTOR-
>getTokenStopIndex(ADAPTOR, retval.start))
<else>
STRSTREAM->toStringTT(STRSTREAM, retval.start, LT(-1))
<endif>
>>
rulePropertyRef_st(scope,attr) ::= "retval.st"

```

```

lexerRulePropertyRef_text(scope,attr) ::= "LEXER->getText(LEXER)"
lexerRulePropertyRef_type(scope,attr) ::= "_type"
lexerRulePropertyRef_line(scope,attr) ::= "LEXSTATE->tokenStartLine"
lexerRulePropertyRef_pos(scope,attr) ::= "LEXSTATE->tokenStartCharPositionInLine"
lexerRulePropertyRef_channel(scope,attr) ::= "LEXSTATE->channel"
lexerRulePropertyRef_start(scope,attr) ::= "LEXSTATE->tokenStartCharIndex"
lexerRulePropertyRef_stop(scope,attr) ::= "(LEXER->getCharIndex(LEXER)-1)"
lexerRulePropertyRef_index(scope,attr) ::= "-1" // undefined token index in lexer
lexerRulePropertyRef_int(scope,attr) ::= "LEXER->getText(LEXER)->toInt32(LEXER->getText(LEXER))"

```

```

// setting $st and $tree is allowed in local rule. everything else is flagged as error
ruleSetPropertyRef_tree(scope,attr,expr) ::= "retval.tree=<expr>";
ruleSetPropertyRef_st(scope,attr,expr) ::= "retval.st=<expr>";

```

```

/** How to deal with an @after for C targets. Because we cannot rely on
 * any garbage collection, after code is executed even in backtracking
 * mode. Must be documented clearly.
 */

```

```

execAfter(action) ::= <<
{
  <action>
}
>>

```

```

/** How to execute an action (when not backtracking) */

```

```

execAction(action) ::= <<
<if(backtracking)>
<if(actions.(actionScope).synpredgate)>
if ( <actions.(actionScope).synpredgate> )
{
  <action>
}
<else>
if ( BACKTRACKING == 0 )
{
  <action>
}
<endif>
<else>
{
  <action>
}
<endif>
>>

```

```

// M I S C (properties, etc...)

```

```
bitsetDeclare(name, words64) ::= <<
```

```
/** Bitset defining follow set for error recovery in rule state: <name> */
```

```
static ANTLR3_BITWORD <name>_bits[] = { <words64:{it |ANTLR3_UINT64_LIT(<it>)}; separator=", "> };
```

```
static ANTLR3_BITSET_LIST <name> = { <name>_bits, <length(words64)> };
```

```
>>
```

```
bitset(name, words64) ::= <<
```

```
antlr3BitsetSetAPI(&<name>);<\n>
```

```
>>
```

```
codeFileExtension() ::= ".c"
```

```
true_value() ::= "ANTLR3_TRUE"
```

```
false_value() ::= "ANTLR3_FALSE"
```

Found in path(s):

* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/C/C.stg

No license file was found, but licenses were detected in source scan.

```
/*
```

```
* [The "BSD license"]
```

```
* Copyright (c) 2011 Terence Parr
```

```
* All rights reserved.
```

```
*
```

```
* Conversion to C#:
```

```
* Copyright (c) 2011 Sam Harwell, Pixel Mine, Inc.
```

```
* All rights reserved.
```

```
*
```

```
* Redistribution and use in source and binary forms, with or without
```

```
* modification, are permitted provided that the following conditions
```

```
* are met:
```

```
* 1. Redistributions of source code must retain the above copyright
```

```
* notice, this list of conditions and the following disclaimer.
```

```
* 2. Redistributions in binary form must reproduce the above copyright
```

```
* notice, this list of conditions and the following disclaimer in the
```

```
* documentation and/or other materials provided with the distribution.
```

```
* 3. The name of the author may not be used to endorse or promote products
```

```
* derived from this software without specific prior written permission.
```

```
*
```

```
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
```

```
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
```

```
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
```

```
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
```

```
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
```

```
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
```

```
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
```

```

* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/
/** Template subgroup to add template rewrite output
* If debugging, then you'll also get STDbg.stg loaded.
*/

```

```

@outputFile.imports() ::= <<
<@super.imports()>
using Antlr3.ST;
using Antlr3.ST.Language;
>>

```

```

@genericParser.members() ::= <<
<@super.members()>
private StringTemplateGroup _templateGroup = new StringTemplateGroup("<name>Templates",
typeof(AngleBracketTemplateLexer) );

```

```

public StringTemplateGroup TemplateGroup
{
    get { return _templateGroup; }
    set { _templateGroup = value; }
}
>>

```

```

ruleReturnBaseType() ::= <%
Template<if(TREE_PARSER)>Tree<else>Parser<endif>RuleReturnScope\<StringTemplate, <labelType>>
%>

```

```

/** x+=rule when output=template */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabelElem(elem={<label>.Template},elemType="StringTemplate",...)>
>>

```

```

rewriteTemplate(alts) ::= <<

```

```

// TEMPLATE REWRITE
<if(backtracking)>
if (<actions.(actionScope).synpredgate>)
{
    <alts:rewriteTemplateAlt(); separator="else ">
    <if(rewriteMode)><replaceTextInLine()><endif>
}
<else>
<alts:rewriteTemplateAlt(); separator="else ">
<if(rewriteMode)><replaceTextInLine()><endif>

```

```

<endif>
>>

replaceTextInLine() ::= <<
<if(TREE_PARSER)>
((TokenRewriteStream)input.TokenStream).Replace(
input.TreeAdaptor.GetTokenStartIndex(retval.Start),
input.TreeAdaptor.GetTokenStopIndex(retval.Start),
retval.Template);
<else>
((TokenRewriteStream)input).Replace(
retval.Start.TokenIndex,
input.LT(-1).TokenIndex,
retval.Template);
<endif>
>>

rewriteTemplateAlt(it) ::= <<
// <it.description>
<if(it.pred)>
if (<it.pred>)
{
retval.Template = <it.alt>;
}<\n>
<else>
{
retval.Template = <it.alt>;
}<\n>
<endif>
>>

rewriteEmptyTemplate(alts) ::= <<
null;
>>

/** Invoke a template with a set of attribute name/value pairs.
* Set the value of the rule's template *after* having set
* the attributes because the rule's template might be used as
* an attribute to build a bigger template; you get a self-embedded
* template.
*/
rewriteExternalTemplate(name,args) ::= <%
TemplateGroup.GetInstanceOf("<name>"<optionalArguments(args)>)
%>

/** expr is a string expression that says what template to load */
rewriteIndirectTemplate(expr,args) ::= <%
TemplateGroup.GetInstanceOf(<expr><optionalArguments(args)>)

```

```
%>
```

```
/** Invoke an inline template with a set of attribute name/value pairs */  
rewriteInlineTemplate(args, template) ::= <%  
new StringTemplate(TemplateGroup, "<template>"<optionalArguments(args)>  
%>
```

```
optionalArguments(args) ::= <<  
<if(args)>,  
new Dictionary<string, object>() { <args.optionalArgument(); separator=", "> }  
<endif>  
>>
```

```
optionalArgument(it) ::= <<  
{ "<it.name>", <it.value> }  
>>
```

```
/** plain -> {foo} action */  
rewriteAction(action) ::= <<  
<action>  
>>
```

```
/** An action has %st.attrName=expr; or % {st}.attrName=expr; */  
actionSetAttribute(st,attrName,expr) ::= <<  
(<st>).SetAttribute("<attrName>",<expr>);  
>>
```

```
/** Translate % {stringExpr} */  
actionStringConstructor(stringExpr) ::= <<  
new StringTemplate(TemplateGroup,<stringExpr>)  
>>
```

Found in path(s):

```
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-  
jar/org/antlr/codegen/templates/CSharp3/ST.stg
```

No license file was found, but licenses were detected in source scan.

```
/*
```

```
[The "BSD license"]
```

```
Copyright (c) 2005-2006 Terence Parr
```

```
All rights reserved.
```

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright

notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

group AST;

```
@outputFile.imports() ::= <<
<@super.imports()>
>>
```

```
@genericParser.members() ::= <<
<@super.members()>
<parserMembers()>
>>
```

/** Add an adaptor property that knows how to build trees */

```
parserMembers() ::= <<
<!protected TreeAdaptor adaptor = new CommonTreeAdaptor();<\n>!>
setTreeAdaptor: function(adaptor) {
  this.adaptor = adaptor;
  <grammar.directDelegates: {g|<g:delegateName()>.setTreeAdaptor(this.adaptor);}>
},
getTreeAdaptor: function() {
  return this.adaptor;
},
>>
```

```
@returnScope.ruleReturnMembers() ::= <<
getTree: function() { return this.tree; }
>>
```

/** Add a variable to track rule's return AST */

```
ruleDeclarations() ::= <<
<super.ruleDeclarations()>
var root_0 = null;<\n>
>>
```

```

ruleLabelDefs() ::= <<
<super.ruleLabelDefs()>
<ruleDescriptor.tokenLabels:{ var <it.label.text>_tree=null;}; separator="\n">
<ruleDescriptor.tokenListLabels:{ var <it.label.text>_tree=null;}; separator="\n">
<ruleDescriptor.allTokenRefsInAltsWithRewrites
: {var stream_<it>=new org.antlr.runtime.tree.RewriteRuleTokenStream(this.adaptor,"token <it>");};
separator="\n">
<ruleDescriptor.allRuleRefsInAltsWithRewrites
: {var stream_<it>=new org.antlr.runtime.tree.RewriteRuleSubtreeStream(this.adaptor,"rule <it>");};
separator="\n">
>>

/** When doing auto AST construction, we must define some variables;
* These should be turned off if doing rewrites. This must be a "mode"
* as a rule could have both rewrite and AST within the same alternative
* block.
*/
@alt.declarations() ::= <<
<if(autoAST)>
<if(outerAlt)>
<if(!rewriteMode)>
root_0 = this.adaptor.nil();<\n>
<endif>
<endif>
<endif>
>>

// T r a c k i n g   R u l e   E l e m e n t s

/** ID and track it for use in a rewrite rule */
tokenRefTrack(token,label,elementIndex,terminalOptions) ::= <<
<tokenRefBang(...)> <! Track implies no auto AST construction!>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>stream_<token>.add(<label>);<\n>
>>

/** ids+=ID and track it for use in a rewrite rule; adds to ids *and*
* to the tracking list stream_ID for use in the rewrite.
*/
tokenRefTrackAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<tokenRefTrack(...)>
<listLabel(elem=label,...)>
>>

/** ^(ID ...) track for rewrite */
tokenRefRuleRootTrack(token,label,elementIndex,terminalOptions) ::= <<
<tokenRefBang(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>stream_<token>.add(<label>);<\n>

```

```

>>

/** Match ^(label+=TOKEN ...) track for rewrite */
tokenRefRuleRootTrackAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<tokenRefRuleRootTrack(...)>
<listLabel(elem=label,...)>
>>

wildcardTrack(label,elementIndex) ::= <<
<super.wildcard(...)>
>>

/** rule when output=AST and tracking for rewrite */
ruleRefTrack(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>stream_<rule.name>.add(<label>.getTree());
>>

/** x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefTrack(...)>
<listLabel(elem=label+".getTree()",...)>
>>

/** ^(rule ...) rewrite */
ruleRefRuleRootTrack(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>stream_<rule>.add(<label>.getTree());
>>

/** ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRootTrack(...)>
<listLabel(elem=label+".getTree()",...)>
>>

```

// R e w r i t e

```

rewriteCode(
  alts, description,
  referencedElementsDeep, // ALL referenced elements to right of ->
  referencedTokenLabels,
  referencedTokenListLabels,
  referencedRuleLabels,
  referencedRuleListLabels,
  referencedWildcardLabels,
  referencedWildcardListLabels,
  rewriteBlockLevel, enclosingTreeLevel, treeLevel) ::=

```

<<

```
// AST REWRITE
// elements: <referencedElementsDeep; separator=", ">
// token labels: <referencedTokenLabels; separator=", ">
// rule labels: <referencedRuleLabels; separator=", ">
// token list labels: <referencedTokenListLabels; separator=", ">
// rule list labels: <referencedRuleListLabels; separator=", ">
<if(backtracking)>
if ( <actions.(actionScope).synpredgate> ) {<\n>
<endif>
<prevRuleRootRef(>).tree = root_0;
<rewriteCodeLabels(>
root_0 = this.adaptor.nil();
<alts:rewriteAlt(> separator="else ">
<! if tree parser and rewrite=true !>
<if(TREE_PARSER)>
<if(rewriteMode)>
<prevRuleRootRef(>).tree = this.adaptor.rulePostProcessing(root_0);
this.input.replaceChildren(this.adaptor.getParent(retval.start),
    this.adaptor.getChildIndex(retval.start),
    this.adaptor.getChildIndex(_last),
    retval.tree);
<endif>
<endif>
<! if parser or tree-parser && rewrite!=true, we need to set result !>
<if(!TREE_PARSER)>
<prevRuleRootRef(>).tree = root_0;
<else>
<if(!rewriteMode)>
<prevRuleRootRef(>).tree = root_0;
<endif>
<endif>
<if(backtracking)>
}
<endif>
>>
```

```
rewriteCodeLabels() ::= <<
```

```
<referencedTokenLabels
```

```
 : {var stream_<it>=new org.antlr.runtime.tree.RewriteRule<rewriteElementType>Stream(this.adaptor,"token
<it>",<it>);};
 separator="\n"
```

```
>
```

```
<referencedTokenListLabels
```

```
 : {var stream_<it>=new org.antlr.runtime.tree.RewriteRule<rewriteElementType>Stream(this.adaptor,"token
<it>",<list_<it>);};
 separator="\n"
```

```

>
<referencedWildcardLabels
  :{var stream_<it>=new org.antlr.runtime.tree.RewriteRuleSubtreeStream(this.adaptor,"wildcard <it>",<it>);};
  separator="\n"
>
<referencedWildcardListLabels
  :{var stream_<it>=new org.antlr.runtime.tree.RewriteRuleSubtreeStream(this.adaptor,"wildcard <it>",<it>);};
  separator="\n"
>
<referencedRuleLabels
  :{var stream_<it>=new org.antlr.runtime.tree.RewriteRuleSubtreeStream(this.adaptor,"token
<it>",<it>!=null?<it>.tree:null);};
  separator="\n"
>
<referencedRuleListLabels
  :{var stream_<it>=new org.antlr.runtime.tree.RewriteRuleSubtreeStream(this.adaptor,"token <it>",<it>);};
  separator="\n"
>
>>

/** Generate code for an optional rewrite block; note it uses the deep ref'd element
 * list rather shallow like other blocks.
 */
rewriteOptionalBlock(
  alt,rewriteBlockLevel,
  referencedElementsDeep, // all nested refs
  referencedElements, // elements in immediately block; no nested blocks
  description) ::=
<<
// <fileName>:<description>
if ( <referencedElementsDeep:{el | stream_<el>.hasNext()}; separator="||"> ) {
  <alt>
}
<referencedElementsDeep:{el | stream_<el>.reset();<n>}>
>>

rewriteClosureBlock(
  alt,rewriteBlockLevel,
  referencedElementsDeep, // all nested refs
  referencedElements, // elements in immediately block; no nested blocks
  description) ::=
<<
// <fileName>:<description>
while ( <referencedElements:{el | stream_<el>.hasNext()}; separator="||"> ) {
  <alt>
}
<referencedElements:{el | stream_<el>.reset();<n>}>
>>

```

```

rewritePositiveClosureBlock(
    alt,rewriteBlockLevel,
    referencedElementsDeep, // all nested refs
    referencedElements, // elements in immediately block; no nested blocks
    description) ::=
<<
if ( !(<referencedElements:{el | stream_<el>.hasNext()}; separator="||">) ) {
    throw new org.antlr.runtime.tree.RewriteEarlyExitException();
}
while ( <referencedElements:{el | stream_<el>.hasNext()}; separator="||"> ) {
    <alt>
}
<referencedElements:{el | stream_<el>.reset();<\n>}>
>>

```

```

rewriteAlt(a) ::= <<
// <a.description>
<if(a.pred)>
if (<a.pred>) {
    <a.alt>
}<\n>
<else>
{
    <a.alt>
}<\n>
<endif>
>>

```

```

/** For empty rewrites: "r : ... -> ;" */
rewriteEmptyAlt() ::= "root_0 = null;"

```

```

rewriteTree(root,children,description,enclosingTreeLevel,treeLevel) ::= <<
// <fileName>:<description>
{
var root_<treeLevel> = this.adaptor.nil();
<root:rewriteElement()>
<children:rewriteElement()>
this.adaptor.addChild(root_<enclosingTreeLevel>, root_<treeLevel>);
}<\n>
>>

```

```

rewriteElementList(elements) ::= "<elements:rewriteElement()>"

```

```

rewriteElement(e) ::= <<
<@pregen()>
<e.el>
>>

```

```

/** Gen ID or ID[args] */
rewriteTokenRef(token,elementIndex,terminalOptions,args) ::= <<
this.adaptor.addChild(root_<treeLevel>, <createRewriteNodeFromElement(...)>);<\n>
>>

/** Gen $label ... where defined via label=ID */
rewriteTokenLabelRef(label,elementIndex) ::= <<
this.adaptor.addChild(root_<treeLevel>, stream_<label>.nextNode());<\n>
>>

/** Gen $label ... where defined via label+=ID */
rewriteTokenListLabelRef(label,elementIndex) ::= <<
this.adaptor.addChild(root_<treeLevel>, stream_<label>.nextNode());<\n>
>>

/** Gen ^($label ...) */
rewriteTokenLabelRefRoot(label,elementIndex) ::= <<
root_<treeLevel> = this.adaptor.becomeRoot(stream_<label>.nextNode(), root_<treeLevel>);<\n>
>>

/** Gen ^($label ...) where label+=... */
rewriteTokenListLabelRefRoot ::= rewriteTokenLabelRefRoot

/** Gen ^(ID ...) or ^(ID[args] ...) */
rewriteTokenRefRoot(token,elementIndex,terminalOptions,args) ::= <<
root_<treeLevel> = this.adaptor.becomeRoot(<createRewriteNodeFromElement(...)>, root_<treeLevel>);<\n>
>>

rewriteImaginaryTokenRef(args,token,terminalOptions,elementIndex) ::= <<
this.adaptor.addChild(root_<treeLevel>, <createImaginaryNode(tokenType=token, ...)>);<\n>
>>

rewriteImaginaryTokenRefRoot(args,token,terminalOptions,elementIndex) ::= <<
root_<treeLevel> = this.adaptor.becomeRoot(<createImaginaryNode(tokenType=token, ...)>,
root_<treeLevel>);<\n>
>>

/** plain -> {foo} action */
rewriteAction(action) ::= <<
root_0 = <action>;<\n>
>>

/** What is the name of the previous value of this rule's root tree? This
* let's us refer to $rule to mean previous value. I am reusing the
* variable 'tree' sitting in retval struct to hold the value of root_0 right
* before I set it during rewrites. The assign will be to retval.tree.
*/

```

```

prevRuleRootRef() ::= "retval"

rewriteRuleRef(rule) ::= <<
this.adaptor.addChild(root_<treeLevel>, stream_<rule>.nextTree());<\n>
>>

rewriteRuleRefRoot(rule) ::= <<
root_<treeLevel> = this.adaptor.becomeRoot(stream_<rule>.nextNode(), root_<treeLevel>);<\n>
>>

rewriteNodeAction(action) ::= <<
this.adaptor.addChild(root_<treeLevel>, <action>);<\n>
>>

rewriteNodeActionRoot(action) ::= <<
root_<treeLevel> = this.adaptor.becomeRoot(<action>, root_<treeLevel>);<\n>
>>

/** Gen $ruleLabel ... where defined via ruleLabel=rule */
rewriteRuleLabelRef(label) ::= <<
this.adaptor.addChild(root_<treeLevel>, stream_<label>.nextTree());<\n>
>>

/** Gen $ruleLabel ... where defined via ruleLabel+=rule */
rewriteRuleListLabelRef(label) ::= <<
this.adaptor.addChild(root_<treeLevel>, stream_<label>.nextTree());<\n>
>>

/** Gen ^($ruleLabel ...) where ruleLabel=rule */
rewriteRuleLabelRefRoot(label) ::= <<
root_<treeLevel> = this.adaptor.becomeRoot(stream_<label>.nextNode(), root_<treeLevel>);<\n>
>>

/** Gen ^($ruleLabel ...) where ruleLabel+=rule */
rewriteRuleListLabelRefRoot(label) ::= <<
root_<treeLevel> = this.adaptor.becomeRoot(stream_<label>.nextNode(), root_<treeLevel>);<\n>
>>

rewriteWildcardLabelRef(label) ::= <<
this.adaptor.addChild(root_<treeLevel>, stream_<label>.nextTree());<\n>
>>

createImaginaryNode(tokenType,terminalOptions,args) ::= <<
<if(terminalOptions.node)>
<! new MethodNode(IDLabel, args) !>
new <terminalOptions.node><tokenType><if(args)>, <args; separator=", "><endif>
<else>
this.adaptor.create(<tokenType>, <args; separator=", "><if(!args)>"<tokenType>"<endif>)

```

```

<endif>
>>

createRewriteNodeFromElement(token,terminalOptions,args) ::= <<
<if(terminalOptions.node)>
new <terminalOptions.node>(stream_<token>.nextToken(<if(args)>, <args; separator=", "><endif>)
<else>
<if(args)> <! must create new node from old !>
this.adaptor.create(<token>, <args; separator=", ">)
<else>
stream_<token>.nextNode()
<endif>
<endif>
>>

```

Found in path(s):

```
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-
jar/org/antlr/codegen/templates/JavaScript/AST.stg
```

No license file was found, but licenses were detected in source scan.

```

/*
* [The "BSD license"]
* Copyright (c) 2010 Terence Parr and Alan Condit
* Copyright (c) 2006 Kay Roepke (Objective-C runtime)
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products
* derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

```

Found in path(s):

* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-jar/org/antlr/codegen/ObjCTarget.java

No license file was found, but licenses were detected in source scan.

/*

* [The "BSD license"]

* Copyright (c) 2007-2008 Johannes Luber

* Copyright (c) 2005-2007 Kunle Odutola

* Copyright (c) 2011 Sam Harwell

* Copyright (c) 2011 Terence Parr

* All rights reserved.

*

* Redistribution and use in source and binary forms, with or without

* modification, are permitted provided that the following conditions

* are met:

* 1. Redistributions of source code must retain the above copyright

* notice, this list of conditions and the following disclaimer.

* 2. Redistributions in binary form must reproduce the above copyright

* notice, this list of conditions and the following disclaimer in the

* documentation and/or other materials provided with the distribution.

* 3. The name of the author may not be used to endorse or promote products

* derived from this software without specific prior written permission.

*

* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR

* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES

* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.

* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,

* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT

* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,

* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY

* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT

* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF

* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

csharpVisibilityMap ::= [

"private": "private",

"protected": "protected",

"public": "public",

"fragment": "private",

default: "private"

]

/** The overall file structure of a recognizer; stores methods for rules

* and cyclic DFAs plus support code.

*/

outputFile(LEXER,PARSER,TREE_PARSER, actionScope, actions,

```

docComment, recognizer,
name, tokens, tokenNames, rules, cyclicDFAs,
bitsets, buildTemplate, buildAST, rewriteMode, profile,
backtracking, synpreds, memoize, numRules,
fileName, ANTLRVersion, generatedTimestamp, trace,
scopes, superClass, literals) ::=
<<
//-----
//\<auto-generated>
// This code was generated by a tool.
// ANTLR Version: <ANTLRVersion>
//
// Changes to this file may cause incorrect behavior and will be lost if
// the code is regenerated.
//\</auto-generated>
//-----

// $ANTLR <ANTLRVersion> <fileName> <generatedTimestamp>

<if(trace)>
#define ANTLR_TRACE
<endif>
<@debugPreprocessor()>
// The variable 'variable' is assigned but its value is never used.
#pragma warning disable 168, 219
// Unreachable code detected.
#pragma warning disable 162
// Missing XML comment for publicly visible type or member 'Type_or_Member'
#pragma warning disable 1591

<actions.(actionScope).header>

<@imports>
using System.Collections.Generic;
using Antlr.Runtime;
using Antlr.Runtime.Misc;
<if(TREE_PARSER)>
using Antlr.Runtime.Tree;
using RewriteRuleITokenStream = Antlr.Runtime.Tree.RewriteRuleTokenStream;
<endif>
using ConditionalAttribute = System.Diagnostics.ConditionalAttribute;
<@end>
<if(actions.(actionScope).namespace)>
namespace <actions.(actionScope).namespace>
{
<endif>
<docComment>
<recognizer>

```

```

<if(actions.(actionScope).namespace)>

} // namespace <actions.(actionScope).namespace>
<endif>
>>

lexerInputStreamType() ::= <<
<actions.(actionScope).inputStreamType; null="ICharStream">
>>

lexer(grammar, name, tokens, scopes, rules, numRules, filterMode, labelType="CommonToken",
superClass={ <if(actions.(actionScope).superClass)><actions.(actionScope).superClass><else>Antlr.Runtime.Lexer
<endif>}) ::= <<
[System.CodeDom.Compiler.GeneratedCode("ANTLR", "<ANTLRVersion>")]
[System.CLSCompliant(false)]
<parserModifier(grammar=grammar, actions=actions)> partial class <grammar.recognizerName> :
<@superClassName><superClass><@end>
{
<tokens:{ it|public const int <it.name; format="id">=<it.type>;}; separator="\n">
<scopes: { it|<if(it.isDynamicGlobalScope)><globalAttributeScope(scope=it)><endif> }>
<actions.lexer.members>

// delegates
<grammar.delegates:
    { g|private <g.recognizerName> <g.delegateName()>;}; separator="\n">
// delegators
<grammar.delegators:
    { g|private <g.recognizerName> <g.delegateName()>;}; separator="\n">
<last(grammar.delegators):{ g|private <g.recognizerName> gParent;}>

<actions.(actionScope).ctorModifier; null="public"> <grammar.recognizerName>()<! needed by subclasses !>
{
    OnCreated();
}

<actions.(actionScope).ctorModifier; null="public"> <grammar.recognizerName>(<lexerInputStreamType()>
input<grammar.delegators:{ g|, <g.recognizerName> <g.delegateName()>}> )
: this(input, new RecognizerSharedState()<grammar.delegators:{ g|, <g.delegateName()>}>)
{
}

<actions.(actionScope).ctorModifier; null="public"> <grammar.recognizerName>(<lexerInputStreamType()> input,
RecognizerSharedState state<grammar.delegators:{ g|, <g.recognizerName> <g.delegateName()>}>)
: base(input, state)
{
<if(memoize)>
<if(grammar.grammarIsRoot)>
    state.ruleMemo = new System.Collections.Generic.Dictionary<int, int>[<numRules>+1];<\n><! index from 1..n !>

```

```

<endif>
<endif>
    <grammar.directDelegates:
        {g|<g:delegateName()> = new <g.recognizerName>(input, this.state<trunc(g.delegates):{p|,
<p:delegateName()>>, this);}; separator="\n">
    <grammar.delegates:
        {g|this.<g:delegateName()> = <g:delegateName()>;}; separator="\n">
    <last(grammar.delegates):{g|gParent = <g:delegateName()>;}>

    OnCreated();
}
public override string GrammarFileName { get { return "<fileName>"; } }

private static readonly bool[] decisionCanBacktrack = new bool[0];

<if(grammar.hasDelegates)>
public override <lexerInputStreamType()> CharStream
{
    get
    {
        return base.CharStream;
    }
    set
    {
        base.CharStream = value;
        <grammar.directDelegates:
            {g|<g:delegateName()> = new <g.recognizerName>(input, state<trunc(g.delegates):{p|, <p:delegateName()>>,
this);}; separator="\n">
        <grammar.delegates:
            {g|this.<g:delegateName()> = <g:delegateName()>;}; separator="\n">
        <last(grammar.delegates):{g|gParent = <g:delegateName()>;}>
    }
}

<if(grammar.delegates)>
public override void SetState(RecognizerSharedState state)
{
    base.SetState(state);
    <grammar.delegates:{g|<g:delegateName()>.SetState(state);}; separator="\n">
}
<endif>

<endif>
<if(filterMode)>
<filteringNextToken()>
<endif>

[Conditional("ANTLR_TRACE")]

```

```

protected virtual void OnCreated() {}
[Conditional("ANTLR_TRACE")]
protected virtual void EnterRule(string ruleName, int ruleIndex) {}
[Conditional("ANTLR_TRACE")]
protected virtual void LeaveRule(string ruleName, int ruleIndex) {}

<rules; separator="\n">

<insertLexerSynpreds(synpreds)>

#region DFA
<cyclicDFAs:{ dfa | DFA<dfa.decisionNumber> dfa<dfa.decisionNumber>; }; separator="\n">

protected override void InitDFAs()
{
    base.InitDFAs();
    <cyclicDFAs:{ dfa | dfa<dfa.decisionNumber> = new DFA<dfa.decisionNumber>(this<if(dfa.specialStateSTs)>,
SpecialStateTransition<dfa.decisionNumber><endif>); }; separator="\n">
}

<cyclicDFAs:cyclicDFA()> <! dump tables for all DFA !>
#endregion

}
>>

/** A override of Lexer.nextToken() that backtracks over mTokens() looking
 * for matches. No error can be generated upon error; just rewind, consume
 * a token and then try again. backtracking needs to be set as well.
 * Make rule memoization happen only at levels above 1 as we start mTokens
 * at backtracking==1.
 */
filteringNextToken() ::= <<
public override IToken NextToken()
{
    while (true)
    {
        if (input.LA(1) == CharStreamConstants.EndOfFile)
        {
            IToken eof = new CommonToken((ICharStream)input, CharStreamConstants.EndOfFile, TokenChannels.Default,
input.Index, input.Index);
            eof.Line = Line;
            eof.CharPositionInLine = CharPositionInLine;
            return eof;
        }
        state.token = null;
        state.channel = TokenChannels.Default;
        state.tokenStartCharIndex = input.Index;

```

```

state.tokenStartCharPositionInLine = input.CharPositionInLine;
state.tokenStartLine = input.Line;
state.text = null;
try
{
int m = input.Mark();
state.backtracking=1;<! means we won't throw slow exception !>
state.failed=false;
mTokens();
state.backtracking=0;
<! mTokens backtracks with synpred at backtracking==2
and we set the synpredgate to allow actions at level 1. !>
if (state.failed)
{
input.Rewind(m);
input.Consume();<! advance one char and try again !>
}
else
{
Emit();
return state.token;
}
}
catch (RecognitionException re)
{
// shouldn't happen in backtracking mode, but...
ReportError(re);
Recover(re);
}
}

public override void Memoize(IIntStream input, int ruleIndex, int ruleStartIndex)
{
if (state.backtracking > 1)
base.Memoize(input, ruleIndex, ruleStartIndex);
}

public override bool AlreadyParsedRule(IIntStream input, int ruleIndex)
{
if (state.backtracking > 1)
return base.AlreadyParsedRule(input, ruleIndex);

return false;
}
>>

actionGate() ::= "state.backtracking == 0"

```

```

filteringActionGate() ::= "state.backtracking == 1"

/** How to generate a parser */
genericParser(grammar, name, scopes, tokens, tokenNames, rules, numRules,
    bitsets, inputStreamType, superClass,
    labelType, members, rewriteElementType,
    filterMode, ASTLabelType="object") ::= <<
[System.CodeDom.Compiler.GeneratedCode("ANTLR", "<ANTLRVersion>")]
[System.CLSCompliant(false)]
<parserModifier(grammar=grammar, actions=actions)> partial class <grammar.recognizerName> :
<@superClassName><superClass><@end>
{
<if(grammar.grammarIsRoot)>
internal static readonly string[] tokenNames = new string[] {
    "\<invalid>", "\<EOR>", "\<DOWN>", "\<UP>", <tokenNames; separator=", ">
};
<endif>
<tokens:{it|public const int <it.name; format="id">=<it.type>;}; separator="\n">

<if(grammar.delegates)>
// delegates
<grammar.delegates:
    {g|private <g.recognizerName> <g:delegateName(>);}; separator="\n">
<endif>
<if(grammar.delegators)>
// delegators
<grammar.delegators:
    {g|private <g.recognizerName> <g:delegateName(>);}; separator="\n">
<last(grammar.delegators):{g|private <g.recognizerName> gParent;}>
<endif>

<if(grammar.delegates)>
public override void SetState(RecognizerSharedState state)
{
    base.SetState(state);
    <grammar.delegates:{g|<g:delegateName(>.SetState(state);}; separator="\n">
}

<if(TREE_PARSER)>
public override void SetTreeNodeStream(ITreeNodeStream input)
{
    base.SetTreeNodeStream(input);
    <grammar.delegates:{g|<g:delegateName(>.SetTreeNodeStream(input);}; separator="\n">
}
<endif>
<endif>

```

```

<scopes:{it|<if(it.isDynamicGlobalScope)><globalAttributeScope(scope=it)><endif> }>
<@members()>

public override string[] TokenNames { get { return
<grammar.composite.rootGrammar.recognizerName>.tokenNames; } }
public override string GrammarFileName { get { return "<fileName>"; } }

<members>

[Conditional("ANTLR_TRACE")]
protected virtual void OnCreated() {}
[Conditional("ANTLR_TRACE")]
protected virtual void EnterRule(string ruleName, int ruleIndex) {}
[Conditional("ANTLR_TRACE")]
protected virtual void LeaveRule(string ruleName, int ruleIndex) {}

#region Rules
<rules; separator="\n">
#endregion Rules

<if(grammar.delegatedRules)>
<! generate rule/method definitions for imported rules so they
appear to be defined in this recognizer. !>
#region Delegated rules
<grammar.delegatedRules:{ruleDescriptor|
<ruleModifier(grammar=grammar,ruleDescriptor=ruleDescriptor)> <return Type(ruleDescriptor)>
<ruleDescriptor.name; format="id"><ruleDescriptor.parameterScope:parameterScope()> <!throws
RecognitionException !>{ <if(ruleDescriptor.hasReturnValue)>return
<endif><ruleDescriptor.grammar:delegateName()>.<ruleDescriptor.name;
format="id"><ruleDescriptor.parameterScope><ruleDescriptor.parameterScope.attributes:{a|a.name;
format="id">}; separator=", "><endif>}; \}}; separator="\n">
#endregion Delegated rules
<endif>

<insertSynpreds(synpreds)>

<if(cyclicDFAs)>
#region DFA
<cyclicDFAs:{ dfa | private DFA<dfa.decisionNumber> dfa<dfa.decisionNumber>;}; separator="\n">

protected override void InitDFAs()
{
base.InitDFAs();
<cyclicDFAs:{ dfa | dfa<dfa.decisionNumber> = new DFA<dfa.decisionNumber>( this<if(dfa.specialStateSTs)>,
SpecialStateTransition<dfa.decisionNumber><endif> );}; separator="\n">
}

<cyclicDFAs:cyclicDFA()><! dump tables for all DFA !>

```

```

#endregion DFA
<endif>

<if(bitsets)>
#region Follow sets
private static class Follow
{
  <bitsets: {it|bitset(name={_<it.name>_in_<it.inName><it.tokenIndex>}, words64=it.bits)>}; separator="\n">
}
#endregion Follow sets
<endif>
}
>>

@genericParser.members() ::= <<
#if ANTLR_DEBUG
private static readonly bool[] decisionCanBacktrack =
new bool[]
{
  false, // invalid decision
  <grammar.decisions: {d | <d.dfa.hasSynPred>}; wrap="\n", separator=", ">
};
#else
private static readonly bool[] decisionCanBacktrack = new bool[0];
#endif
<! WARNING. bug in ST: this is cut-n-paste into Dbg.stg !>
<actions.(actionScope).ctorModifier; null="public"> <grammar.recognizerName>(<inputStreamType>
input<grammar.delegates: {g|, <g.recognizerName> <g.delegateName()>}>)
: this(input, new RecognizerSharedState()<grammar.delegates: {g|, <g.delegateName()>}>)
{
}
<actions.(actionScope).ctorModifier; null="public"> <grammar.recognizerName>(<inputStreamType> input,
RecognizerSharedState state<grammar.delegates: {g|, <g.recognizerName> <g.delegateName()>}>)
: base(input, state)
{
<if(grammar.directDelegates)>
<grammar.directDelegates:
{g|<g.delegateName()> = new <g.recognizerName>(input, state<trunc(g.delegates): {p|, <p.delegateName()>}>,
this);}; separator="\n">
<endif>
<if(grammar.indirectDelegates)>
<grammar.indirectDelegates: {g | <g.delegateName()> = <g.delegator.delegateName()>.<g.delegateName()>};
separator="\n">
<endif>
<if(grammar.delegates)>
<last(grammar.delegates): {g|gParent = <g.delegateName()>};>
<endif>
<parserCtorBody()>

```

```

    OnCreated();
}
>>

// imported grammars are 'public' (can't be internal because their return scope classes must be accessible)
parserModifier(grammar, actions) ::= <<
<if(grammar.grammarIsRoot)><actions.(actionScope).modifier; null="public"><else>public<endif>
>>

parserCtorBody() ::= <<
<if(memoize)>
<if(grammar.grammarIsRoot)>
this.state.ruleMemo = new System.Collections.Generic.Dictionary<int,
int>[<length(grammar.allImportedRules)>+1];<\n><! index from 1..n !>
<endif>
<endif>
<grammar.delegators:
{g|this.<g:delegateName()> = <g:delegateName()>;}; separator="\n">
>>

parser(grammar, name, scopes, tokens, tokenNames, rules, numRules, bitsets,
    ASTLabelType="object",
superClass={ <if(actions.(actionScope).superClass)><actions.(actionScope).superClass><else>Antlr.Runtime.Parser
<endif>}, labelType="IToken",
    members={ <actions.parser.members>}) ::= <<
<genericParser(inputStreamType="ITokenStream", rewriteElementType="IToken", ...)>
>>

/** How to generate a tree parser; same as parser except the input
 * stream is a different type.
 */
treeParser(grammar, name, scopes, tokens, tokenNames, globalAction, rules,
    numRules, bitsets, filterMode, labelType={ <ASTLabelType>}, ASTLabelType="object",
superClass={ <if(actions.(actionScope).superClass)><actions.(actionScope).superClass><else>Antlr.Runtime.Tree.<
if(filterMode)><if(buildAST)>TreeRewriter<else>TreeFilter<endif><else>TreeParser<endif><endif>},
    members={ <actions.treeparser.members>}) ::= <<
<genericParser(inputStreamType="ITreeNodeStream", rewriteElementType="Node", ...)>
>>

/** A simpler version of a rule template that is specific to the imaginary
 * rules created for syntactic predicates. As they never have return values
 * nor parameters etc..., just give simplest possible method. Don't do
 * any of the normal memoization stuff in here either; it's a waste.
 * As predicates cannot be inlined into the invoking rule, they need to
 * be in a rule by themselves.
 */
synpredRule(ruleName, ruleDescriptor, block, description, nakedBlock) ::=
<<

```

```

[Conditional("ANTLR_TRACE")]
protected virtual void EnterRule_<ruleName>_fragment() {}
[Conditional("ANTLR_TRACE")]
protected virtual void LeaveRule_<ruleName>_fragment() {}

// $ANTLR start <ruleName>
<ruleModifier(grammar,ruleDescriptor)> void
<ruleName>_fragment(<ruleDescriptor.parameterScope:parameterScope(>
{
    <ruleLabelDefs(>
    EnterRule_<ruleName>_fragment();
    EnterRule("<ruleName>_fragment", <ruleDescriptor.index>);
    TraceIn("<ruleName>_fragment", <ruleDescriptor.index>);
    try
    {
        <block>
    }
    finally
    {
        TraceOut("<ruleName>_fragment", <ruleDescriptor.index>);
        LeaveRule("<ruleName>_fragment", <ruleDescriptor.index>);
        LeaveRule_<ruleName>_fragment();
    }
}
// $ANTLR end <ruleName>
>>

```

```

insertLexerSynpreds(synpreds) ::= <<
<insertSynpreds(synpreds)>
>>

```

```

insertSynpreds(synpreds) ::= <<
<if(synpreds)>
#region Synpreds
private bool EvaluatePredicate(System.Action fragment)
{
    bool success = false;
    state.backtracking++;
    <@start(>
    try { DebugBeginBacktrack(state.backtracking);
    int start = input.Mark();
    try
    {
        fragment();
    }
    catch ( RecognitionException re )
    {
        System.Console.Error.WriteLine("impossible: "+re);
    }
}

```

```

    }
    success = !state.failed;
    input.Rewind(start);
    } finally { DebugEndBacktrack(state.backtracking, success); }
    <@stop()>
    state.backtracking--;
    state.failed=false;
    return success;
    }
#endregion Synpreds
<endif>
>>

ruleMemoization(name) ::= <<
<if(memoize)>
if (state.backtracking > 0 && AlreadyParsedRule(input, <ruleDescriptor.index>)) { <returnFromRule()> }
<endif>
>>

/** How to test for failure and return from rule */
checkRuleBacktrackFailure() ::= <<
<if(backtracking)>if (state.failed) <returnFromRule()><endif>
>>

/** This rule has failed, exit indicating failure during backtrack */
ruleBacktrackFailure() ::= <<
<if(backtracking)>if (state.backtracking>0) { state.failed=true; <returnFromRule()>}<endif>
>>

ruleWrapperMap ::= [
"bottomup":{<ruleWrapperBottomup()>},
"topdown":{<ruleWrapperTopdown()>},
default:""
]

ruleWrapperBottomup() ::= <<
<if(TREE_PARSER && filterMode)>
protected override <if(buildAST)>IAstRuleReturnScope<else>void<endif> Bottomup() { return bottomup(); }
<endif>
>>

ruleWrapperTopdown() ::= <<
<if(TREE_PARSER && filterMode)>
protected override <if(buildAST)>IAstRuleReturnScope<else>void<endif> Topdown() { return topdown(); }
<endif>
>>

/** How to generate code for a rule. This includes any return type

```

```

* data aggregates required for multiple return values.
*/
rule(ruleName,ruleDescriptor,block,emptyRule,description,exceptions,finally,memoize) ::= <<
<ruleAttributeScope(scope=ruleDescriptor.ruleScope)>
<returnScope(ruleDescriptor.returnScope)>

[Conditional("ANTLR_TRACE")]
protected virtual void EnterRule_<ruleName>() {}
[Conditional("ANTLR_TRACE")]
protected virtual void LeaveRule_<ruleName>() {}
<ruleWrapperMap.(ruleName)>
// $ANTLR start "<ruleName>"
// <fileName>:<description>
[GrammarRule("<ruleName>")]
<ruleModifier(grammar=grammar,ruleDescriptor=ruleDescriptor)> <returnType(ruleDescriptor)> <ruleName;
format="id">(<ruleDescriptor.parameterScope:parameterScope(>)
{
EnterRule_<ruleName>();
EnterRule("<ruleName>", <ruleDescriptor.index>);
TraceIn("<ruleName>", <ruleDescriptor.index>);
<ruleScopeSetUp(>
<ruleDeclarations(>
<ruleLabelDefs(>
<ruleDescriptor.actions.init>
try { DebugEnterRule(GrammarFileName, "<ruleName>");
DebugLocation(<ruleDescriptor.tree.line>, <ruleDescriptor.EORNode.charPositionInLine>);
<@preamble(>
try
{
<ruleMemoization(name=ruleName)>
<block>
<ruleCleanUp(>
<(ruleDescriptor.actions.after):execAction(>
}
<if(exceptions)>
<exceptions: {e|<catch(decl=e.decl,action=e.action)><\n>}>
<else>
<if(!emptyRule)>
<if(actions.(actionScope).rulecatch)>
<actions.(actionScope).rulecatch>
<else>
catch (RecognitionException re)
{
ReportError(re);
Recover(input,re);
<@setErrorReturnValue(>
}
<endif>

```

```

<endif>
<endif>
finally
{
  TraceOut("<ruleName>", <ruleDescriptor.index>);
  LeaveRule("<ruleName>", <ruleDescriptor.index>);
  LeaveRule_<ruleName>();
  <memoize()>
  <ruleScopeCleanUp()>
  <finally>
}
DebugLocation(<ruleDescriptor.EORNode.line>, <ruleDescriptor.EORNode.charPositionInLine>);
} finally { DebugExitRule(GrammarFileName, "<ruleName>"); }
<@postamble()>
<returnFromRule()><\n>
}
// $ANTLR end "<ruleName>"
>>

// imported grammars need to have internal rules
ruleModifier(grammar,ruleDescriptor) ::= <<
<if(grammar.grammarIsRoot)><csharpVisibilityMap.(ruleDescriptor.modifier);
null="private"><else>internal<endif>
>>

// imported grammars need to have public return scopes
returnScopeModifier(grammar,ruleDescriptor) ::= <<
<if(grammar.grammarIsRoot)><csharpVisibilityMap.(ruleDescriptor.modifier);
null="private"><else>public<endif>
>>

catch(decl,action) ::= <<
catch (<e.decl>)
{
  <e.action>
}
>>

ruleDeclarations() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<returnType(ruleDescriptor)> retval = new <returnType(ruleDescriptor)>();
retval.Start = (<labelType>)input.LT(1);
<elseif(ruleDescriptor.returnScope)>
<ruleDescriptor.returnScope.attributes: { a |
<a.type> <a.name; format="id"> = <if(a.initValue)><a.initValue><else><initValue(a.type)><endif>;
}>
<endif>
<if(memoize)>

```

```

int <ruleDescriptor.name>_startIndex = input.Index;
<endif>
>>

ruleScopeSetUp() ::= <<
<ruleDescriptor.useScopes: {it|<it>_stack.Push(new <it>_scope());<it>_scopeInit(<it>_stack.Peek());};
separator="\n">
<ruleDescriptor.ruleScope: {it|<it.name>_stack.Push(new
<it.name>_scope());<it.name>_scopeInit(<it.name>_stack.Peek());}; separator="\n">
>>

ruleScopeCleanUp() ::= <<
<ruleDescriptor.useScopes: {it|<it>_scopeAfter(<it>_stack.Peek());<it>_stack.Pop();}; separator="\n">
<ruleDescriptor.ruleScope: {it|<it.name>_scopeAfter(<it.name>_stack.Peek());<it.name>_stack.Pop();};
separator="\n">
>>

ruleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,ruleDescriptor.tokenListLabels,ruleDescriptor.wildcardTreeLabels,ruleDescriptor.wildcardTreeListLabels]
: {it|<labelType> <it.label.text> = default(<labelType>);}; separator="\n"
>
<ruleDescriptor.tokenListLabels
: {it|List<<labelType>> list_<it.label.text> = null;}; separator="\n"
>
<[ruleDescriptor.ruleListLabels,ruleDescriptor.wildcardTreeListLabels]
: {it|List<<ASTLabelType>> list_<it.label.text> = null;}; separator="\n"
>
<ruleDescriptor.ruleLabels:ruleLabelDef(); separator="\n">
<ruleDescriptor.ruleListLabels:ruleLabelDef(); separator="\n">
>>

lexerRuleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,
ruleDescriptor.tokenListLabels,
ruleDescriptor.ruleLabels]
: {it|<labelType> <it.label.text> = default(<labelType>);}; separator="\n"
>
<[ruleDescriptor.charListLabels,
ruleDescriptor.charLabels]
: {it|int <it.label.text> = 0;}; separator="\n"
>
<[ruleDescriptor.tokenListLabels,
ruleDescriptor.ruleListLabels]
: {it|List<<labelType>> list_<it.label.text> = null;}; separator="\n"
>
<ruleDescriptor.charListLabels: {it|List<int> list_<it.label.text> = null;}; separator="\n"
>

```

```
>>
```

```
returnFromRule() ::= <%  
return  
<if(!ruleDescriptor.isSynPred)>  
<if(ruleDescriptor.hasReturnValue)>  
<if(ruleDescriptor.hasSingleReturnValue)>  
<! This comment is a hack to make sure the following  
  single space appears in the output. !> <ruleDescriptor.singleValueReturnName>  
<else>  
<!!> retval  
<endif>  
<endif>  
<endif>  
;  
>%>
```

```
ruleCleanUp() ::= <<  
<if(ruleDescriptor.hasMultipleReturnValues)>  
<if(!TREE_PARSER)>  
retval.Stop = (<labelType>)input.LT(-1);  
<endif>  
<endif>  
>>
```

```
memoize() ::= <<  
<if(memoize)>  
<if(backtracking)>  
if (state.backtracking > 0) { Memoize(input, <ruleDescriptor.index>, <ruleDescriptor.name>_StartIndex); }  
<endif>  
<endif>  
>>
```

```
/** How to generate a rule in the lexer; naked blocks are used for  
* fragment rules.  
*/
```

```
lexerRule(ruleName,nakedBlock,ruleDescriptor,block,memoize) ::= <<
```

```
[Conditional("ANTLR_TRACE")]  
protected virtual void EnterRule_<ruleName>() {}  
[Conditional("ANTLR_TRACE")]  
protected virtual void LeaveRule_<ruleName>() {}
```

```
// $ANTLR start "<ruleName>"  
[GrammarRule("<ruleName>")]  
<ruleModifier(grammar=grammar,ruleDescriptor=ruleDescriptor)> void  
m<ruleName>(<ruleDescriptor.parameterScope:parameterScope(>))  
{
```

```

EnterRule_<ruleName>();
EnterRule("<ruleName>", <ruleDescriptor.index>);
TraceIn("<ruleName>", <ruleDescriptor.index>);
  <ruleScopeSetUp()>
  <ruleDeclarations()>
  try
  {
<if(nakedBlock)>
  <ruleMemoization(name=ruleName)>
  <lexerRuleLabelDefs()>
  <ruleDescriptor.actions.init>
  <block>
<else>
  int _type = <ruleName>;
  int _channel = DefaultTokenChannel;
  <ruleMemoization(name=ruleName)>
  <lexerRuleLabelDefs()>
  <ruleDescriptor.actions.init>
  <block>
  <ruleCleanUp()>
  state.type = _type;
  state.channel = _channel;
  <(ruleDescriptor.actions.after):execAction()>
<endif>
  }
  finally
  {
  TraceOut("<ruleName>", <ruleDescriptor.index>);
  LeaveRule("<ruleName>", <ruleDescriptor.index>);
  LeaveRule_<ruleName>();
  <ruleScopeCleanUp()>
  <memoize()>
  }
  }
// $ANTLR end "<ruleName>"
>>

/** How to generate code for the implicitly-defined lexer grammar rule
 * that chooses between lexer rules.
 */
tokensRule(ruleName,nakedBlock,args,block,ruleDescriptor) ::= <<

public override void mTokens()
{
  <block><\n>
}
>>

```

// S U B R U L E S

/** A (...) subrule with multiple alternatives */

```
block(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
// <fileName>:<description>
int alt<decisionNumber>=<maxAlt>;
<decls>
<@predecision()>
try { DebugEnterSubRule(<decisionNumber>);
try { DebugEnterDecision(<decisionNumber>, decisionCanBacktrack[<decisionNumber>]);
<decision>
} finally { DebugExitDecision(<decisionNumber>); }
<@postdecision()>
<@prebranch()>
switch (alt<decisionNumber>)
{
<alts:{a|altSwitchCase(i,a)}>
}
} finally { DebugExitSubRule(<decisionNumber>); }
<@postbranch()>
>>
```

/** A rule block with multiple alternatives */

```
ruleBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
// <fileName>:<description>
int alt<decisionNumber>=<maxAlt>;
<decls>
<@predecision()>
try { DebugEnterDecision(<decisionNumber>, decisionCanBacktrack[<decisionNumber>]);
<decision>
} finally { DebugExitDecision(<decisionNumber>); }
<@postdecision()>
switch (alt<decisionNumber>)
{
<alts:{a|altSwitchCase(i,a)}>
}
>>
```

ruleBlockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<

```
// <fileName>:<description>
<decls>
<@prealt()>
DebugEnterAlt(1);
<alts>
<@postalt()>
>>
```

/** A special case of a (...) subrule with a single alternative */

```

blockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
// <fileName>:<description>
<decls>
<@prealt()>
DebugEnterAlt(1);
<alts>
<@postalt()>
>>

/** A (..)+ block with 1 or more alternatives */
positiveClosureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,descriptio
n) ::= <<
// <fileName>:<description>
int cnt<decisionNumber>=0;
<decls>
<@preloop()>
try { DebugEnterSubRule(<decisionNumber>);
while (true)
{
int alt<decisionNumber>=<maxAlt>;
<@predecision()>
try { DebugEnterDecision(<decisionNumber>, decisionCanBacktrack[<decisionNumber>]);
<decision>
} finally { DebugExitDecision(<decisionNumber>); }
<@postdecision()>
switch (alt<decisionNumber>)
{
<alts:{a|<altSwitchCase(i,a)>}>
default:
if (cnt<decisionNumber> >= 1)
goto loop<decisionNumber>;

<ruleBacktrackFailure()>
EarlyExitException eee<decisionNumber> = new EarlyExitException( <decisionNumber>, input );
DebugRecognitionException(eee<decisionNumber>);
<@earlyExitException()>
throw eee<decisionNumber>;
}
cnt<decisionNumber>++;
}
loop<decisionNumber>:
;

} finally { DebugExitSubRule(<decisionNumber>); }
<@postloop()>
>>

positiveClosureBlockSingleAlt ::= positiveClosureBlock

```

```

/** A (..)* block with 1 or more alternatives */
closureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::=
<<
// <fileName>:<description>
<decls>
<@preloop()>
try { DebugEnterSubRule(<decisionNumber>);
while (true)
{
int alt<decisionNumber>=<maxAlt>;
<@predecision()>
try { DebugEnterDecision(<decisionNumber>, decisionCanBacktrack[<decisionNumber>]);
<decision>
} finally { DebugExitDecision(<decisionNumber>); }
<@postdecision()>
switch ( alt<decisionNumber> )
{
<alts:{a|<altSwitchCase(i,a)>}>
default:
goto loop<decisionNumber>;
}
}

loop<decisionNumber>:
;

} finally { DebugExitSubRule(<decisionNumber>); }
<@postloop()>
>>

closureBlockSingleAlt ::= closureBlock

/** Optional blocks (x)? are translated to (x|) by before code generation
* so we can just use the normal block template
*/
optionalBlock ::= block

optionalBlockSingleAlt ::= block

/** A case in a switch that jumps to an alternative given the alternative
* number. A DFA predicts the alternative and then a simple switch
* does the jump to the code that actually matches that alternative.
*/
altSwitchCase(altNum,alt) ::= <<
case <altNum>:
<@prealt()>
DebugEnterAlt(<altNum>);

```

```

<alt>
break;<\n>
>>

/** An alternative is just a list of elements; at outermost level */
alt(elements,altNum,description,autoAST,outerAlt,treeLevel,rew) ::= <<
// <fileName>:<description>
{
<@declarations()>
<elements:element()>
<rew>
<@cleanup()>
}
>>

/** What to emit when there is no rewrite. For auto build
* mode, does nothing.
*/
noRewrite(rewriteBlockLevel, treeLevel) ::= ""

// E L E M E N T S

/** Dump the elements one per line */
element(it) ::= <%
<@prematch()>
DebugLocation(<it.line>, <it.pos>);<\n>
<it.el><\n>
%>

/** match a token optionally with a label in front */
tokenRef(token,label,elementIndex,terminalOptions={ }) ::= <<
<if(label)><label>=<(<labelType>)<endif>Match(input,<token>,Follow._<token>_in_<ruleName><elementIndex>);
<checkRuleBacktrackFailure()>
>>

/** ids+=ID */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions={ }) ::= <<
<tokenRef(...)>
<listLabelElem(elem=label,elemType=labelType,...)>
>>

listLabelElem(label,elem,elemType) ::= <<
if (list_<label>==null) list_<label>=new List<<elemType; null={<labelType>}>>();
list_<label>.Add(<elem>);<\n>
>>

/** match a character */
charRef(char,label) ::= <<

```

```

<if(label)>
<label> = input.LA(1);<\n>
<endif>
Match(<char>); <checkRuleBacktrackFailure()>
>>

/** match a character range */
charRangeRef(a,b,label) ::= <<
<if(label)>
<label> = input.LA(1);<\n>
<endif>
MatchRange(<a>,<b>); <checkRuleBacktrackFailure()>
>>

/** For now, sets are interval tests and must be tested inline */
matchSet(s,label,elementIndex,postmatchCode="",terminalOptions={}) ::= <<
<if(label)>
<matchSetLabel()>
<endif>
if (<s>)
{
input.Consume();
<postmatchCode>
<if(!LEXER)>state.errorRecovery=false;<endif><if(backtracking)>state.failed=false;<endif>
}
else
{
<ruleBacktrackFailure()>
MismatchedSetException mse = new MismatchedSetException(null,input);
DebugRecognitionException(mse);
<@mismatchedSetException()>
<if(LEXER)>
Recover(mse);
throw mse;
<else>
throw mse;
<! use following code to make it recover inline; remove throw mse;
recoverFromMismatchedSet(input,mse,Follow._set_in_<ruleName><elementIndex>);
!>
<endif>
}<\n>
>>

matchSetUnchecked(s,label,elementIndex,postmatchCode=false) ::= <%
<if(label)>
<matchSetLabel()><\n>
<endif>
input.Consume();<\n>

```

```

<if(postmatchCode)>
<postmatchCode><\n>
<endif>
<if(!LEXER)>state.errorRecovery=false;<endif><if(backtracking)>state.failed=false;<endif>
%>

matchSetLabel() ::= <%
<if(LEXER)>
<label>= input.LA(1);
<else>
<label>=(<labelType>)input.LT(1);
<endif>
%>

matchRuleBlockSet ::= matchSet

matchSetAndListLabel(s,label,elementIndex,postmatchCode) ::= <<
<matchSet(...)>
<listLabelElem(elem=label,elemType=labelType,...)>
>>

/** Match a string literal */
lexerStringRef(string,label,elementIndex) ::= <%
<if(label)>
int <label>Start = CharIndex;<\n>
Match(<string>); <checkRuleBacktrackFailure()><\n>
int <label>StartLine<elementIndex> = Line;<\n>
int <label>StartCharPos<elementIndex> = CharPositionInLine;<\n>
<label> = new <labelType>(input, TokenType.Invalid, TokenChannels.Default, <label>Start, CharIndex-1);<\n>
<label>.Line = <label>StartLine<elementIndex>;<\n>
<label>.CharPositionInLine = <label>StartCharPos<elementIndex>;
<else>
Match(<string>); <checkRuleBacktrackFailure()><\n>
<endif>
%>

wildcard(token,label,elementIndex,terminalOptions={ }) ::= <<
<if(label)>
<label>=(<labelType>)input.LT(1);<\n>
<endif>
MatchAny(input); <checkRuleBacktrackFailure()>
>>

wildcardAndListLabel(token,label,elementIndex,terminalOptions={ }) ::= <<
<wildcard(...)>
<listLabelElem(elem=label,elemType=labelType,...)>
>>

```

```

/** Match . wildcard in lexer */
wildcardChar(label, elementIndex) ::= <<
<if(label)>
<label> = input.LA(1);<\n>
<endif>
MatchAny(); <checkRuleBacktrackFailure()>
>>

wildcardCharListLabel(label, elementIndex) ::= <<
<wildcardChar(...)>
<listLabelElem(elem=label,elemType=labelType,...)>
>>

/** Match a rule reference by invoking it possibly with arguments
 * and a return value or values. The 'rule' argument was the
 * target rule name, but now is type Rule, whose toString is
 * same: the rule name. Now though you can access full rule
 * descriptor stuff.
 */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
PushFollow(Follow._<rule.name>_in_<ruleName><elementIndex>);
<if(label)><label>=<endif><if(scope)><scope.delegateName()>.<endif><rule.name; format="id">(<args;
separator=", ">);
PopFollow();
<checkRuleBacktrackFailure()>
>>

/** ids+=r */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabelElem(elem=label,elemType={<ASTLabelType>},...)>
>>

/** A lexer rule reference.
 *
 * The 'rule' argument was the target rule name, but now
 * is type Rule, whose toString is same: the rule name.
 * Now though you can access full rule descriptor stuff.
 */
lexerRuleRef(rule,label,args,elementIndex,scope) ::= <%
<if(label)>
int <label>Start<elementIndex> = CharIndex;<\n>
int <label>StartLine<elementIndex> = Line;<\n>
int <label>StartCharPos<elementIndex> = CharPositionInLine;<\n>
<if(scope)><scope.delegateName()>.<endif>m<rule.name>(<args; separator=", ">);
<checkRuleBacktrackFailure()><\n>
<label> = new <labelType>(input, TokenType.Invalid, TokenChannels.Default, <label>Start<elementIndex>,
CharIndex-1);<\n>

```

```

<label>.Line = <label>StartLine<elementIndex>;<\n>
<label>.CharPositionInLine = <label>StartCharPos<elementIndex>;
<else>
<if(scope)><scope.delegateName().<endif>m<rule.name>(<args; separator=", ">);
<checkRuleBacktrackFailure()>
<endif>
%>

/** i+=INT in lexer */
lexerRuleRefAndListLabel(rule,label,args,elementIndex,scope) ::= <<
<lexerRuleRef(...)>
<listLabelElem(elem=label,elemType=labelType,...)>
>>

/** EOF in the lexer */
lexerMatchEOF(label,elementIndex) ::= <%
<if(label)>
int <label>Start<elementIndex> = CharIndex;<\n>
int <label>StartLine<elementIndex> = Line;<\n>
int <label>StartCharPos<elementIndex> = CharPositionInLine;<\n>
Match(EOF); <checkRuleBacktrackFailure()><\n>
<labelType> <label> = new <labelType>(input, EOF, TokenChannels.Default, <label>Start<elementIndex>,
CharIndex-1);<\n>
<label>.Line = <label>StartLine<elementIndex>;<\n>
<label>.CharPositionInLine = <label>StartCharPos<elementIndex>;
<else>
Match(EOF); <checkRuleBacktrackFailure()>
<endif>
%>

// used for left-recursive rules
recRuleDefArg()          ::= "int <recRuleArg()>"
recRuleArg()             ::= "_p"
recRuleAltPredicate(ruleName,opPrec) ::= "<recRuleArg()> \<= <opPrec>"
recRuleSetResultAction() ::= "root_0=$<ruleName>_primary.tree;"
recRuleSetReturnAction(src,name) ::= "$<name>=$<src>.<name>;"

/** match ^(root children) in tree parser */
tree(root, actionsAfterRoot, children, nullableChildList,
enclosingTreeLevel, treeLevel) ::= <<
<root:element()>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if (input.LA(1) == TokenTypes.Down)
{
Match(input, TokenTypes.Down, null); <checkRuleBacktrackFailure()>
<children:element()>
Match(input, TokenTypes.Up, null); <checkRuleBacktrackFailure()>

```

```

}
<else>
Match(input, TokenType.Down, null); <checkRuleBacktrackFailure()>
<children:element()>
Match(input, TokenType.Up, null); <checkRuleBacktrackFailure()>
<endif>
>>

/** Every predicate is used as a validating predicate (even when it is
 * also hoisted into a prediction expression).
 */
validateSemanticPredicate(pred,description) ::= <<
if (!(<evalPredicate(...)>))
{
<ruleBacktrackFailure()>
throw new FailedPredicateException(input, "<ruleName>", "<description>");
}
>>

// F i x e d D F A (if-then-else)

dfaState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
int LA<decisionNumber>_<k> = input.LA(<k>);<n>
<edges; separator="\nelse ">
<if(!isTrue.(last(edges).labelExpr) && (!last(edges).predicates))>
else
{
<if(eotPredictsAlt)>
alt<decisionNumber> = <eotPredictsAlt>;
<else>
<ruleBacktrackFailure()>
NoViableAltException nvae = new NoViableAltException("<description>", <decisionNumber>, <stateNumber>,
input, <k>);
DebugRecognitionException(nvae);
<@noViableAltException()>
throw nvae;
<endif>
}
<endif>
>>

/** Same as a normal DFA state except that we don't examine lookahead
 * for the bypass alternative. It delays error detection but this
 * is faster, smaller, and more what people expect. For (X)? people
 * expect "if ( LA(1)==X ) match(X);" and that's it.
 */
dfaOptionalBlockState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
int LA<decisionNumber>_<k> = input.LA(<k>);<n>

```

```

<edges; separator="\nelse ">
>>

/** A DFA state that is actually the loopback decision of a closure
 * loop. If end-of-token (EOT) predicts any of the targets then it
 * should act like a default clause (i.e., no error can be generated).
 * This is used only in the lexer so that for ('a')* on the end of a rule
 * anything other than 'a' predicts exiting.
 */
dfaLoopbackState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
int LA<decisionNumber>_<k> = input.LA(<k>);<\n>
<edges; separator="\nelse "><\n>
<if(eotPredictsAlt)>
<if(!edges)>
alt<decisionNumber> = <eotPredictsAlt>;<! if no edges, don't gen ELSE !>
<else>
else
{
alt<decisionNumber> = <eotPredictsAlt>;
}<\n>
<endif>
<endif>
>>

/** An accept state indicates a unique alternative has been predicted */
dfaAcceptState(alt) ::= "alt<decisionNumber> = <alt>;"

/** A simple edge with an expression. If the expression is satisfied,
 * enter to the target state. To handle gated productions, we may
 * have to evaluate some predicates for this edge.
 */
dfaEdge(labelExpr, targetState, predicates) ::= <<
if ((<labelExpr>)<if(predicates)> && (<predicates>)<endif>)
{
<targetState>
}
>>

// F i x e d D F A (switch case)

/** A DFA state where a SWITCH may be generated. The code generator
 * decides if this is possible: CodeGenerator.canGenerateSwitch().
 */
dfaStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch (input.LA(<k>))
{
<edges; separator="\n">
default:

```

```

<if(eotPredictsAlt)>
  alt<decisionNumber>=<eotPredictsAlt>;
  break;<\n>
<else>
  {
    <ruleBacktrackFailure()>
    NoViableAltException nvae = new NoViableAltException("<description>", <decisionNumber>, <stateNumber>,
input, <k>);
    DebugRecognitionException(nvae);
    <@noViableAltException()>
    throw nvae;
  }
<endif>
}<\n>
>>

```

```

dfaOptionalBlockStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch (input.LA(<k>))
{
<edges; separator="\n">
}<\n>
>>

```

```

dfaLoopbackStateSwitch(k, edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch (input.LA(<k>))
{
<edges; separator="\n">
<if(eotPredictsAlt)>
default:
  alt<decisionNumber>=<eotPredictsAlt>;
  break;<\n>
<endif>
}<\n>
>>

```

```

dfaEdgeSwitch(labels, targetState) ::= <<
<labels:{it|case <it>;}; separator="\n">
{
  <targetState>
}
break;
>>

```

```
// C y c l i c D F A
```

```
/** The code to initiate execution of a cyclic DFA; this is used
```

```
* in the rule to predict an alt just like the fixed DFA case.
```

```
* The <name> attribute is inherited via the parser, lexer, ...
```

```

*/
dfaDecision(decisionNumber,description) ::= <<
try
{
alt<decisionNumber> = dfa<decisionNumber>.Predict(input);
}
catch (NoViableAltException nvae)
{
DebugRecognitionException(nvae);
throw;
}
>>

/* Dump DFA tables as run-length-encoded Strings of octal values.
* Can't use hex as compiler translates them before compilation.
* These strings are split into multiple, concatenated strings.
* Java puts them back together at compile time thankfully.
* Java cannot handle large static arrays, so we're stuck with this
* encode/decode approach. See analysis and runtime DFA for
* the encoding methods.
*/
cyclicDFA(dfa) ::= <<
private class DFA<dfa.decisionNumber> : DFA
{
private const string DFA<dfa.decisionNumber>_eotS =
"<dfa.javaCompressedEOT; wrap=\""+\n\t\t\">";
private const string DFA<dfa.decisionNumber>_eofS =
"<dfa.javaCompressedEOF; wrap=\""+\n\t\t\">";
private const string DFA<dfa.decisionNumber>_minS =
"<dfa.javaCompressedMin; wrap=\""+\n\t\t\">";
private const string DFA<dfa.decisionNumber>_maxS =
"<dfa.javaCompressedMax; wrap=\""+\n\t\t\">";
private const string DFA<dfa.decisionNumber>_acceptS =
"<dfa.javaCompressedAccept; wrap=\""+\n\t\t\">";
private const string DFA<dfa.decisionNumber>_specialS =
"<dfa.javaCompressedSpecial; wrap=\""+\n\t\t\">>";
private static readonly string[] DFA<dfa.decisionNumber>_transitionS =
{
<dfa.javaCompressedTransition:{s|\"<s; wrap=\""+\n\t\t\">\"}; separator=\", \n\">
};

private static readonly short[] DFA<dfa.decisionNumber>_eot =
DFA.UnpackEncodedString(DFA<dfa.decisionNumber>_eotS);
private static readonly short[] DFA<dfa.decisionNumber>_eof =
DFA.UnpackEncodedString(DFA<dfa.decisionNumber>_eofS);
private static readonly char[] DFA<dfa.decisionNumber>_min =
DFA.UnpackEncodedStringToUnsignedChars(DFA<dfa.decisionNumber>_minS);
private static readonly char[] DFA<dfa.decisionNumber>_max =

```

```

DFA.UnpackEncodedStringToUnsignedChars(DFA<dfa.decisionNumber>_maxS);
private static readonly short[] DFA<dfa.decisionNumber>_accept =
DFA.UnpackEncodedString(DFA<dfa.decisionNumber>_acceptS);
private static readonly short[] DFA<dfa.decisionNumber>_special =
DFA.UnpackEncodedString(DFA<dfa.decisionNumber>_specialS);
private static readonly short[][] DFA<dfa.decisionNumber>_transition;

static DFA<dfa.decisionNumber>()
{
int numStates = DFA<dfa.decisionNumber>_transitionS.Length;
DFA<dfa.decisionNumber>_transition = new short[numStates][];
for ( int i=0; i \< numStates; i++ )
{
DFA<dfa.decisionNumber>_transition[i] =
DFA.UnpackEncodedString(DFA<dfa.decisionNumber>_transitionS[i]);
}
}

public DFA<dfa.decisionNumber>( BaseRecognizer recognizer<if(dfa.specialStateSTs)>,
SpecialStateTransitionHandler specialStateTransition<endif> )
<if(dfa.specialStateSTs)>
: base(specialStateTransition)
<endif>
{
this.recognizer = recognizer;
this.decisionNumber = <dfa.decisionNumber>;
this.eot = DFA<dfa.decisionNumber>_eot;
this.eof = DFA<dfa.decisionNumber>_eof;
this.min = DFA<dfa.decisionNumber>_min;
this.max = DFA<dfa.decisionNumber>_max;
this.accept = DFA<dfa.decisionNumber>_accept;
this.special = DFA<dfa.decisionNumber>_special;
this.transition = DFA<dfa.decisionNumber>_transition;
}

public override string Description { get { return "<dfa.description>"; } }

public override void Error(NoViableAltException nvae)
{
DebugRecognitionException(nvae);
}
}<\n>
<if(dfa.specialStateSTs)>
private int SpecialStateTransition<dfa.decisionNumber>(DFA dfa, int s, IIntStream _input)<! throws
NoViableAltException!>
{
<if(LEXER)>
IIntStream input = _input;

```

```

<endif>
<if(PARSER)>
ITokenStream input = (ITokenStream)_input;
<endif>
<if(TREE_PARSER)>
ITreeNodeStream input = (ITreeNodeStream)_input;
<endif>
int _s = s;
s = -1;
<! pull these outside the switch cases to save space on locals !>
int LA<dfa.decisionNumber>_1 = input.LA(1);
int index<dfa.decisionNumber>_1 = input.Index;
switch (_s)
{
<dfa.specialStateSTs:{state |case <i0>:<! compressed special state numbers 0..n-1 !>
<state>}; separator="\n">

default:
break;
}

if (s >= 0)
return s;

<if(backtracking)>
if (state.backtracking > 0) { state.failed=true; return -1;}
<endif>
NoViableAltException nvae = new NoViableAltException(dfa.Description, <dfa.decisionNumber>, _s, input);
dfa.Error(nvae);
throw nvae;
}
<endif>
>>

/** A state in a cyclic DFA; it's a special state and part of a big switch on
 * state.
 */
cyclicDFASState(decisionNumber,stateNumber,edges,needErrorClause,semPredState) ::= <<
{
<if(semPredState)>
<! get next lookahead symbol to test edges, then rewind !>
input.Rewind();
<endif>
<edges; separator="\nelse ">
<if(semPredState)>
<! return input cursor to state before we rewound !>
input.Seek(index<decisionNumber>_1);
<endif>

```

```

break;
}
>>

/** Just like a fixed DFA edge, test the lookahead and indicate what
 * state to jump to next if successful.
 */
cyclicDFAEdge(labelExpr, targetStateNumber, edgeNumber, predicates) ::= <<
if ((<labelExpr><if(predicates)> && (<predicates><endif>) {s = <targetStateNumber>;}<\n>
>>

/** An edge pointing at end-of-token; essentially matches any char;
 * always jump to the target.
 */
eotDFAEdge(targetStateNumber,edgeNumber, predicates) ::= <<
s = <targetStateNumber>;<\n>
>>

// D F A E X P R E S S I O N S

andPredicates(left,right) ::= "<left>&&<right>"

orPredicates(operands) ::= "<operands; separator='||'>"

notPredicate(pred) ::= "!(<evalPredicate(...)>)"

evalPredicate(pred,description) ::= "<pred>"

evalSynPredicate(pred,description) ::= "EvaluatePredicate(<pred>_fragment)"

lookaheadTest(atom,k,atomAsInt) ::= "LA<decisionNumber>_<k>==<atom>"

/** Sometimes a lookahead test cannot assume that LA(k) is in a temp variable
 * somewhere. Must ask for the lookahead directly.
 */
isolatedLookaheadTest(atom,k,atomAsInt) ::= "input.LA(<k>)==<atom>"

lookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= <%
(LA<decisionNumber>_<k><ge()><lower> && LA<decisionNumber>_<k><le()><upper>)
%>

isolatedLookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::=
"(input.LA(<k>)<ge()><lower> && input.LA(<k>)<le()><upper>)"

le() ::= "\<="
ge() ::= ">="

```

```

setTest(ranges) ::= <<
<ranges; separator="||">
>>

// A T T R I B U T E S

attributeScope(scope) ::= <<
<if(scope)>
<if(scope.attributes)>
protected sealed partial class <scope.name>_scope
{
<scope.attributes: {it|public <it.decl>;}; separator="\n">
}
<if(scope.actions.scopeinit)>
protected void <scope.name>_scopeInit( <scope.name>_scope scope )
{
<scope.actions.scopeinit>
}
<else>
protected virtual void <scope.name>_scopeInit( <scope.name>_scope scope ) {}
<endif>
<if(scope.actions.scopeafter)>
protected void <scope.name>_scopeAfter( <scope.name>_scope scope )
{
<scope.actions.scopeafter>
}
<else>
protected virtual void <scope.name>_scopeAfter( <scope.name>_scope scope ) {}
<endif>
protected readonly ListStack\<<scope.name>_scope> <scope.name>_stack = new
ListStack\<<scope.name>_scope>();
<endif>
<endif>
>>

globalAttributeScope(scope) ::= <<
<attributeScope(...)>
>>

ruleAttributeScope(scope) ::= <<
<attributeScope(...)>
>>

returnStructName(it) ::= "<it.name>_return"

returnType(ruleDescriptor) ::= <%
<if(ruleDescriptor.returnScope.attributes && ruleDescriptor.hasMultipleReturnValues)>
<ruleDescriptor.grammar.recognizerName>.<ruleDescriptor:returnStructName()>

```

```

<elseif(ruleDescriptor.hasMultipleReturnValues)>
  <ruleReturnBaseType()>
<elseif(ruleDescriptor.hasSingleReturnValue)>
  <ruleDescriptor.singleValueReturnType>
<else>
  void
<endif>
%>

/** Generate the C# type associated with a single or multiple return
 * values.
 */
ruleLabelType(referencedRule) ::= <%
<if(referencedRule.hasMultipleReturnValues)>
  <ruleReturnBaseType()>
<elseif(referencedRule.hasSingleReturnValue)>
  <referencedRule.singleValueReturnType>
<else>
  void
<endif>
%>

delegateName(it) ::= <<
<if(it.label)><it.label><else>g<it.name><endif>
>>

/** Using a type to init value map, try to init a type; if not in table
 * must be an object, default value is "null".
 */
initValue(typeName) ::= <<
default(<typeName>)
>>

/** Define a rule label including default value */
ruleLabelDef(label) ::= <%
<ruleLabelType(referencedRule=label.referencedRule)> <label.label.text> =
<initValue(typeName=ruleLabelType(referencedRule=label.referencedRule))>;
%>

/** Define a return struct for a rule if the code needs to access its
 * start/stop tokens, tree stuff, attributes, ... Leave a hole for
 * subgroups to stick in members.
 */
returnScope(scope) ::= <<
<if(scope.attributes && ruleDescriptor.hasMultipleReturnValues)>
<returnScopeModifier(grammar=grammar,ruleDescriptor=ruleDescriptor)> sealed partial class
<ruleDescriptor:returnStructName()> : <ruleReturnBaseType()><@ruleReturnInterfaces()>
{

```

```

<scope.attributes:{it|public <it.decl>;}; separator="\n">
<@ruleReturnMembers()>
}
<endif>
>>

ruleReturnBaseType() ::= <%
<if(TREE_PARSER)>Tree<else>Parser<endif>RuleReturnScope\<<labelType>>
%>

@returnScope.ruleReturnMembers() ::= <<
>>

parameterScope(scope) ::= <<
<scope.attributes:{it|<it.decl>;}; separator=" ">
>>

parameterAttributeRef(attr) ::= <<
<attr.name; format="id">
>>

parameterSetAttributeRef(attr,expr) ::= <<
<attr.name; format="id"> =<expr>;
>>

scopeAttributeRef(scope,attr,index,negIndex) ::= <%
<if(negIndex)>
<scope>_stack[<scope>_stack.Count - <negIndex> - 1].<attr.name; format="id">
<else>
<if(index)>
<scope>_stack[<index>].<attr.name; format="id">
<else>
<scope>_stack.Peek().<attr.name; format="id">
<endif>
<endif>
%>

scopeSetAttributeRef(scope,attr,expr,index,negIndex) ::= <%
<if(negIndex)>
<scope>_stack[<scope>_stack.Count - <negIndex> - 1].<attr.name; format="id"> = <expr>;
<else>
<if(index)>
<scope>_stack[<index>].<attr.name; format="id"> = <expr>;
<else>
<scope>_stack.Peek().<attr.name; format="id"> = <expr>;
<endif>
<endif>
%>

```

```

/** $x is either global scope or x is rule with dynamic scope; refers
 * to stack itself not top of stack. This is useful for predicates
 * like {$function.Count>0 && $function::name.Equals("foo")}?
 */
isolatedDynamicScopeRef(scope) ::= "<scope>_stack"

/** reference an attribute of rule; might only have single return value */
ruleLabelRef(referencedRule,scope,attr) ::= <%
<if(referencedRule.hasMultipleReturnValues)>
(<scope>!=null?(<returnType(referencedRule)><scope>).<attr.name; format="id">:<initValue(attr.type)>)
<else>
<scope>
<endif>
%>

returnAttributeRef(ruleDescriptor,attr) ::= <%
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name; format="id">
<else>
<attr.name; format="id">
<endif>
%>

returnSetAttributeRef(ruleDescriptor,attr,expr) ::= <%
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name; format="id"> =<expr>;
<else>
<attr.name; format="id"> =<expr>;
<endif>
%>

/** How to translate $tokenLabel */
tokenLabelRef(label) ::= "<label>"

/** ids+=ID {$ids} or e+=expr {$e} */
listLabelRef(label) ::= "list_<label>"

// not sure the next are the right approach

tokenLabelPropertyRef_text(scope,attr) ::= "(<scope>!=null?<scope>.Text:default(string))"
tokenLabelPropertyRef_type(scope,attr) ::= "(<scope>!=null?<scope>.Type:0)"
tokenLabelPropertyRef_line(scope,attr) ::= "(<scope>!=null?<scope>.Line:0)"
tokenLabelPropertyRef_pos(scope,attr) ::= "(<scope>!=null?<scope>.CharPositionInLine:0)"
tokenLabelPropertyRef_channel(scope,attr) ::= "(<scope>!=null?<scope>.Channel:0)"
tokenLabelPropertyRef_index(scope,attr) ::= "(<scope>!=null?<scope>.TokenIndex:0)"
tokenLabelPropertyRef_tree(scope,attr) ::= "<scope>_tree"

```

```

tokenLabelPropertyRef_int(scope,attr) ::= "<scope>!=null?int.Parse(<scope>.Text):0)"

ruleLabelPropertyRef_start(scope,attr) ::= "<scope>!=null?((<labelType><scope>.Start):default(<labelType>))"
ruleLabelPropertyRef_stop(scope,attr) ::= "<scope>!=null?((<labelType><scope>.Stop):default(<labelType>))"
ruleLabelPropertyRef_tree(scope,attr) ::=
"<scope>!=null?((<ASTLabelType><scope>.Tree):default(<ASTLabelType>))"
ruleLabelPropertyRef_text(scope,attr) ::= <%
<if(TREE_PARSER)>
(<scope>!=null?(input.TokenStream.ToString(
input.TreeAdaptor.GetTokenStartIndex(<scope>.Start),
input.TreeAdaptor.GetTokenStopIndex(<scope>.Start))):default(string))
<else>
(<scope>!=null?input.ToString(<scope>.Start,<scope>.Stop):default(string))
<endif>
%>

ruleLabelPropertyRef_st(scope,attr) ::= "<scope>!=null?<scope>.Template:null)"

/** Isolated $RULE ref ok in lexer as it's a Token */
lexerRuleLabel(label) ::= "<label>"

lexerRuleLabelPropertyRef_type(scope,attr) ::=
"<scope>!=null?<scope>.Type:0)"

lexerRuleLabelPropertyRef_line(scope,attr) ::=
"<scope>!=null?<scope>.Line:0)"

lexerRuleLabelPropertyRef_pos(scope,attr) ::=
"<scope>!=null?<scope>.CharPositionInLine:-1)"

lexerRuleLabelPropertyRef_channel(scope,attr) ::=
"<scope>!=null?<scope>.Channel:0)"

lexerRuleLabelPropertyRef_index(scope,attr) ::=
"<scope>!=null?<scope>.TokenIndex:0)"

lexerRuleLabelPropertyRef_text(scope,attr) ::=
"<scope>!=null?<scope>.Text:default(string)"

lexerRuleLabelPropertyRef_int(scope,attr) ::=
"<scope>!=null?int.Parse(<scope>.Text):0)"

// Somebody may ref $template or $tree or $stop within a rule:
rulePropertyRef_start(scope,attr) ::= "retval.Start"
rulePropertyRef_stop(scope,attr) ::= "retval.Stop"
rulePropertyRef_tree(scope,attr) ::= "retval.Tree"
rulePropertyRef_text(scope,attr) ::= <%
<if(TREE_PARSER)>

```

```

input.TokenStream.ToString(
    input.TreeAdaptor.GetTokenStartIndex(retval.Start),
    input.TreeAdaptor.GetTokenStopIndex(retval.Start))
<else>
input.ToString(retval.Start,input.LT(-1))
<endif>
%>
rulePropertyRef_st(scope,attr) ::= "retval.Template"

lexerRulePropertyRef_text(scope,attr) ::= "Text"
lexerRulePropertyRef_type(scope,attr) ::= "_type"
lexerRulePropertyRef_line(scope,attr) ::= "state.tokenStartLine"
lexerRulePropertyRef_pos(scope,attr) ::= "state.tokenStartCharPositionInLine"
lexerRulePropertyRef_index(scope,attr) ::= "-1" // undefined token index in lexer
lexerRulePropertyRef_channel(scope,attr) ::= "_channel"
lexerRulePropertyRef_start(scope,attr) ::= "state.tokenStartCharIndex"
lexerRulePropertyRef_stop(scope,attr) ::= "(CharIndex-1)"
lexerRulePropertyRef_int(scope,attr) ::= "int.Parse(<scope>.Text)"

// setting $st and $tree is allowed in local rule. everything else
// is flagged as error
ruleSetPropertyRef_tree(scope,attr,expr) ::= "retval.Tree = <expr>";
ruleSetPropertyRef_st(scope,attr,expr) ::= "retval.Template =<expr>";

/** How to execute an action (only when not backtracking) */
execAction(action) ::= <%
<if(backtracking)>
if (<actions.(actionScope).synpredgate><\n>
{<\n>
<@indentedAction()><\n>
}
<else>
<action>
<endif>
%>

@execAction.indentedAction() ::= <<
<action>
>>

/** How to always execute an action even when backtracking */
execForcedAction(action) ::= "<action>"

// M I S C (properties, etc...)

bitset(name, words64) ::= <<
public static readonly BitSet <name> = new BitSet(new ulong[] {<words64: {it|<it>UL};separator=",">});
>>

```

```
codeFileExtension() ::= ".cs"
```

```
true_value() ::= "true"
```

```
false_value() ::= "false"
```

```
isTrue ::= [  
  "true" : true,  
  default : false  
]
```

Found in path(s):

```
* /opt/cola/permits/1274703855_1645234694.1/0/antlr-3-5-2-sources-  
jar/org/antlr/codegen/templates/CSharp2/CSharp2.stg
```

1.196 apache-log4j-api 2.17.1

1.196.1 Available under license :

Apache Log4j API

Copyright 1999-1969 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the

Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside

or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer,

and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.197 junit 4.13.1

1.197.1 Available under license :

JUnit

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

- a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
- b) in the case of each subsequent Contributor:
 - i) changes to the Program, and
 - ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents " mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is

licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

- a) it complies with the terms and conditions of this Agreement; and
- b) its license agreement:
 - i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
 - ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
 - iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
 - iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

- a) it must be made available under this Agreement; and
- b) a copy of this Agreement must be included with each copy of the

Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using

and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of

the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

1.198 feign-httpclient 8.18.0

1.198.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2015 Netflix, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1274702072_1645234420.76/0/feign-httpclient-8-18-0-sources-
jar/feign/httpclient/HttpClient.java
```

1.199 micronaut-liquibase 5.4.1

1.199.1 Available under license :

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of

the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works

that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A

PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.200 fabric8-:::kubernetes-model-:::-certificates 4.13.3

1.200.1 Available under license :

No license file was found, but licenses were detected in source scan.

*

* Copyright (C) 2015 Red Hat, Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*

Found in path(s):

* /opt/cola/permits/1288519892_1647861699.31/0/kubernetes-model-certificates-4-13-3-jar/manifest.vm

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0

Bnd-LastModified: 1619068558702

Build-Jdk-Spec: 1.8

Bundle-Description: Java client for Kubernetes and OpenShift

Bundle-DocURL: <http://redhat.com>

Bundle-License: <http://www.apache.org/licenses/LICENSE-2.0.txt>

Bundle-ManifestVersion: 2

Bundle-Name: Fabric8 :: Kubernetes Model :: Certificates

Bundle-SymbolicName: io.fabric8.kubernetes-model-certificates

Bundle-Vendor: Red Hat

Bundle-Version: 4.13.3

Created-By: Apache Maven Bundle Plugin

Export-Package: io.fabric8.kubernetes.api.model.certificates;uses:="com.fasterxml.jackson.annotation,com.fasterxml.jackson.databind,com.fas

terxml,jackson.databind.annotation,io.fabric8.kubernetes.api.builder,
io.fabric8.kubernetes.api.model,io.fabric8.kubernetes.model.annotatio
n";version="4.13.3"

Implementation-Title: Fabric8 :: Kubernetes Model :: Certificates

Implementation-Vendor: Red Hat

Implementation-Version: 4.13.3

Import-Package: com.fasterxml.jackson.annotation;version="[2.11,3)",co
m.fasterxml.jackson.databind;version="[2.11,3)",com.fasterxml.jackson
.databind.annotation;version="[2.11,3)",io.fabric8.kubernetes.api.bui
lder;version="[4.13,5)",io.fabric8.kubernetes.api.model,io.fabric8.ku
bernetes.model.annotation;version="[4.13,5)"

Require-Capability: osgi.ee;filter="(&(osgi.ee=JavaSE)(version=1.8))"

Specification-Title: Fabric8 :: Kubernetes Model :: Certificates

Specification-Vendor: Red Hat

Specification-Version: 4.13

Tool: Bnd-5.1.1.202006162103

Found in path(s):

* /opt/cola/permits/1288519892_1647861699.31/0/kubernetes-model-certificates-4-13-3-jar/META-
INF/MANIFEST.MF

No license file was found, but licenses were detected in source scan.

Copyright (C) 2015 Red Hat, Inc.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE>

2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Found in path(s):

* /opt/cola/permits/1288519892_1647861699.31/0/kubernetes-model-certificates-4-13-3-jar/META-
INF/maven/io.fabric8/kubernetes-model-certificates/pom.xml

1.201 javatm-ee-6-specification-apis 8.0

1.201.1 Available under license :

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.1

1. Definitions.

- 1.1. "Contributor" means each individual or entity that creates or contributes to the creation of Modifications.
- 1.2. "Contributor Version" means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.
- 1.3. "Covered Software" means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.
- 1.4. "Executable" means the Covered Software in any form other than Source Code.
- 1.5. "Initial Developer" means the individual or entity that first makes Original Software available under this License.
- 1.6. "Larger Work" means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.
- 1.7. "License" means this document.
- 1.8. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.
- 1.9. "Modifications" means the Source Code and Executable form of any of the following:
- A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;
 - B. Any new file that contains any part of the Original Software or previous Modification; or
 - C. Any new file that is contributed or otherwise made available under the terms of this License.
- 1.10. "Original Software" means the Source Code and Executable form of computer software code that is originally released under this License.
- 1.11. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. "Source Code" means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject

to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works.

You may create a Larger Work by combining Covered Software with

other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Oracle is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY

OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as "Participant") alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. If You assert a patent infringement claim against Participant alleging that the Participant Software directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

6.4. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE

INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and "commercial computer software documentation" as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control

laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor
Boston, MA 02110-1335
USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price.

Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this

License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source

code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the

integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH

YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1335 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type
`show w'. This is free software, and you are welcome to redistribute
it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the
appropriate parts of the General Public License. Of course, the commands
you use may be called something other than `show w' and `show c'; they
could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the
program `Gnomovision' (which makes passes at compilers) written by
James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program
into proprietary programs. If your program is a subroutine library, you
may consider it more useful to permit linking proprietary applications
with the library. If this is what you want to do, use the GNU Library
General Public License instead of this License.

#

Certain source files distributed by Oracle America, Inc. and/or its
affiliates are subject to the following clarification and special
exception to the GPLv2, based on the GNU Project exception for its
Classpath libraries, known as the GNU Classpath Exception, but only
where Oracle has expressly included in the particular source file's
header the words "Oracle designates this particular file as subject to
the "Classpath" exception as provided by Oracle in the LICENSE file
that accompanied this code."

You should also note that Oracle includes multiple, independent
programs in this software package. Some of those programs are provided
under licenses deemed incompatible with the GPLv2 by the Free Software
Foundation and others. For example, the package includes programs
licensed under the Apache License, Version 2.0. Such programs are
licensed to you under their original licenses.

Oracle facilitates your further distribution of this package by adding
the Classpath Exception to the necessary parts of its GPLv2 code, which
permits you to use that code in combination with other independent

modules not licensed under the GPLv2. However, note that this would not permit you to commingle code under an incompatible license with Oracle's GPLv2 licensed code by, for example, cutting and pasting such code into a file also containing Oracle's GPLv2 licensed code and then distributing the result. Additionally, if you were to remove the Classpath Exception from any of the files to which it applies and distribute the result, you would likely be required to license some or all of the other code in that distribution under the GPLv2 as well, and since the GPLv2 is incompatible with the license terms of some items included in the distribution by Oracle, removing the Classpath Exception could therefore effectively compromise your ability to further distribute the package.

Proceed with caution and we recommend that you obtain the advice of a lawyer skilled in open source matters before removing the Classpath Exception or making modifications to this package which may subsequently be redistributed and/or involve the use of third party software.

CLASSPATH EXCEPTION

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

1.202 opentelemetry 1.9.1

1.202.1 Available under license :

Note that publicsuffixes.gz is compiled from The Public Suffix List:
https://publicsuffix.org/list/public_suffix_list.dat

It is subject to the terms of the Mozilla Public License, v. 2.0:
<https://mozilla.org/MPL/2.0/>

1.203 kotlin 1.6.21

1.203.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
{ "version":3,"file":"kotlin.js","sources":["wrapper.js","js/arrayUtils.js","js/callableReferenceUtils.js","js/conversions.js","js/core.js","js/long.js","js/markerFunctions.js","js/misc.js","js/polyfills.js","js/rtti.js","runtime/arrayUtils.kt","runtime/Enum.kt","primitiveCompanionObjects.kt","common/src/generated/_Arrays.kt","common/src/generated/_Ranges.kt","unsigned/src/kotlin/UByte.kt","unsigned/src/kotlin/UInt.kt","unsigned/src/kotlin/UShort.kt","builtin-sources/Ranges.kt","src/kotlin/collections/Collections.kt","src/kotlin/collections/Maps.kt","src/kotlin/collections/Sets.kt","src/kotlin/text/StringNumberConversions.kt","src/kotlin/time/Duration.kt","unsigned/src/kotlin/UnsignedUtils.kt","src/kotlin/collections/Iterables.kt","src/kotlin/collections/Sequences.kt","src/kotlin/util/Preconditions.kt","js/src/generated/_ArraysJs.kt","src/kotlin/comparisons/Comparisons.kt","src/kotlin/util/Standard.kt","js/src/generated/_ComparisonsJs.kt","unsigned/src/kotlin/ULong.kt","common/src/generated/_Collections.kt","js/src/kotlin/collections.kt","src/kotlin/collections/Iterators.kt","common/src/generated/_Comparisons.kt","common/src/generated/_Maps.kt","common/src/generated/_OneToManyTitlecaseMappings.kt","js/src/kotlin/text/char.kt","js/src/kotlin/text/string.kt","src/kotlin/text/Char.kt","src/kotlin/CharCode.kt","common/src/generated/_Sequences.kt","common/src/generated/_Sets.kt","common/src/generated/_Strings.kt","src/kotlin/text/Strings.kt","unsigned/src/kotlin/UByteArray.kt","unsigned/src/kotlin/UIntArray.kt","unsigned/src/kotlin/ULongArray.kt","unsigned/src/kotlin/UShortArray.kt","common/src/generated/_UArrays.kt","common/src/generated/_UCollections.kt","common/src/generated/_UComparisons.kt","common/src/generated/_URanges.kt","common/src/generated/_USequences.kt","common/src/kotlin/ExceptionsH.kt","common/src/kotlin/JsAnnotationsH.kt","common/src/kotlin/ioH.kt","builtin-sources/Collections.kt","builtin-sources/Iterators.kt","builtin-sources/ProgressionIterators.kt","builtin-sources/Progressions.kt","builtin-sources/Range.kt","builtin-sources/Unit.kt","builtin-sources/annotation/Annotations.kt","builtin-sources/internal/InternalAnnotations.kt","builtin-sources/internal/progressionUtil.kt","src/kotlin/builtins.kt","src/kotlin/jsTypeOf.kt","src/kotlin/kotlin.kt","src/kotlin/CharCode_js-v1.kt","src/kotlin/coroutines/CoroutineImpl.kt","src/kotlin/util/Result.kt","src/kotlin/coroutines/Continuation.kt","src/kotlin/coroutines/intrinsics/IntrinsicsJs.kt","src/kotlin/currentBeMisc.kt","src/kotlin/exceptions.kt","src/kotlin/jsOperators.kt","src/kotlin/math_js-v1.kt","src/kotlin/numbers_js-v1.kt","src/kotlin/reflection_js-v1.kt","src/kotlin/text/numberConversions_js-v1.kt","js/src/generated/_CharCategories.kt","js/src/generated/_CollectionsJs.kt","js/src/generated/_DigitChars.kt","js/src/generated/_LetterChars.kt","js/src/generated/_OtherLowercaseChars.kt","js/src/generated/_OtherUppercaseChars.kt","js/src/generated/_StringsJs.kt","js/src/generated/_TitlecaseMappings.kt","js/src/generated/_UArraysJs.kt","js/src/generated/_WhitespaceChars.kt","js/src/kotlin/Comparator.kt","js/src/kotlin/annotations.kt","js/src/kotlin/annotationsJVM.kt","js/src/kotlin/collections/AbstractMutableCollection.kt","js/src/kotlin/collections/AbstractMutableList.kt","js/src/kotlin/collections/AbstractMutableMap.kt","js/src/kotlin/collections/AbstractMutableSet.kt","js/src/kotlin/collections/ArrayList.kt","js/src/kotlin/collections/ArraySorting.kt","js/src/kotlin/collections/ArraysJs.kt","js/src/kotlin/collections/EqualityComparator.kt","js/src/kotlin/collections/HashMap.kt","js/src/kotlin/collections/HashSet.kt","js/src/kotlin/collections/InternalHashCodeMap.kt","js/src/kotlin/collections/InternalMap.kt","js/src/kotlin/collections/InternalStringMap.kt","js/src/kotlin/collections/LinkedHashMap.kt","js/src/kotlin/collections/LinkedHashSet.kt","js/src/kotlin/concurrent.kt","js/src/kotlin/console.kt","js/src/kotlin/coroutines/SafeContinuationJs.kt","js/src/kotlin/coroutines/cancellation/CancellationException.kt","js/src/kotlin/coroutines/js/internal/EmptyContinuation.kt","js/src/kotlin/date.kt","js/src/kotlin/dom/Builders.kt","js/src/kotlin/dom/Classes.kt","js/src/kotlin/dom/Dom.kt","js/src/kotlin/dom/EventListener.kt","js/src/kotlin/dom/ItemArrayLike.kt","js/src/kotlin/dom/Mutations.kt","js/src/kotlin/dynamic.kt","js/src/kotlin/exceptionUtils.kt","js/src/kotlin/grouping.kt","src/kotlin/collections/Grouping.kt","js/src/kotlin/internalAnnotations.kt","js/src/kotlin/json.kt","js/src/kotlin/math.kt","js/src/kotlin/numbers.kt","js/src/kotlin/promise.
```

kt", "js/src/kotlin/random/PlatformRandom.kt", "js/src/kotlin/reflect/AssociatedObjects.kt", "js/src/kotlin/reflect/JsClass.kt", "js/src/kotlin/reflect/KClassImpl.kt", "js/src/kotlin/reflect/KClassesImpl.kt", "js/src/kotlin/reflect/KTypeHelpers.kt", "js/src/kotlin/reflect/KTypeImpl.kt", "js/src/kotlin/reflect/KTypeParameterImpl.kt", "js/src/kotlin/reflect/primitives.kt", "js/src/kotlin/reflect/reflection.kt", "js/src/kotlin/regexp.kt", "js/src/kotlin/sequence.kt", "js/src/kotlin/text/CharCategoryJS.kt", "js/src/kotlin/text/CharacterCodingExceptionJs.kt", "js/src/kotlin/text/StringBuilderJs.kt", "js/src/kotlin/text/numberConversions.kt", "js/src/kotlin/text/regex.kt", "src/kotlin/text/StringBuilder.kt", "js/src/kotlin/text/stringsCode.kt", "js/src/kotlin/text/utf8Encoding.kt", "js/src/kotlin/throwableExtensions.kt", "js/src/kotlin/time/DurationJs.kt", "js/src/kotlin/time/DurationUnit.kt", "js/src/kotlin/time/MonoTimeSource.kt", "js/src/kotlinx/dom/Builders.kt", "js/src/kotlinx/dom/Classes.kt", "src/kotlin/text/regex/RegexExtensions.kt", "js/src/kotlinx/dom/Dom.kt", "js/src/kotlinx/dom/Mutations.kt", "js/src/org.w3c/deprecated.kt", "js/src/org.w3c/org.khronos.webgl.kt", "js/src/org.w3c/org.w3c.dom.clipboard.kt", "js/src/org.w3c/org.w3c.dom.css.kt", "js/src/org.w3c/org.w3c.dom.encryptedmedia.kt", "js/src/org.w3c/org.w3c.dom.events.kt", "js/src/org.w3c/org.w3c.dom.kt", "js/src/org.w3c/org.w3c.fetch.kt", "js/src/org.w3c/org.w3c.dom.mediacapture.kt", "js/src/org.w3c/org.w3c.dom.mediasource.kt", "js/src/org.w3c/org.w3c.dom.pointerevents.kt", "js/src/org.w3c/org.w3c.dom.svg.kt", "js/src/org.w3c/org.w3c.files.kt", "js/src/org.w3c/org.w3c.notifications.kt", "js/src/org.w3c/org.w3c.workers.kt", "js/src/org.w3c/org.w3c.xhr.kt", "src/kotlin/annotations/Experimental.kt", "src/kotlin/annotations/ExperimentalStdlibApi.kt", "src/kotlin/annotations/Inference.kt", "src/kotlin/annotations/Multiplatform.kt", "src/kotlin/annotations/OptIn.kt", "src/kotlin/collections/AbstractCollection.kt", "src/kotlin/collections/AbstractIterator.kt", "src/kotlin/collections/AbstractList.kt", "src/kotlin/collections/AbstractMap.kt", "src/kotlin/collections/AbstractSet.kt", "src/kotlin/collections/ArrayDeque.kt", "src/kotlin/collections/Arrays.kt", "src/kotlin/collections/BrittleContainsOptimization.kt", "src/kotlin/collections/IndexedValue.kt", "src/kotlin/collections/MapAccessors.kt", "src/kotlin/collections/MapWithDefault.kt", "src/kotlin/collections/MutableCollections.kt", "src/kotlin/collections/ReversedViews.kt", "src/kotlin/collections/SequenceBuilder.kt", "src/kotlin/collections/SlidingWindow.kt", "src/kotlin/collections/UArraySorting.kt", "src/kotlin/comparisons/compareTo.kt", "src/kotlin/contracts/ContractBuilder.kt", "src/kotlin/coroutines/ContinuationInterceptor.kt", "src/kotlin/coroutines/CoroutineContext.kt", "src/kotlin/coroutines/CoroutineContextImpl.kt", "src/kotlin/coroutines/intrinsics/Intrinsics.kt", "src/kotlin/experimental/bitwiseOperations.kt", "src/kotlin/experimental/inferenceMarker.kt", "src/kotlin/internal/Annotations.kt", "src/kotlin/properties/Delegates.kt", "src/kotlin/properties/Interfaces.kt", "src/kotlin/properties/ObservableProperty.kt", "src/kotlin/properties/PropertyReferenceDelegates.kt", "src/kotlin/random/Random.kt", "src/kotlin/random/URandom.kt", "src/kotlin/random/XorWowRandom.kt", "src/kotlin/ranges/Ranges.kt", "src/kotlin/reflect/KClasses.kt", "src/kotlin/reflect/KTypeProjection.kt", "src/kotlin/reflect/KVariance.kt", "src/kotlin/reflect/typeOf.kt", "src/kotlin/text/Appendable.kt", "src/kotlin/text/Indent.kt", "src/kotlin/text/Typography.kt", "src/kotlin/text/regex/MatchResult.kt", "src/kotlin/time/DurationUnit.kt", "src/kotlin/time/ExperimentalTime.kt", "src/kotlin/time/TimeSource.kt", "src/kotlin/time/TimeSources.kt", "src/kotlin/time/measureTime.kt", "src/kotlin/util/DeepRecursive.kt", "src/kotlin/util/FloorDivMod.kt", "src/kotlin/util/HashCode.kt", "src/kotlin/util/KotlinVersion.kt", "src/kotlin/util/Lateinit.kt", "src/kotlin/util/Lazy.kt", "src/kotlin/util/Numbers.kt", "src/kotlin/util/Suspend.kt", "src/kotlin/util/Tuples.kt", "unsigned/src/kotlin/UIntRange.kt", "unsigned/src/kotlin/UIterators.kt", "unsigned/src/kotlin/ULongRange.kt", "unsigned/src/kotlin/UMath.kt", "unsigned/src/kotlin/UNumbers.kt", "unsigned/src/kotlin/UProgressionUtil.kt", "unsigned/src/kotlin/UStrings.kt", "unsigned/src/kotlin/annotations/Unsigned.kt", "common/src/kotlin/MathH.kt"], "sourcesContent": ["(function (root, factory) {\n if (typeof define === 'function' && define.amd) {\n define('kotlin', ['exports'], factory);\n }\n else if (typeof exports === 'object') {\n factory(module.exports);\n }\n else {\n root.kotlin = {};\n factory(root.kotlin);\n }\n})(this, function (Kotlin) {\n var _ = Kotlin;\n\n insertContent();\n});\n", /*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors. \n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n *\n\nKotlin.isBooleanArray = function (a) {\n return (Array.isArray(a) || a instanceof Int8Array) && a.\$type\$ === \"BooleanArray\";\n}\n\nKotlin.isByteArray = function (a) {\n return a instanceof Int8Array && a.\$type\$!== \"BooleanArray\";\n}\n\nKotlin.isShortArray = function (a) {\n return a instanceof Int16Array;\n}\n\nKotlin.isCharArray = function (a) {\n return a instanceof Uint16Array && a.\$type\$ === \"CharArray\";\n}\n\nKotlin.isIntArray = function (a) {\n return a instanceof Int32Array;\n}\n\nKotlin.isFloatArray = function (a) {\n return a instanceof

```

Float32Array\n};\n\nKotlin.isDoubleArray = function (a) {\n  return a instanceof
Float64Array\n};\n\nKotlin.isLongArray = function (a) {\n  return Array.isArray(a) && a.$type$ ===
\"LongArray\"\n};\n\nKotlin.isArray = function (a) {\n  return Array.isArray(a) &&
!a.$type$;\n};\n\nKotlin.isArrayish = function (a) {\n  return Array.isArray(a) ||
ArrayBuffer.isView(a)\n};\n\nKotlin.arrayToString = function (a) {\n  if (a === null) return \"null\"\n  var
toString = Kotlin.isCharArray(a) ? String.fromCharCode : Kotlin.toString;\n  return \"[\" +
Array.prototype.map.call(a, function(e) { return toString(e); }).join(\", \") + \"]\";\n};\n\nKotlin.arrayDeepToString
= function (arr) {\n  return Kotlin.kotlin.collections.contentDeepToStringImpl(arr);\n};\n\nKotlin.arrayEquals =
function (a, b) {\n  if (a === b) {\n    return true;\n  }\n  if (a === null || b === null || !Kotlin.isArrayish(b) ||
a.length !== b.length) {\n    return false;\n  }\n  for (var i = 0, n = a.length; i < n; i++) {\n    if
(!Kotlin.equals(a[i], b[i])) {\n      return false;\n    }\n  }\n  return true;\n};\n\nKotlin.arrayDeepEquals =
function (a, b) {\n  return Kotlin.kotlin.collections.contentDeepEqualsImpl(a, b);\n};\n\nKotlin.arrayHashCode =
function (arr) {\n  if (arr === null) return 0\n  var result = 1;\n  for (var i = 0, n = arr.length; i < n; i++) {\n
result = ((31 * result | 0) + Kotlin.hashCode(arr[i])) | 0;\n  }\n  return result;\n};\n\nKotlin.arrayDeepHashCode =
function (arr) {\n  return
Kotlin.kotlin.collections.contentDeepHashCodeImpl(arr);\n};\n\nKotlin.primitiveArraySort = function (array) {\n
array.sort(Kotlin.doubleCompareTo)\n};\n\n\"/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming
Language contributors. \n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\nKotlin.getCallableRef = function(name, f) {\n  f.callableName = name;\n  return
f;\n};\n\nKotlin.getPropertyCallableRef = function(name, paramCount, getter, setter) {\n  getter.get = getter;\n
getter.set = setter;\n  getter.callableName = name;\n  return getPropertyRefClass(getter, setter,
propertyRefClassMetadataCache[paramCount]);\n};\n\nfunction getPropertyRefClass(obj, setter, cache) {\n
obj.$metadata$ = getPropertyRefMetadata(typeof setter === \"function\" ? cache.mutable : cache.immutable);\n
obj.constructor = obj;\n  return obj;\n}\n\nvar propertyRefClassMetadataCache = [\n  {\n    mutable: { value:
null, implementedInterface: function () {\n      return Kotlin.kotlin.reflect.KMutableProperty0 }\n    },\n
immutable: { value: null, implementedInterface: function () {\n      return Kotlin.kotlin.reflect.KProperty0 }\n
}\n  },\n  {\n    mutable: { value: null, implementedInterface: function () {\n      return
Kotlin.kotlin.reflect.KMutableProperty1 }\n    },\n    immutable: { value: null, implementedInterface: function
() {\n      return Kotlin.kotlin.reflect.KProperty1 }\n    }\n  }\n];\n\nfunction getPropertyRefMetadata(cache)
{\n  if (cache.value === null) {\n    cache.value = {\n      interfaces: [cache.implementedInterface()],\n
baseClass: null,\n      functions: {},\n      properties: {},\n      types: {},\n      staticMembers: {}\n
};\n  }\n  return cache.value;\n}\n\n\"/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming
Language contributors. \n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\nKotlin.toShort = function (a) {\n  return (a & 0xFFFF) << 16 >>
16;\n};\n\nKotlin.toByte = function (a) {\n  return (a & 0xFF) << 24 >> 24;\n};\n\nKotlin.toChar = function (a) {\n
return a & 0xFFFF;\n};\n\nKotlin.numberToLong = function (a) {\n  return a instanceof Kotlin.Long ? a :
Kotlin.Long.fromNumber(a);\n};\n\nKotlin.numberToInt = function (a) {\n  return a instanceof Kotlin.Long ?
a.toInt() : Kotlin.doubleToInt(a);\n};\n\nKotlin.numberToShort = function (a) {\n  return
Kotlin.toShort(Kotlin.numberToInt(a));\n};\n\nKotlin.numberToByte = function (a) {\n  return
Kotlin.toByte(Kotlin.numberToInt(a));\n};\n\nKotlin.numberToDouble = function (a) {\n  return
+a;\n};\n\nKotlin.numberToChar = function (a) {\n  return
Kotlin.toChar(Kotlin.numberToInt(a));\n};\n\nKotlin.doubleToInt = function(a) {\n  if (a > 2147483647) return
2147483647;\n  if (a < -2147483648) return -2147483648;\n  return a | 0;\n};\n\nKotlin.toBoxedChar = function
(a) {\n  if (a == null) return a;\n  if (a instanceof Kotlin.BoxedChar) return a;\n  return new
Kotlin.BoxedChar(a);\n};\n\nKotlin.unboxChar = function(a) {\n  if (a == null) return a;\n  return
Kotlin.toChar(a);\n};\n\n\"/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
contributors. \n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\nKotlin.equals = function (obj1, obj2) {\n  if (obj1 == null) {\n    return obj2 ==

```

```

null;\n } \n\n if (obj2 == null) {\n     return false;\n } \n\n if (obj1 !== obj2) {\n     return obj2 !== obj2;\n } \n\n if (typeof obj1 === "object" && typeof obj1.equals === "function") {\n     return obj1.equals(obj2);\n } \n\n if (typeof obj1 === "number" && typeof obj2 === "number") {\n     return obj1 === obj2 && (obj1 !== 0 || 1 / obj1 === 1 / obj2)\n } \n\n return obj1 === obj2;\n};\n\nKotlin.hashCode = function (obj) {\n if (obj == null) {\n     return 0;\n } \n var objType = typeof obj;\n if ("object" === objType) {\n     return "function" === typeof obj.hashCode ? obj.hashCode() : getObjectHashCode(obj);\n } \n if ("function" === objType) {\n     return getObjectHashCode(obj);\n } \n if ("number" === objType) {\n     return Kotlin.numberHashCode(obj);\n } \n if ("boolean" === objType) {\n     return Number(obj)\n } \n\n var str = String(obj);\n return getStringHashCode(str);\n};\n\nKotlin.toString = function (o) {\n if (o == null) {\n     return "null";\n } \n else if (Kotlin.isArrayish(o)) {\n     return "[...]";\n } \n else {\n     return o.toString();\n } \n};\n\n/** @const */\nvar POW_2_32 = 4294967296;\n// TODO: consider switching to Symbol type once we are on ES6.\n/** @const */\nvar OBJECT_HASH_CODE_PROPERTY_NAME = "kotlinHashCodeValue";\n\nfunction getObjectHashCode(obj) {\n if (!(OBJECT_HASH_CODE_PROPERTY_NAME in obj)) {\n     var hash = (Math.random() * POW_2_32) | 0;\n     // Make 32-bit signed integer.\n     Object.defineProperty(obj, OBJECT_HASH_CODE_PROPERTY_NAME, {\n         value: hash, enumerable: false });\n } \n return obj[OBJECT_HASH_CODE_PROPERTY_NAME];\n}\n\nfunction getStringHashCode(str) {\n var hash = 0;\n for (var i = 0; i < str.length; i++) {\n     var code = str.charCodeAt(i);\n     hash = (hash * 31 + code) | 0;\n // Keep it 32-bit.\n } \n return hash;\n}\n\nKotlin.identityHashCode = getObjectHashCode;\n", /*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors. \n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n *\n * Copyright 2009 The Closure Library Authors. All Rights Reserved.\n *\n * Licensed under the Apache License, Version 2.0 (the "License");\n *\n * you may not use this file except in compliance with the License.\n *\n * You may obtain a copy of the License at\n *\n * http://www.apache.org/licenses/LICENSE-2.0\n *\n * Unless required by applicable law or agreed to in writing, software\n *\n * distributed under the License is distributed on an "AS-IS" BASIS,\n *\n * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.\n *\n * Constructs a 64-bit two's-complement integer, given its low and high 32-bit\n *\n * values as *signed* integers. See the from* functions below for more\n *\n * convenient ways of constructing Longs.\n *\n * The internal representation of a long is the two given signed, 32-bit values.\n *\n * We use 32-bit pieces because these are the size of integers on which\n *\n * Javascript performs bit-operations. For operations like addition and\n *\n * multiplication, we split each number into 16-bit pieces, which can easily be\n *\n * multiplied within Javascript's floating-point representation without overflow\n *\n * or change in sign.\n *\n * In the algorithms below, we frequently reduce the negative case to the\n *\n * positive case by negating the input(s) and then post-processing the result.\n *\n * Note that we must ALWAYS check specially whether those values are MIN_VALUE\n *\n * (-2^63) because -MIN_VALUE == MIN_VALUE (since 2^63 cannot be represented as\n *\n * a positive number, it overflows back into a negative). Not handling this\n *\n * case would often result in infinite recursion.\n *\n * @param {number} low The low (signed) 32 bits of the long.\n *\n * @param {number} high The high (signed) 32 bits of the long.\n *\n * @constructor\n *\n * @final\n *\n * Kotlin.Long = function(low, high) {\n /**\n * @type {number}\n * @private\n */\n this.low_ = low | 0; // force into 32 signed bits.\n\n /**\n * @type {number}\n * @private\n */\n this.high_ = high | 0; // force into 32 signed bits.\n};\n\nKotlin.Long.$metadata$ = {\n kind: "class",\n simpleName: "Long",\n interfaces: []\n};\n\n// NOTE: Common constant values ZERO, ONE, NEG_ONE, etc. are defined below the\n // from* methods on which they depend.\n\n/**\n * A cache of the Long representations of small integer values.\n *\n * @type {!Object}\n *\n * @private\n */\nKotlin.Long.IntCache_ = {};\n\n/**\n * Returns a Long representing the given (32-bit) integer value.\n *\n * @param {number} value The 32-bit integer in question.\n *\n * @return {!Kotlin.Long} The corresponding Long value.\n */\nKotlin.Long.fromInt = function(value) {\n if (-128 <= value && value < 128) {\n     var cachedObj = Kotlin.Long.IntCache_[value];\n     if (cachedObj) {\n         return cachedObj;\n     } \n } \n\n var obj = new Kotlin.Long(value | 0, value < 0 ? -1 : 0);\n if (-128 <= value && value < 128) {\n     Kotlin.Long.IntCache_[value] = obj;\n } \n return obj;\n};\n\n/**\n * Converts this number value to `Long`.\n *\n * The fractional part, if any, is rounded down towards zero.\n *\n * Returns

```

```

zero if this `Double` value is `NaN`, `Long.MIN_VALUE` if it's less than `Long.MIN_VALUE`, `n *
`Long.MAX_VALUE` if it's bigger than `Long.MAX_VALUE`. `n * @param {number} value The number in
question. `n * @return {!Kotlin.Long} The corresponding Long value. `n */
Kotlin.Long.fromNumber =
function(value) { `n if (isNaN(value)) { `n return Kotlin.Long.ZERO; `n } else if (value <= -
Kotlin.Long.TWO_PWR_63_DBL_) { `n return Kotlin.Long.MIN_VALUE; `n } else if (value + 1 >=
Kotlin.Long.TWO_PWR_63_DBL_) { `n return Kotlin.Long.MAX_VALUE; `n } else if (value < 0) { `n return
Kotlin.Long.fromNumber(-value).negate(); `n } else { `n return new Kotlin.Long(`n (value %
Kotlin.Long.TWO_PWR_32_DBL_) | 0, `n (value / Kotlin.Long.TWO_PWR_32_DBL_) | 0); `n
}}; `n `n /** `n * Returns a Long representing the 64-bit integer that comes by concatenating `n * the given high and
low bits. Each is assumed to use 32 bits. `n * @param {number} lowBits The low 32-bits. `n * @param {number}
highBits The high 32-bits. `n * @return {!Kotlin.Long} The corresponding Long value. `n */
Kotlin.Long.fromBits =
function(lowBits, highBits) { `n return new Kotlin.Long(lowBits, highBits); `n }; `n `n /** `n * Returns a Long
representation of the given string, written using the given `n * radix. `n * @param {string} str The textual
representation of the Long. `n * @param {number=} opt_radix The radix in which the text is written. `n * @return
{!Kotlin.Long} The corresponding Long value. `n */
Kotlin.Long.fromString =
function(str, opt_radix) { `n if
(str.length == 0) { `n throw Error('number format error: empty string'); `n } `n `n var radix = opt_radix || 10; `n if
(radix < 2 || 36 < radix) { `n throw Error('radix out of range: ' + radix); `n } `n `n if (str.charAt(0) == '-') { `n return
Kotlin.Long.fromString(str.substring(1), radix).negate(); `n } else if (str.indexOf('-') >= 0) { `n throw Error('number
format error: interior "-" character: ' + str); `n } `n `n // Do several (8) digits each time through the loop, so as to `n //
minimize the calls to the very expensive emulated div. `n var radixToPower =
Kotlin.Long.fromNumber(Math.pow(radix, 8)); `n `n var result = Kotlin.Long.ZERO; `n for (var i = 0; i < str.length;
i += 8) { `n var size = Math.min(8, str.length - i); `n var value = parseInt(str.substring(i, i + size), radix); `n if
(size < 8) { `n var power = Kotlin.Long.fromNumber(Math.pow(radix, size)); `n result =
result.multiply(power).add(Kotlin.Long.fromNumber(value)); `n } else { `n result =
result.multiply(radixToPower); `n result = result.add(Kotlin.Long.fromNumber(value)); `n } `n } `n return
result; `n }; `n `n // NOTE: the compiler should inline these constant values below and then remove `n // these
variables, so there should be no runtime penalty for these. `n `n /** `n * Number used repeated below in calculations.
This must appear before the `n * first call to any from* function below. `n * @type {number} `n * @private `n
*/
Kotlin.Long.TWO_PWR_16_DBL_ = 1 << 16; `n `n /** `n * @type {number} `n * @private `n
*/
Kotlin.Long.TWO_PWR_24_DBL_ = 1 << 24; `n `n /** `n * @type {number} `n * @private `n
*/
Kotlin.Long.TWO_PWR_32_DBL_ = `n Kotlin.Long.TWO_PWR_16_DBL_ *
Kotlin.Long.TWO_PWR_16_DBL_; `n `n /** `n * @type {number} `n * @private `n
*/
Kotlin.Long.TWO_PWR_31_DBL_ = `n Kotlin.Long.TWO_PWR_32_DBL_ / 2; `n `n /** `n * @type
{number} `n * @private `n */
Kotlin.Long.TWO_PWR_48_DBL_ = `n Kotlin.Long.TWO_PWR_32_DBL_ *
Kotlin.Long.TWO_PWR_16_DBL_; `n `n /** `n * @type {number} `n * @private `n
*/
Kotlin.Long.TWO_PWR_64_DBL_ = `n Kotlin.Long.TWO_PWR_32_DBL_ *
Kotlin.Long.TWO_PWR_32_DBL_; `n `n /** `n * @type {number} `n * @private `n
*/
Kotlin.Long.TWO_PWR_63_DBL_ = `n Kotlin.Long.TWO_PWR_64_DBL_ / 2; `n `n /** `n * @type
{!Kotlin.Long} `n */
Kotlin.Long.ZERO = Kotlin.Long.fromInt(0); `n `n /** `n * @type {!Kotlin.Long} `n
*/
Kotlin.Long.ONE = Kotlin.Long.fromInt(1); `n `n /** `n * @type {!Kotlin.Long} `n */
Kotlin.Long.NEG_ONE =
Kotlin.Long.fromInt(-1); `n `n /** `n * @type {!Kotlin.Long} `n */
Kotlin.Long.MAX_VALUE = `n
Kotlin.Long.fromBits(0xFFFFFFFF | 0, 0x7FFFFFFF | 0); `n `n /** `n * @type {!Kotlin.Long} `n
*/
Kotlin.Long.MIN_VALUE = Kotlin.Long.fromBits(0, 0x80000000 | 0); `n `n /** `n * @type {!Kotlin.Long} `n *
@private `n */
Kotlin.Long.TWO_PWR_24_ = Kotlin.Long.fromInt(1 << 24); `n `n /** `n * @return {number} The
value, assuming it is a 32-bit integer. `n */
Kotlin.Long.prototype.toInt =
function() { `n return this.low_; `n }; `n `n /** `n * @return {number} The closest floating-point representation to this value. `n */
Kotlin.Long.prototype.toNumber =
function() { `n return this.high_ * Kotlin.Long.TWO_PWR_32_DBL_ + `n
this.getLowBitsUnsigned(); `n }; `n `n /** `n * @return {number} The 32-bit hashCode of this value.

```

```

*\nKotlin.Long.prototype.hashCode = function() {\n return this.high_ ^ this.low_;\n};\n\n/**\n * @param
 {number=} opt_radix The radix in which the text should be written.\n * @return {string} The textual representation
 of this value.\n * @override\n */\nKotlin.Long.prototype.toString = function(opt_radix) {\n var radix = opt_radix ||
 10;\n if (radix < 2 || 36 < radix) {\n throw Error('radix out of range: ' + radix);\n }\n\n if (this.isZero()) {\n
return '0';\n }\n\n if (this.isNegative()) {\n if (this.equalsLong(Kotlin.Long.MIN_VALUE)) {\n // We need to
change the Long value before it can be negated, so we remove\n // the bottom-most digit in this base and then
recurse to do the rest.\n var radixLong = Kotlin.Long.fromNumber(radix);\n var div = this.div(radixLong);\n
var rem = div.multiply(radixLong).subtract(this);\n return div.toString(radix) + rem.toInt().toString(radix);\n }
else {\n return '-' + this.negate().toString(radix);\n }\n }\n\n // Do several (6) digits each time through the loop,
so as to\n // minimize the calls to the very expensive emulated div.\n var radixToPower =
Kotlin.Long.fromNumber(Math.pow(radix, 6));\n\n var rem = this;\n var result = '';\n while (true) {\n var
remDiv = rem.div(radixToPower);\n var intVal = rem.subtract(remDiv.multiply(radixToPower)).toInt();\n var
digits = intVal.toString(radix);\n rem = remDiv;\n if (rem.isZero()) {\n return digits + result;\n } else {\n
while (digits.length < 6) {\n digits = '0' + digits;\n }\n result = " + digits + result;\n }\n }\n};\n\n\n/**
 * @return {number} The high 32-bits as a signed value. */\nKotlin.Long.prototype.getHighBits = function() {\n
return this.high_;\n};\n\n\n/** * @return {number} The low 32-bits as a signed value.
*\nKotlin.Long.prototype.getLowBits = function() {\n return this.low_;\n};\n\n\n/** * @return {number} The low
32-bits as an unsigned value. */\nKotlin.Long.prototype.getLowBitsUnsigned = function() {\n return (this.low_ >=
0) ?\n this.low_ : Kotlin.Long.TWO_PWR_32_DBL_ + this.low_;\n};\n\n\n/**\n * @return {number} Returns
the number of bits needed to represent the absolute\n * value of this Long.\n
*\nKotlin.Long.prototype.getNumBitsAbs = function() {\n if (this.isNegative()) {\n if
(this.equalsLong(Kotlin.Long.MIN_VALUE)) {\n return 64;\n } else {\n return
this.negate().getNumBitsAbs();\n }\n } else {\n var val = this.high_ != 0 ? this.high_ : this.low_;\n for (var bit
= 31; bit > 0; bit--) {\n if ((val & (1 << bit)) != 0) {\n break;\n }\n }\n return this.high_ != 0 ? bit + 33
: bit + 1;\n }\n};\n\n\n/** * @return {boolean} Whether this value is zero. */\nKotlin.Long.prototype.isZero =
function() {\n return this.high_ == 0 && this.low_ == 0;\n};\n\n\n/** * @return {boolean} Whether this value is
negative. */\nKotlin.Long.prototype.isNegative = function() {\n return this.high_ < 0;\n};\n\n\n/** * @return
{boolean} Whether this value is odd. */\nKotlin.Long.prototype.isOdd = function() {\n return (this.low_ & 1) ==
1;\n};\n\n\n/**\n * @param {Kotlin.Long} other Long to compare against.\n * @return {boolean} Whether this
Long equals the other.\n */\nKotlin.Long.prototype.equalsLong = function(other) {\n return (this.high_ ==
other.high_) && (this.low_ == other.low_);\n};\n\n\n/**\n * @param {Kotlin.Long} other Long to compare
against.\n * @return {boolean} Whether this Long does not equal the other.\n
*\nKotlin.Long.prototype.notEqualsLong = function(other) {\n return (this.high_ != other.high_) || (this.low_ !=
other.low_);\n};\n\n\n/**\n * @param {Kotlin.Long} other Long to compare against.\n * @return {boolean}
Whether this Long is less than the other.\n */\nKotlin.Long.prototype.lessThan = function(other) {\n return
this.compare(other) < 0;\n};\n\n\n/**\n * @param {Kotlin.Long} other Long to compare against.\n * @return
{boolean} Whether this Long is less than or equal to the other.\n */\nKotlin.Long.prototype.lessThanOrEqual =
function(other) {\n return this.compare(other) <= 0;\n};\n\n\n/**\n * @param {Kotlin.Long} other Long to
compare against.\n * @return {boolean} Whether this Long is greater than the other.\n
*\nKotlin.Long.prototype.greaterThan = function(other) {\n return this.compare(other) > 0;\n};\n\n\n\n/**\n *
@param {Kotlin.Long} other Long to compare against.\n * @return {boolean} Whether this Long is greater than or
equal to the other.\n */\nKotlin.Long.prototype.greaterThanOrEqual = function(other) {\n return
this.compare(other) >= 0;\n};\n\n\n\n/**\n * @param {Kotlin.Long} other Long to compare against.\n * @return
{number} 0 if they are the same, 1 if the this is greater, and -1\n * if the given one is greater.\n */\nKotlin.Long.prototype.compare = function(other) {\n if (this.equalsLong(other)) {\n
return 0;\n }\n\n var thisNeg = this.isNegative();\n var otherNeg = other.isNegative();\n if (thisNeg &&
!otherNeg) {\n return -1;\n }\n if (!thisNeg && otherNeg) {\n return 1;\n }\n\n // at this point, the signs are the
same, so subtraction will not overflow\n if (this.subtract(other).isNegative()) {\n return -1;\n } else {\n return

```

```

1;\n };\n\n\n/** @return {!Kotlin.Long} The negation of this value. *\nKotlin.Long.prototype.negate =
function() {\n if (this.equalsLong(Kotlin.Long.MIN_VALUE)) {\n return Kotlin.Long.MIN_VALUE;\n } else
{\n return this.not().add(Kotlin.Long.ONE);\n };\n\n\n/**\n * Returns the sum of this and the given Long.\n *
@param {Kotlin.Long} other Long to add to this one.\n * @return {!Kotlin.Long} The sum of this and the given
Long.\n *\nKotlin.Long.prototype.add = function(other) {\n // Divide each number into 4 chunks of 16 bits, and
then sum the chunks.\n\n var a48 = this.high_ >>> 16;\n var a32 = this.high_ & 0xFFFF;\n var a16 = this.low_
>>> 16;\n var a00 = this.low_ & 0xFFFF;\n\n var b48 = other.high_ >>> 16;\n var b32 = other.high_ & 0xFFFF;\n
var b16 = other.low_ >>> 16;\n var b00 = other.low_ & 0xFFFF;\n\n var c48 = 0, c32 = 0, c16 = 0, c00 = 0;\n c00
+= a00 + b00;\n c16 += c00 >>> 16;\n c00 &= 0xFFFF;\n c16 += a16 + b16;\n c32 += c16 >>> 16;\n c16 &=
0xFFFF;\n c32 += a32 + b32;\n c48 += c32 >>> 16;\n c32 &= 0xFFFF;\n c48 += a48 + b48;\n c48 &=
0xFFFF;\n return Kotlin.Long.fromBits((c16 << 16) | c00, (c48 << 16) | c32);\n};\n\n\n/**\n * Returns the
difference of this and the given Long.\n * @param {Kotlin.Long} other Long to subtract from this.\n * @return
{!Kotlin.Long} The difference of this and the given Long.\n *\nKotlin.Long.prototype.subtract = function(other)
{\n return this.add(other.negate());\n};\n\n\n/**\n * Returns the product of this and the given long.\n * @param
{Kotlin.Long} other Long to multiply with this.\n * @return {!Kotlin.Long} The product of this and the other.\n
*\nKotlin.Long.prototype.multiply = function(other) {\n if (this.isZero()) {\n return Kotlin.Long.ZERO;\n } else
if (other.isZero()) {\n return Kotlin.Long.ZERO;\n }\n\n if (this.equalsLong(Kotlin.Long.MIN_VALUE)) {\n
return other.isOdd() ? Kotlin.Long.MIN_VALUE : Kotlin.Long.ZERO;\n } else if
(other.equalsLong(Kotlin.Long.MIN_VALUE)) {\n return this.isOdd() ? Kotlin.Long.MIN_VALUE :
Kotlin.Long.ZERO;\n }\n\n if (this.isNegative()) {\n if (other.isNegative()) {\n return
this.negate().multiply(other.negate());\n } else {\n return this.negate().multiply(other).negate();\n }\n } else if
(other.isNegative()) {\n return this.multiply(other.negate()).negate();\n }\n\n // If both longs are small, use float
multiplication\n if (this.lessThan(Kotlin.Long.TWO_PWR_24_) &&\n
other.lessThan(Kotlin.Long.TWO_PWR_24_)) {\n return Kotlin.Long.fromNumber(this.toNumber() *
other.toNumber());\n }\n\n // Divide each long into 4 chunks of 16 bits, and then add up 4x4 products.\n // We can
skip products that would overflow.\n\n var a48 = this.high_ >>> 16;\n var a32 = this.high_ & 0xFFFF;\n var a16 =
this.low_ >>> 16;\n var a00 = this.low_ & 0xFFFF;\n\n var b48 = other.high_ >>> 16;\n var b32 = other.high_ &
0xFFFF;\n var b16 = other.low_ >>> 16;\n var b00 = other.low_ & 0xFFFF;\n\n var c48 = 0, c32 = 0, c16 = 0, c00
= 0;\n c00 += a00 * b00;\n c16 += c00 >>> 16;\n c00 &= 0xFFFF;\n c16 += a16 * b00;\n c32 += c16 >>> 16;\n
c16 &= 0xFFFF;\n c16 += a00 * b16;\n c32 += c16 >>> 16;\n c16 &= 0xFFFF;\n c32 += a32 * b00;\n c48 +=
c32 >>> 16;\n c32 &= 0xFFFF;\n c32 += a16 * b16;\n c48 += c32 >>> 16;\n c32 &= 0xFFFF;\n c32 += a00 *
b32;\n c48 += c32 >>> 16;\n c32 &= 0xFFFF;\n c48 += a48 * b00 + a32 * b16 + a16 * b32 + a00 * b48;\n c48
&= 0xFFFF;\n return Kotlin.Long.fromBits((c16 << 16) | c00, (c48 << 16) | c32);\n};\n\n\n\n/**\n * Returns this
Long divided by the given one.\n * @param {Kotlin.Long} other Long by which to divide.\n * @return
{!Kotlin.Long} This Long divided by the given one.\n *\nKotlin.Long.prototype.div = function(other) {\n if
(other.isZero()) {\n throw Error('division by zero');\n } else if (this.isZero()) {\n return Kotlin.Long.ZERO;\n
}\n\n if (this.equalsLong(Kotlin.Long.MIN_VALUE)) {\n if (other.equalsLong(Kotlin.Long.ONE) ||\n
other.equalsLong(Kotlin.Long.NEG_ONE)) {\n return Kotlin.Long.MIN_VALUE; // recall that -MIN_VALUE
== MIN_VALUE\n } else if (other.equalsLong(Kotlin.Long.MIN_VALUE)) {\n return Kotlin.Long.ONE;\n
}\n } else {\n // At this point, we have |other| >= 2, so |this/other| < |MIN_VALUE|.\n var halfThis =
this.shiftRight(1);\n var approx = halfThis.div(other).shiftLeft(1);\n if
(approx.equalsLong(Kotlin.Long.ZERO)) {\n return other.isNegative() ? Kotlin.Long.ONE :
Kotlin.Long.NEG_ONE;\n } else {\n var rem = this.subtract(other.multiply(approx));\n var result =
approx.add(rem.div(other));\n return result;\n }\n }\n } else if
(other.equalsLong(Kotlin.Long.MIN_VALUE)) {\n return Kotlin.Long.ZERO;\n }\n\n if (this.isNegative()) {\n
if (other.isNegative()) {\n return this.negate().div(other.negate());\n } else {\n return
this.negate().div(other).negate();\n }\n } else if (other.isNegative()) {\n return
this.div(other.negate()).negate();\n }\n\n // Repeat the following until the remainder is less than other: find a\n //

```

```

floating-point that approximates remainder / other *from below*, add this\n // into the result, and subtract it from
the remainder. It is critical that\n // the approximate value is less than or equal to the real value so that the\n //
remainder never becomes negative.\n var res = Kotlin.Long.ZERO;\n var rem = this;\n while
(rem.greaterThanOrEqual(other)) {\n // Approximate the result of division. This may be a little greater or\n //
smaller than the actual value.\n var approx = Math.max(1, Math.floor(rem.toNumber() / other.toNumber()));\n\n
// We will tweak the approximate result by changing it in the 48-th digit or\n // the smallest non-fractional digit,
whichever is larger.\n var log2 = Math.ceil(Math.log(approx) / Math.LN2);\n var delta = (log2 <= 48) ? 1 :
Math.pow(2, log2 - 48);\n // Decrease the approximation until it is smaller than the remainder. Note\n // that if
it is too large, the product overflows and is negative.\n var approxRes = Kotlin.Long.fromNumber(approx);\n
var approxRem = approxRes.multiply(other);\n while (approxRem.isNegative() || approxRem.greaterThan(rem))
{\n approx -= delta;\n approxRes = Kotlin.Long.fromNumber(approx);\n approxRem =
approxRes.multiply(other);\n }\n\n // We know the answer can't be zero... and actually, zero would cause\n //
infinite recursion since we would make no progress.\n if (approxRes.isZero()) {\n approxRes =
Kotlin.Long.ONE;\n }\n\n res = res.add(approxRes);\n rem = rem.subtract(approxRem);\n }\n return
res;\n};\n\n\n/**\n * Returns this Long modulo the given one.\n * @param {Kotlin.Long} other Long by which to
mod.\n * @return {!Kotlin.Long} This Long modulo the given one.\n */\nKotlin.Long.prototype.modulo =
function(other) {\n return this.subtract(this.div(other).multiply(other));\n};\n\n\n/**\n * @return {!Kotlin.Long} The
bitwise-NOT of this value. *\nKotlin.Long.prototype.not = function() {\n return Kotlin.Long.fromBits(~this.low_,
~this.high_);\n};\n\n\n/**\n * Returns the bitwise-AND of this Long and the given one.\n * @param {Kotlin.Long}
other The Long with which to AND.\n * @return {!Kotlin.Long} The bitwise-AND of this and the other.\n
*/\nKotlin.Long.prototype.and = function(other) {\n return Kotlin.Long.fromBits(this.low_ & other.low_,\n
this.high_ & other.high_);\n};\n\n\n/**\n * Returns the bitwise-OR of this Long and the given one.\n *
@param {Kotlin.Long} other The Long with which to OR.\n * @return {!Kotlin.Long} The bitwise-OR of this and
the other.\n */\nKotlin.Long.prototype.or = function(other) {\n return Kotlin.Long.fromBits(this.low_ |
other.low_,\n this.high_ | other.high_);\n};\n\n\n/**\n * Returns the bitwise-XOR of this Long
and the given one.\n * @param {Kotlin.Long} other The Long with which to XOR.\n * @return {!Kotlin.Long}
The bitwise-XOR of this and the other.\n */\nKotlin.Long.prototype.xor = function(other) {\n return
Kotlin.Long.fromBits(this.low_ ^ other.low_,\n this.high_ ^ other.high_);\n};\n\n\n/**\n *
Returns this Long with bits shifted to the left by the given amount.\n * @param {number} numBits The number of
bits by which to shift.\n * @return {!Kotlin.Long} This shifted to the left by the given amount.\n
*/\nKotlin.Long.prototype.shiftLeft = function(numBits) {\n numBits &= 63;\n if (numBits == 0) {\n return
this;\n } else {\n var low = this.low_;\n if (numBits < 32) {\n var high = this.high_;\n return
Kotlin.Long.fromBits(\n low << numBits,\n (high << numBits) | (low >>> (32 - numBits)));\n } else
{\n return Kotlin.Long.fromBits(0, low << (numBits - 32));\n }\n }\n};\n\n\n/**\n * Returns this Long with
bits shifted to the right by the given amount.\n * @param {number} numBits The number of bits by which to shift.\n
* @return {!Kotlin.Long} This shifted to the right by the given amount.\n */\nKotlin.Long.prototype.shiftRight =
function(numBits) {\n numBits &= 63;\n if (numBits == 0) {\n return this;\n } else {\n var high = this.high_;\n
if (numBits < 32) {\n var low = this.low_;\n return Kotlin.Long.fromBits(\n (low >>> numBits) | (high
<<< (32 - numBits)),\n high >> numBits);\n } else {\n return Kotlin.Long.fromBits(\n high >>
(numBits - 32),\n high >= 0 ? 0 : -1);\n }\n }\n};\n\n\n/**\n * Returns this Long with bits shifted to the right
by the given amount, with\n * zeros placed into the new leading bits.\n * @param {number} numBits The number
of bits by which to shift.\n * @return {!Kotlin.Long} This shifted to the right by the given amount, with\n *
zeros placed into the new leading bits.\n */\nKotlin.Long.prototype.shiftRightUnsigned = function(numBits) {\n
numBits &= 63;\n if (numBits == 0) {\n return this;\n } else {\n var high = this.high_;\n if (numBits < 32) {\n
var low = this.low_;\n return Kotlin.Long.fromBits(\n (low >>> numBits) | (high <<< (32 - numBits)),\n
high >>> numBits);\n } else if (numBits == 32) {\n return Kotlin.Long.fromBits(high, 0);\n } else {\n
return Kotlin.Long.fromBits(high >>> (numBits - 32), 0);\n }\n }\n};\n\n\n// Support for
Kotlin\nKotlin.Long.prototype.equals = function (other) {\n return other instanceof Kotlin.Long &&

```

```

this.equalsLong(other);\n};\n\nKotlin.Long.prototype.compareTo_11rb$ =
Kotlin.Long.prototype.compare;\n\nKotlin.Long.prototype.inc = function() {\n  return
this.add(Kotlin.Long.ONE);\n};\n\nKotlin.Long.prototype.dec = function() {\n  return
this.add(Kotlin.Long.NEG_ONE);\n};\n\nKotlin.Long.prototype.valueOf = function() {\n  return
this.toNumber();\n};\n\nKotlin.Long.prototype.unaryPlus = function() {\n  return
this;\n};\n\nKotlin.Long.prototype.unaryMinus = Kotlin.Long.prototype.negate;\n\nKotlin.Long.prototype.inv =
Kotlin.Long.prototype.not;\n\nKotlin.Long.prototype.rangeTo = function (other) {\n  return new
Kotlin.kotlin.ranges.LongRange(this, other);\n};\n\n/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin
Programming Language contributors. \n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n */\n\n * @param {string} id\n * @param {Object} declaration\n\n *\nKotlin.defineModule = function (id, declaration) {\n};\n\nKotlin.defineInlineFunction = function(tag, fun) {\n
return fun;\n};\n\nKotlin.wrapFunction = function(fun) {\n  var f = function() {\n    f = fun();\n    return
f.apply(this, arguments);\n  }; \n  return function() {\n    return f.apply(this, arguments);\n
};\n};\n\nKotlin.isTypeOf = function(type) {\n  return function (object) {\n    return typeof object === type;\n
}\n};\n\nKotlin.isInstanceOf = function (klass) {\n  return function (object) {\n    return Kotlin.isType(object,
klass);\n  }\n};\n\nKotlin.orNull = function (fn) {\n  return function (object) {\n    return object == null ||
fn(object);\n  }\n};\n\nKotlin.andPredicate = function (a, b) {\n  return function (object) {\n    return a(object)
&& b(object);\n  }\n};\n\nKotlin.kotlinModuleMetadata = function (abiVersion, moduleName, data)
{\n};\n\nKotlin.suspendCall = function(value) {\n  return value;\n};\n\nKotlin.coroutineResult = function(qualifier)
{\n  throwMarkerError();\n};\n\nKotlin.coroutineController = function(qualifier) {\n
throwMarkerError();\n};\n\nKotlin.coroutineReceiver = function(qualifier) {\n
throwMarkerError();\n};\n\nKotlin.setCoroutineResult = function(value, qualifier) {\n
throwMarkerError();\n};\n\nKotlin.getReifiedTypeParameterKType = function(typeParameter) {\n
throwMarkerError();\n};\n\nfunction throwMarkerError() {\n  throw new Error(\n    "This marker function
should never be called. \n" +\n    "Looks like compiler did not eliminate it properly. \n" +\n    "Please, report
an issue if you caught this exception.\n");\n}\n\nKotlin.getFunctionById = function(id, defaultValue) {\n  return
function() {\n    return defaultValue;\n  }\n};\n\n/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin
Programming Language contributors. \n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n */\n\nKotlin.compareTo = function (a, b) {\n  var typeA = typeof a;\n  if
(typeA === "number") {\n    if (typeof b === "number") {\n      return Kotlin.doubleCompareTo(a, b);\n
}\n  }\n  return Kotlin.primitiveCompareTo(a, b);\n}\n\n if (typeA === "string" || typeA === "boolean") {\n
return Kotlin.primitiveCompareTo(a, b);\n}\n  return
a.compareTo_11rb$(b);\n};\n\nKotlin.primitiveCompareTo = function (a, b) {\n  return a < b ? -1 : a > b ? 1 :
0;\n};\n\nKotlin.doubleCompareTo = function (a, b) {\n  if (a < b) return -1;\n  if (a > b) return 1;\n\n  if (a ===
b) {\n    if (a !== 0) return 0;\n\n    var ia = 1 / a;\n    return ia === 1 / b ? 0 : (ia < 0 ? -1 : 1);\n  }\n\n
return a !== a ? (b !== b ? 0 : 1) : -1;\n};\n\nKotlin.charInc = function (value) {\n  return
Kotlin.toChar(value+1);\n};\n\nKotlin.charDec = function (value) {\n  return Kotlin.toChar(value-
1);\n};\n\nKotlin.imul = Math.imul || imul;\n\nKotlin.imulEmulated = imul;\n\nfunction imul(a, b) {\n  return ((a &
0xffff0000) * (b & 0xffff) + (a & 0xffff) * (b | 0)) | 0;\n}\n\n(function() {\n  var buf = new ArrayBuffer(8);\n  var
bufFloat64 = new Float64Array(buf);\n  var bufFloat32 = new Float32Array(buf);\n  var bufInt32 = new
Int32Array(buf);\n  var lowIndex = 0;\n  var highIndex = 1;\n\n  bufFloat64[0] = -1; // bff00000_00000000\n  if
(bufInt32[lowIndex] !== 0) {\n    lowIndex = 1;\n    highIndex = 0;\n  }\n\n  Kotlin.doubleToBits =
function(value) {\n    return Kotlin.doubleToRawBits(isNaN(value) ? NaN : value);\n  };\n\n  Kotlin.doubleToRawBits = function(value) {\n    bufFloat64[0] = value;\n    return
Kotlin.Long.fromBits(bufInt32[lowIndex], bufInt32[highIndex]);\n  };\n\n  Kotlin.doubleFromBits =
function(value) {\n    bufInt32[lowIndex] = value.low_;\n    bufInt32[highIndex] = value.high_;\n    return
bufFloat64[0];\n  };\n\n  Kotlin.floatToBits = function(value) {\n    return Kotlin.floatToRawBits(isNaN(value)
? NaN : value);\n  };\n\n  Kotlin.floatToRawBits = function(value) {\n    bufFloat32[0] = value;\n    return

```



```

    }\n        else\n            {\n                return Math.log(x + Math.sqrt(x * x - 1));\n            }\n}\n        else\n            {\n                var y = Math.sqrt(x - 1);\n                // approximation by taylor series in y at 0 up to order 2\n                var result = y;\n                if (y >= taylor_2_bound)\n                    {\n                        var y3 = y * y * y;\n                        // approximation by taylor series in y at 0 up to order 4\n                        result -= y3 / 12;\n                    }\n                return Math.sqrt(2) * result;\n            }\n        };\n    }\n    if (typeof Math.atanh === \"undefined\")\n        {\n            Math.atanh = function(x) {\n                if (Math.abs(x) < taylor_n_bound) {\n                    var result = x;\n                }\n                if (Math.abs(x) > taylor_2_bound) {\n                    result += (x * x * x) / 3;\n                }\n                return result;\n            }\n            return Math.log((1 + x) / (1 - x)) / 2;\n        };\n    }\n    if (typeof Math.log1p === \"undefined\") {\n        Math.log1p = function(x) {\n            if (Math.abs(x) < taylor_n_bound) {\n                var x2 = x * x;\n                var x3 = x2 * x;\n                var x4 = x3 * x;\n                // approximation by taylor series in x at 0 up to order 4\n                return (-x4 / 4 + x3 / 3 - x2 / 2 + x);\n            }\n            return Math.log(x + 1);\n        };\n    }\n    if (typeof Math.expm1 === \"undefined\") {\n        Math.expm1 = function(x) {\n            if (Math.abs(x) < taylor_n_bound)\n                {\n                    var x2 = x * x;\n                    var x3 = x2 * x;\n                    var x4 = x3 * x;\n                    // approximation by taylor series in x at 0 up to order 4\n                    return (x4 / 24 + x3 / 6 + x2 / 2 + x);\n                }\n            return Math.exp(x) - 1;\n        };\n    }\n    }\n    }\n    if (typeof Math.hypot === \"undefined\") {\n        Math.hypot = function() {\n            var y = 0;\n            var length = arguments.length;\n            for (var i = 0; i < length; i++) {\n                if (arguments[i] === Infinity || arguments[i] === -Infinity) {\n                    return Infinity;\n                }\n                y += arguments[i] * arguments[i];\n            }\n            return Math.sqrt(y);\n        };\n    }\n    }\n    if (typeof Math.log10 === \"undefined\") {\n        Math.log10 = function(x) {\n            return Math.log(x) * Math.LOG10E;\n        };\n    }\n    }\n    if (typeof Math.log2 === \"undefined\") {\n        Math.log2 = function(x) {\n            return Math.log(x) * Math.LOG2E;\n        };\n    }\n    }\n    if (typeof Math.clz32 === \"undefined\") {\n        Math.clz32 = (function(log, LN2) {\n            return function(x) {\n                var asUint = x >>> 0;\n                if (asUint === 0) {\n                    return 32;\n                }\n                return 31 - (log(asUint) / LN2 | 0) | 0; // the \"| 0\" acts like math.floor\n            };\n        })(Math.log, Math.LN2);\n    }\n    }\n    }\n    if (typeof ArrayBuffer.isView === \"undefined\") {\n        ArrayBuffer.isView = function(a) {\n            return a != null && a.__proto__ != null && a.__proto__.__proto__ === Int8Array.prototype.__proto__;\n        };\n    }\n    }\n    }\n    if (typeof Array.prototype.fill === \"undefined\") {\n        // Polyfill from https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/fill#Polyfill\n        Object.defineProperty(Array.prototype, 'fill', {\n            value: function (value) {\n                // Steps 1-2.\n                if (this == null) {\n                    throw new TypeError('this is null or not defined');\n                }\n                var O = Object(this);\n                // Steps 3-5.\n                var len = O.length >>> 0;\n                // Steps 6-7.\n                var start = arguments[1];\n                var relativeStart = start >> 0;\n                // Step 8.\n                var k = relativeStart < 0 ?\n                    Math.max(len + relativeStart, 0) :\n                    Math.min(relativeStart, len);\n                // Steps 9-10.\n                var end = arguments[2];\n                var relativeEnd = end === undefined ?\n                    len : end >> 0;\n                // Step 11.\n                var finalValue = relativeEnd < 0 ?\n                    Math.max(len + relativeEnd, 0) :\n                    Math.min(relativeEnd, len);\n                // Step 12.\n                while (k < finalValue) {\n                    O[k] = value;\n                    k++;\n                }\n                // Step 13.\n                return O;\n            };\n        });\n    }\n    }\n    }\n    (function() {\n        function normalizeOffset(offset, length) {\n            if (offset < 0) return Math.max(0, offset + length);\n            return Math.min(offset, length);\n        }\n        function typedArraySlice(begin, end) {\n            if (typeof end === \"undefined\") {\n                end = this.length;\n            }\n            begin = normalizeOffset(begin || 0, this.length);\n            end = Math.max(begin, normalizeOffset(end, this.length));\n            return new this.constructor(this.subarray(begin, end));\n        }\n        var arrays = [Int8Array, Int16Array, Uint16Array, Int32Array, Float32Array, Float64Array];\n        for (var i = 0; i < arrays.length; ++i) {\n            var TypedArray = arrays[i];\n            if (typeof TypedArray.prototype.fill === \"undefined\") {\n                Object.defineProperty(TypedArray.prototype, 'fill', {\n                    value: Array.prototype.fill\n                });\n            }\n            if (typeof TypedArray.prototype.slice === \"undefined\") {\n                Object.defineProperty(TypedArray.prototype, 'slice', {\n                    value: typedArraySlice\n                });\n            }\n        }\n        // Patch apply to work with TypedArrays if needed.\n        try {\n            (function() {}).apply(null, new Int32Array(0))\n        } catch (e) {\n            var apply = Function.prototype.apply;\n            Object.defineProperty(Function.prototype, 'apply', {\n                value: function(self, array) {\n                    return apply.call(this, self, [].slice.call(array));\n                }\n            });\n        }\n        // Patch map to

```



```

Kotlin.kotlin.Comparable);
}

Kotlin.isCharSequence = function (value) {
    return typeof value === "string" ||
        Kotlin.isType(value, Kotlin.kotlin.CharSequence);
};

/*
 * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */
package kotlin

@PublishedApi
internal fun <T> Array(size: Int): Array<T> {
    @JsName("newArray")
    fun <T> newArray(size: Int, initialValue: T) = fillArrayVal(Array<T>(size), initialValue)
    @JsName("newArrayF")
    inline fun <T> arrayWithFun(size: Int, init: (Int) -> T) = fillArrayFun(Array<T>(size), init)
    @JsName("fillArray")
    inline fun <T> fillArrayFun(array: Array<T>, init: (Int) -> T): Array<T> {
        for (i in 0..array.size - 1) {
            array[i] = init(i)
        }
        return array
    }
    @JsName("booleanArray")
    fun booleanArray(size: Int, init: dynamic): Array<Boolean> {
        val result: dynamic = Array<Boolean>(size)
        result.`$type$` = "BooleanArray"
        return when (init) {
            null, true -> fillArrayVal(result, false)
            false -> result
            else -> fillArrayFun<Boolean>(result, init)
        }
    }
    @JsName("booleanArrayF")
    inline fun booleanArrayWithFun(size: Int, init: (Int) -> Boolean): Array<Boolean> = fillArrayFun(booleanArray(size, false), init)
    @JsName("charArray")
    @Suppress("UNUSED_PARAMETER")
    fun charArray(size: Int, init: dynamic): Array<Char> {
        val result = js("new Uint16Array(size)")
        result.`$type$` = "CharArray"
        return when (init) {
            null, true, false -> result // For consistency
            else -> fillArrayFun<Char>(result, init)
        }
    }
    @JsName("charArrayF")
    inline fun charArrayWithFun(size: Int, init: (Int) -> Char): Array<Char> {
        val array = charArray(size, null)
        for (i in 0..array.size - 1) {
            @Suppress("UNUSED_VARIABLE") // used in js block
            val value = init(i)
            js("array[i] = value;")
        }
        return array
    }
    @JsName("untypedCharArrayF")
    inline fun untypedCharArrayWithFun(size: Int, init: (Int) -> Char): Array<Char> {
        val array = Array<Char>(size)
        for (i in 0..array.size - 1) {
            @Suppress("UNUSED_VARIABLE") // used in js block
            val value = init(i)
            js("array[i] = value;")
        }
        return array
    }
    @JsName("longArray")
    fun longArray(size: Int, init: dynamic): Array<Long> {
        val result: dynamic = Array<Long>(size)
        result.`$type$` = "LongArray"
        return when (init) {
            null, true -> fillArrayVal(result, 0L)
            false -> result
            else -> fillArrayFun<Long>(result, init)
        }
    }
    @JsName("longArrayF")
    inline fun longArrayWithFun(size: Int, init: (Int) -> Long): Array<Long> = fillArrayFun(longArray(size, false), init)
    private fun <T> fillArrayVal(array: Array<T>, initialValue: T): Array<T> {
        for (i in 0..array.size - 1) {
            array[i] = initialValue
        }
        return array
    }
}

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */
package kotlin

public class Enum<T : Enum<T>> : Comparable<Enum<T>> {
    @JsName("name$") private var _name: String = ""
    @JsName("ordinal$") private var _ordinal: Int = 0
    val name: String
    get() = _name
    val ordinal: Int
    get() = _ordinal
    override fun compareTo(other: Enum<T>) = ordinal.compareTo(other.ordinal)
    override fun equals(other: Any?) = this === other
    override fun hashCode(): Int = js("Kotlin.identityHashCode")(this)
    override fun toString() = name
    companion object {
        /*
        Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
        Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
        */
        package kotlin.js.internal
        @JsName("DoubleCompanionObject")
        internal object DoubleCompanionObject {
            @JsName("MIN_VALUE")
            const val MIN_VALUE: Double = 4.9E-324
            @JsName("MAX_VALUE")
            const val MAX_VALUE: Double = 1.7976931348623157E308
            @JsName("POSITIVE_INFINITY")
            @Suppress("DIVISION_BY_ZERO")
            const val POSITIVE_INFINITY: Double = 1.0 / 0.0
            @JsName("NEGATIVE_INFINITY")
            @Suppress("DIVISION_BY_ZERO")
            const val NEGATIVE_INFINITY: Double = -1.0 / 0.0
            @JsName("NaN")
            @Suppress("DIVISION_BY_ZERO")
            const val NaN: Double = -(0.0 / 0.0)
            @JsName("SIZE_BYTES")
            const val SIZE_BYTES = 8
            @JsName("SIZE_BITS")
            const val SIZE_BITS = 64
        }
        @JsName("FloatCompanionObject")
        internal object FloatCompanionObject {
            @JsName("MIN_VALUE")
            const val MIN_VALUE: Float = 1.4E-45F
            @JsName("MAX_VALUE")
            const val MAX_VALUE: Float = 3.4E38F
        }
    }
}

```

```

const val MAX_VALUE: Float = 3.4028235E38\n @JsName("POSITIVE_INFINITY")\n
@Suppress("DIVISION_BY_ZERO")\n const val POSITIVE_INFINITY: Float = 1.0F / 0.0F\n\n
@JsName("NEGATIVE_INFINITY")\n @Suppress("DIVISION_BY_ZERO")\n const val
NEGATIVE_INFINITY: Float = -1.0F / 0.0F\n\n @JsName("NaN")\n
@Suppress("DIVISION_BY_ZERO")\n const val NaN: Float = -(0.0F / 0.0F)\n\n
@JsName("SIZE_BYTES")\n const val SIZE_BYTES = 4\n\n @JsName("SIZE_BITS")\n const val
SIZE_BITS = 32\n}\n\n@JsName("IntCompanionObject")\ninternal object IntCompanionObject {\n
@JsName("MIN_VALUE")\n val MIN_VALUE: Int = -2147483647 - 1\n\n @JsName("MAX_VALUE")\n
val MAX_VALUE: Int = 2147483647\n\n @JsName("SIZE_BYTES")\n const val SIZE_BYTES = 4\n\n
@JsName("SIZE_BITS")\n const val SIZE_BITS = 32\n}\n\n@JsName("LongCompanionObject")\ninternal
object LongCompanionObject {\n @JsName("MIN_VALUE")\n val MIN_VALUE: Long =
js("Kotlin.Long.MIN_VALUE")\n\n @JsName("MAX_VALUE")\n val MAX_VALUE: Long =
js("Kotlin.Long.MAX_VALUE")\n\n @JsName("SIZE_BYTES")\n const val SIZE_BYTES = 8\n\n
@JsName("SIZE_BITS")\n const val SIZE_BITS = 64\n}\n\n@JsName("ShortCompanionObject")\ninternal
object ShortCompanionObject {\n @JsName("MIN_VALUE")\n val MIN_VALUE: Short = -32768\n\n
@JsName("MAX_VALUE")\n val MAX_VALUE: Short = 32767\n\n @JsName("SIZE_BYTES")\n const
val SIZE_BYTES = 2\n\n @JsName("SIZE_BITS")\n const val SIZE_BITS =
16\n}\n\n@JsName("ByteCompanionObject")\ninternal object ByteCompanionObject {\n
@JsName("MIN_VALUE")\n val MIN_VALUE: Byte = -128\n\n @JsName("MAX_VALUE")\n val
MAX_VALUE: Byte = 127\n\n @JsName("SIZE_BYTES")\n const val SIZE_BYTES = 1\n\n
@JsName("SIZE_BITS")\n const val SIZE_BITS = 8\n}\n\n@JsName("CharCompanionObject")\ninternal
object CharCompanionObject {\n @JsName("MIN_VALUE")\n public const val MIN_VALUE: Char =
"\u0000"\n\n @JsName("MAX_VALUE")\n public const val MAX_VALUE: Char = "\uFFFF"\n\n
@JsName("MIN_HIGH_SURROGATE")\n public const val MIN_HIGH_SURROGATE: Char = "\uD800"\n\n
@JsName("MAX_HIGH_SURROGATE")\n public const val MAX_HIGH_SURROGATE: Char =
"\uDBFF"\n\n @JsName("MIN_LOW_SURROGATE")\n public const val MIN_LOW_SURROGATE: Char =
"\uDC00"\n\n @JsName("MAX_LOW_SURROGATE")\n public const val MAX_LOW_SURROGATE: Char =
"\uDFFF"\n\n @JsName("MIN_SURROGATE")\n public const val MIN_SURROGATE: Char =
MIN_HIGH_SURROGATE\n\n @JsName("MAX_SURROGATE")\n public const val MAX_SURROGATE:
Char = MAX_LOW_SURROGATE\n\n @JsName("SIZE_BYTES")\n const val SIZE_BYTES = 2\n\n
@JsName("SIZE_BITS")\n const val SIZE_BITS = 16\n}\n\ninternal object StringCompanionObject
{\n}\n\ninternal object BooleanCompanionObject {\n}\n\n", "/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n
*\n\n@file:kotlin.jvm.JvmMultifileClass\n\n@file:kotlin.jvm.JvmName("ArraysKt")\n\npackage
kotlin.collections\n\n/\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See:
https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\n/\n\nimport kotlin.random.*\nimport
kotlin.ranges.contains\nimport kotlin.ranges.reversed\n\n/**\n * Returns 1st *element* from the array.\n * \n * If the
size of this array is less than 1, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior
is unspecified.\n * \n * @kotlin.internal.InlineOnly\n * public inline operator fun <T> Array<out T>.component1(): T
{\n return get(0)\n}\n\n/**\n * Returns 1st *element* from the array.\n * \n * If the size of this array is less than 1,
throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n * @kotlin.internal.InlineOnly\n * public inline operator fun ByteArray.component1(): Byte {\n return
get(0)\n}\n\n/**\n * Returns 1st *element* from the array.\n * \n * If the size of this array is less than 1, throws an
[IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n * @kotlin.internal.InlineOnly\n * public inline operator fun ShortArray.component1(): Short {\n return
get(0)\n}\n\n/**\n * Returns 1st *element* from the array.\n * \n * If the size of this array is less than 1, throws an
[IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n

```

```

*\/n@kotlin.internal.InlineOnly\npublic inline operator fun IntArray.component1(): Int {\n  return
get(0)\n}\n\n**\n * Returns 1st *element* from the array.\n * \n * If the size of this array is less than 1, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\/n@kotlin.internal.InlineOnly\npublic inline operator fun LongArray.component1(): Long {\n  return
get(0)\n}\n\n**\n * Returns 1st *element* from the array.\n * \n * If the size of this array is less than 1, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\/n@kotlin.internal.InlineOnly\npublic inline operator fun FloatArray.component1(): Float {\n  return
get(0)\n}\n\n**\n * Returns 1st *element* from the array.\n * \n * If the size of this array is less than 1, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\/n@kotlin.internal.InlineOnly\npublic inline operator fun DoubleArray.component1(): Double {\n  return
get(0)\n}\n\n**\n * Returns 1st *element* from the array.\n * \n * If the size of this array is less than 1, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\/n@kotlin.internal.InlineOnly\npublic inline operator fun BooleanArray.component1(): Boolean {\n  return
get(0)\n}\n\n**\n * Returns 1st *element* from the array.\n * \n * If the size of this array is less than 1, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\/n@kotlin.internal.InlineOnly\npublic inline operator fun CharArray.component1(): Char {\n  return
get(0)\n}\n\n**\n * Returns 2nd *element* from the array.\n * \n * If the size of this array is less than 2, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\/n@kotlin.internal.InlineOnly\npublic inline operator fun <T> Array<out T>.component2(): T {\n  return
get(1)\n}\n\n**\n * Returns 2nd *element* from the array.\n * \n * If the size of this array is less than 2, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\/n@kotlin.internal.InlineOnly\npublic inline operator fun ByteArray.component2(): Byte {\n  return
get(1)\n}\n\n**\n * Returns 2nd *element* from the array.\n * \n * If the size of this array is less than 2, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\/n@kotlin.internal.InlineOnly\npublic inline operator fun ShortArray.component2(): Short {\n  return
get(1)\n}\n\n**\n * Returns 2nd *element* from the array.\n * \n * If the size of this array is less than 2, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\/n@kotlin.internal.InlineOnly\npublic inline operator fun IntArray.component2(): Int {\n  return
get(1)\n}\n\n**\n * Returns 2nd *element* from the array.\n * \n * If the size of this array is less than 2, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\/n@kotlin.internal.InlineOnly\npublic inline operator fun LongArray.component2(): Long {\n  return
get(1)\n}\n\n**\n * Returns 2nd *element* from the array.\n * \n * If the size of this array is less than 2, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\/n@kotlin.internal.InlineOnly\npublic inline operator fun FloatArray.component2(): Float {\n  return
get(1)\n}\n\n**\n * Returns 2nd *element* from the array.\n * \n * If the size of this array is less than 2, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\/n@kotlin.internal.InlineOnly\npublic inline operator fun DoubleArray.component2(): Double {\n  return
get(1)\n}\n\n**\n * Returns 2nd *element* from the array.\n * \n * If the size of this array is less than 2, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\/n@kotlin.internal.InlineOnly\npublic inline operator fun BooleanArray.component2(): Boolean {\n  return
get(1)\n}\n\n**\n * Returns 2nd *element* from the array.\n * \n * If the size of this array is less than 2, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\/n@kotlin.internal.InlineOnly\npublic inline operator fun CharArray.component2(): Char {\n  return
get(1)\n}\n\n**\n * Returns 3rd *element* from the array.\n * \n * If the size of this array is less than 3, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\/n@kotlin.internal.InlineOnly\npublic inline operator fun <T> Array<out T>.component3(): T {\n  return
get(2)\n}\n\n**\n * Returns 3rd *element* from the array.\n * \n * If the size of this array is less than 3, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n

```

```

*\n@kotlin.internal.InlineOnly\npublic inline operator fun ByteArray.component3(): Byte {\n    return
get(2)\n}\n\n/**\n * Returns 3rd *element* from the array.\n * \n * If the size of this array is less than 3, throws an
[IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun ShortArray.component3(): Short {\n    return
get(2)\n}\n\n/**\n * Returns 3rd *element* from the array.\n * \n * If the size of this array is less than 3, throws an
[IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun IntArray.component3(): Int {\n    return
get(2)\n}\n\n/**\n * Returns 3rd *element* from the array.\n * \n * If the size of this array is less than 3, throws an
[IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun LongArray.component3(): Long {\n    return
get(2)\n}\n\n/**\n * Returns 3rd *element* from the array.\n * \n * If the size of this array is less than 3, throws an
[IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun FloatArray.component3(): Float {\n    return
get(2)\n}\n\n/**\n * Returns 3rd *element* from the array.\n * \n * If the size of this array is less than 3, throws an
[IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun DoubleArray.component3(): Double {\n    return
get(2)\n}\n\n/**\n * Returns 3rd *element* from the array.\n * \n * If the size of this array is less than 3, throws an
[IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun BooleanArray.component3(): Boolean {\n    return
get(2)\n}\n\n/**\n * Returns 3rd *element* from the array.\n * \n * If the size of this array is less than 3, throws an
[IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun CharArray.component3(): Char {\n    return
get(2)\n}\n\n/**\n * Returns 4th *element* from the array.\n * \n * If the size of this array is less than 4, throws an
[IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun <T> Array<out T>.component4(): T {\n    return
get(3)\n}\n\n/**\n * Returns 4th *element* from the array.\n * \n * If the size of this array is less than 4, throws an
[IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun ByteArray.component4(): Byte {\n    return
get(3)\n}\n\n/**\n * Returns 4th *element* from the array.\n * \n * If the size of this array is less than 4, throws an
[IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun ShortArray.component4(): Short {\n    return
get(3)\n}\n\n/**\n * Returns 4th *element* from the array.\n * \n * If the size of this array is less than 4, throws an
[IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun IntArray.component4(): Int {\n    return
get(3)\n}\n\n/**\n * Returns 4th *element* from the array.\n * \n * If the size of this array is less than 4, throws an
[IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun LongArray.component4(): Long {\n    return
get(3)\n}\n\n/**\n * Returns 4th *element* from the array.\n * \n * If the size of this array is less than 4, throws an
[IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun FloatArray.component4(): Float {\n    return
get(3)\n}\n\n/**\n * Returns 4th *element* from the array.\n * \n * If the size of this array is less than 4, throws an
[IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun DoubleArray.component4(): Double {\n    return
get(3)\n}\n\n/**\n * Returns 4th *element* from the array.\n * \n * If the size of this array is less than 4, throws an
[IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun BooleanArray.component4(): Boolean {\n    return
get(3)\n}\n\n/**\n * Returns 4th *element* from the array.\n * \n * If the size of this array is less than 4, throws an
[IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n

```

```

*\/n@kotlin.internal.InlineOnly\npublic inline operator fun CharArray.component4(): Char {\n    return
get(3)\n}\n\n/**\n * Returns 5th *element* from the array.\n * \n * If the size of this array is less than 5, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\/n@kotlin.internal.InlineOnly\npublic inline operator fun <T> Array<out T>.component5(): T {\n    return
get(4)\n}\n\n/**\n * Returns 5th *element* from the array.\n * \n * If the size of this array is less than 5, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\/n@kotlin.internal.InlineOnly\npublic inline operator fun ByteArray.component5(): Byte {\n    return
get(4)\n}\n\n/**\n * Returns 5th *element* from the array.\n * \n * If the size of this array is less than 5, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\/n@kotlin.internal.InlineOnly\npublic inline operator fun ShortArray.component5(): Short {\n    return
get(4)\n}\n\n/**\n * Returns 5th *element* from the array.\n * \n * If the size of this array is less than 5, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\/n@kotlin.internal.InlineOnly\npublic inline operator fun IntArray.component5(): Int {\n    return
get(4)\n}\n\n/**\n * Returns 5th *element* from the array.\n * \n * If the size of this array is less than 5, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\/n@kotlin.internal.InlineOnly\npublic inline operator fun LongArray.component5(): Long {\n    return
get(4)\n}\n\n/**\n * Returns 5th *element* from the array.\n * \n * If the size of this array is less than 5, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\/n@kotlin.internal.InlineOnly\npublic inline operator fun FloatArray.component5(): Float {\n    return
get(4)\n}\n\n/**\n * Returns 5th *element* from the array.\n * \n * If the size of this array is less than 5, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\/n@kotlin.internal.InlineOnly\npublic inline operator fun DoubleArray.component5(): Double {\n    return
get(4)\n}\n\n/**\n * Returns 5th *element* from the array.\n * \n * If the size of this array is less than 5, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\/n@kotlin.internal.InlineOnly\npublic inline operator fun BooleanArray.component5(): Boolean {\n    return
get(4)\n}\n\n/**\n * Returns 5th *element* from the array.\n * \n * If the size of this array is less than 5, throws an
[IndexOutOfBoundsExcep] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\/n@kotlin.internal.InlineOnly\npublic inline operator fun CharArray.component5(): Char {\n    return
get(4)\n}\n\n/**\n * Returns `true` if [element] is found in the array.\n */\n\npublic operator fun
<@kotlin.internal.OnlyInputTypes T> Array<out T>.contains(element: T): Boolean {\n    return indexOf(element)
>= 0\n}\n\n/**\n * Returns `true` if [element] is found in the array.\n */\n\npublic operator fun
ByteArray.contains(element: Byte): Boolean {\n    return indexOf(element) >= 0\n}\n\n/**\n * Returns `true` if
[element] is found in the array.\n */\n\npublic operator fun ShortArray.contains(element: Short): Boolean {\n    return
indexOf(element) >= 0\n}\n\n/**\n * Returns `true` if [element] is found in the array.\n */\n\npublic operator fun
IntArray.contains(element: Int): Boolean {\n    return indexOf(element) >= 0\n}\n\n/**\n * Returns `true` if
[element] is found in the array.\n */\n\npublic operator fun LongArray.contains(element: Long): Boolean {\n    return
indexOf(element) >= 0\n}\n\n/**\n * Returns `true` if [element] is found in the array.\n */\n\n@Deprecated("The
function has unclear behavior when searching for NaN or zero values and will be removed soon. Use 'any { it ==
element }' instead to continue using this behavior, or '.asList().contains(element: T)' to get the same search behavior
as in a list.", ReplaceWith("any { it == element }"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince =
"1.6")\n@Suppress("DEPRECATION_ERROR")\npublic operator fun FloatArray.contains(element: Float):
Boolean {\n    return indexOf(element) >= 0\n}\n\n/**\n * Returns `true` if [element] is found in the array.\n */\n\n@Deprecated("The
function has unclear behavior when searching for NaN or zero values and will be removed soon. Use 'any { it == element }'
instead to continue using this behavior, or '.asList().contains(element: T)' to get the same search behavior as in a list.",
ReplaceWith("any { it == element }"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince =
"1.6")\n@Suppress("DEPRECATION_ERROR")\npublic operator fun DoubleArray.contains(element: Double):
Boolean {\n    return indexOf(element) >= 0\n}\n\n/**\n * Returns `true` if [element] is found in the array.\n

```

```

*  

public operator fun BooleanArray.contains(element: Boolean): Boolean {  

    return indexOf(element) >= 0  

}  

Returns `true` if [element] is found in the array.  

public operator fun  

CharArray.contains(element: Char): Boolean {  

    return indexOf(element) >= 0  

}  

Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.  

@sample samples.collections.Collections.Elements.elementAt  

public expect fun <T> Array<out T>.elementAt(index: Int): T  

Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.  

@sample samples.collections.Collections.Elements.elementAt  

public expect fun ByteArray.elementAt(index: Int): Byte  

Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.  

@sample samples.collections.Collections.Elements.elementAt  

public expect fun ShortArray.elementAt(index: Int): Short  

Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.  

@sample samples.collections.Collections.Elements.elementAt  

public expect fun IntArray.elementAt(index: Int): Int  

Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.  

@sample samples.collections.Collections.Elements.elementAt  

public expect fun LongArray.elementAt(index: Int): Long  

Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.  

@sample samples.collections.Collections.Elements.elementAt  

public expect fun FloatArray.elementAt(index: Int): Float  

Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.  

@sample samples.collections.Collections.Elements.elementAt  

public expect fun DoubleArray.elementAt(index: Int): Double  

Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.  

@sample samples.collections.Collections.Elements.elementAt  

public expect fun BooleanArray.elementAt(index: Int): Boolean  

Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.  

@sample samples.collections.Collections.Elements.elementAt  

public expect fun CharArray.elementAt(index: Int): Char  

Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.  

@sample samples.collections.Collections.Elements.elementAtOrElse  

@kotlin.internal.InlineOnly  

public inline fun <T> Array<out T>.elementAtOrElse(index: Int, defaultValue: (Int) -> T): T {  

    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)  

}  

Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.  

@sample samples.collections.Collections.Elements.elementAtOrElse  

@kotlin.internal.InlineOnly  

public inline fun ByteArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Byte): Byte {  

    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)  

}  

Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.  

@sample samples.collections.Collections.Elements.elementAtOrElse  

@kotlin.internal.InlineOnly  

public inline fun ShortArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Short): Short {  

    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)  

}  

Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.  

@sample samples.collections.Collections.Elements.elementAtOrElse  

@kotlin.internal.InlineOnly  

public inline fun IntArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Int): Int {  

    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)  

}  

Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.  

@sample samples.collections.Collections.Elements.elementAtOrElse  

@kotlin.internal.InlineOnly  

public inline fun LongArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Long): Long {  

    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)  

}  

Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.  

@sample samples.collections.Collections.Elements.elementAtOrElse  

@kotlin.internal.InlineOnly  

public inline fun

```

```

FloatArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Float): Float {
    return if (index >= 0 && index <=
lastIndex) get(index) else defaultValue(index)}\n\n/**\n * Returns an element at the given [index] or the result of
calling the [defaultValue] function if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrElse\n */\n@kotlin.internal.InlineOnly\npublic inline fun
DoubleArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Double): Double {
    return if (index >= 0 &&
index <= lastIndex) get(index) else defaultValue(index)}\n\n/**\n * Returns an element at the given [index] or the
result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrElse\n */\n@kotlin.internal.InlineOnly\npublic inline fun
BooleanArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Boolean): Boolean {
    return if (index >= 0 &&
index <= lastIndex) get(index) else defaultValue(index)}\n\n/**\n * Returns an element at the given [index] or the
result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrElse\n */\n@kotlin.internal.InlineOnly\npublic inline fun
CharArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Char): Char {
    return if (index >= 0 && index <=
lastIndex) get(index) else defaultValue(index)}\n\n/**\n * Returns an element at the given [index] or `null` if the
[index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrNull\n */\n@kotlin.internal.InlineOnly\npublic inline fun
<T> Array<out T>.elementAtOrNull(index: Int): T? {
    return this.getOrNull(index)}\n\n/**\n * Returns an
element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrNull\n */\n@kotlin.internal.InlineOnly\npublic inline fun
ByteArray.elementAtOrNull(index: Int): Byte? {
    return this.getOrNull(index)}\n\n/**\n * Returns an element
at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrNull\n */\n@kotlin.internal.InlineOnly\npublic inline fun
ShortArray.elementAtOrNull(index: Int): Short? {
    return this.getOrNull(index)}\n\n/**\n * Returns an element
at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrNull\n */\n@kotlin.internal.InlineOnly\npublic inline fun
IntArray.elementAtOrNull(index: Int): Int? {
    return this.getOrNull(index)}\n\n/**\n * Returns an element at
the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrNull\n */\n@kotlin.internal.InlineOnly\npublic inline fun
LongArray.elementAtOrNull(index: Int): Long? {
    return this.getOrNull(index)}\n\n/**\n * Returns an element
at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrNull\n */\n@kotlin.internal.InlineOnly\npublic inline fun
FloatArray.elementAtOrNull(index: Int): Float? {
    return this.getOrNull(index)}\n\n/**\n * Returns an element
at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrNull\n */\n@kotlin.internal.InlineOnly\npublic inline fun
DoubleArray.elementAtOrNull(index: Int): Double? {
    return this.getOrNull(index)}\n\n/**\n * Returns an
element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrNull\n */\n@kotlin.internal.InlineOnly\npublic inline fun
BooleanArray.elementAtOrNull(index: Int): Boolean? {
    return this.getOrNull(index)}\n\n/**\n * Returns an
element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrNull\n */\n@kotlin.internal.InlineOnly\npublic inline fun
CharArray.elementAtOrNull(index: Int): Char? {
    return this.getOrNull(index)}\n\n/**\n * Returns the first
element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out
T>.find(predicate: (T) -> Boolean): T? {
    return firstOrNull(predicate)}\n\n/**\n * Returns the first element
matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun
ByteArray.find(predicate: (Byte) -> Boolean): Byte? {
    return firstOrNull(predicate)}\n\n/**\n * Returns the
first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample

```

```

samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun
ShortArray.find(predicate: (Short) -> Boolean): Short? {\n    return firstOrNull(predicate)\n}\n\n/**\n * Returns the
first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun
IntArray.find(predicate: (Int) -> Boolean): Int? {\n    return firstOrNull(predicate)\n}\n\n/**\n * Returns the first
element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun
LongArray.find(predicate: (Long) -> Boolean): Long? {\n    return firstOrNull(predicate)\n}\n\n/**\n * Returns the
first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun
FloatArray.find(predicate: (Float) -> Boolean): Float? {\n    return firstOrNull(predicate)\n}\n\n/**\n * Returns the
first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun
DoubleArray.find(predicate: (Double) -> Boolean): Double? {\n    return firstOrNull(predicate)\n}\n\n/**\n *
Returns the first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun
BooleanArray.find(predicate: (Boolean) -> Boolean): Boolean? {\n    return firstOrNull(predicate)\n}\n\n/**\n *
Returns the first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun
CharArray.find(predicate: (Char) -> Boolean): Char? {\n    return firstOrNull(predicate)\n}\n\n/**\n * Returns the
last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out
T>.findLast(predicate: (T) -> Boolean): T? {\n    return lastOrNull(predicate)\n}\n\n/**\n * Returns the last element
matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun
ByteArray.findLast(predicate: (Byte) -> Boolean): Byte? {\n    return lastOrNull(predicate)\n}\n\n/**\n * Returns
the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun
ShortArray.findLast(predicate: (Short) -> Boolean): Short? {\n    return lastOrNull(predicate)\n}\n\n/**\n * Returns
the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun
IntArray.findLast(predicate: (Int) -> Boolean): Int? {\n    return lastOrNull(predicate)\n}\n\n/**\n * Returns the last
element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun
LongArray.findLast(predicate: (Long) -> Boolean): Long? {\n    return lastOrNull(predicate)\n}\n\n/**\n * Returns
the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun
FloatArray.findLast(predicate: (Float) -> Boolean): Float? {\n    return lastOrNull(predicate)\n}\n\n/**\n * Returns
the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun
DoubleArray.findLast(predicate: (Double) -> Boolean): Double? {\n    return lastOrNull(predicate)\n}\n\n/**\n *
Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun
BooleanArray.findLast(predicate: (Boolean) -> Boolean): Boolean? {\n    return lastOrNull(predicate)\n}\n\n/**\n *
Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun
CharArray.findLast(predicate: (Char) -> Boolean): Char? {\n    return lastOrNull(predicate)\n}\n\n/**\n * Returns
first element.\n * @throws [NoSuchElementException] if the array is empty.\n */\n@kotlin.internal.InlineOnly\npublic fun <T> Array<out

```

```

T>.first(): T { \n  if (isEmpty())\n    throw NoSuchElementException("Array is empty.")\n  return
this[0]\n}\n\n/**\n * Returns first element.\n * @throws [NoSuchElementException] if the array is empty.\n
*/\npublic fun ByteArray.first(): Byte { \n  if (isEmpty())\n    throw NoSuchElementException("Array is
empty.")\n  return this[0]\n}\n\n/**\n * Returns first element.\n * @throws [NoSuchElementException] if the
array is empty.\n */\npublic fun ShortArray.first(): Short { \n  if (isEmpty())\n    throw
NoSuchElementException("Array is empty.")\n  return this[0]\n}\n\n/**\n * Returns first element.\n * @throws
[NoSuchElementException] if the array is empty.\n */\npublic fun IntArray.first(): Int { \n  if (isEmpty())\n
throw NoSuchElementException("Array is empty.")\n  return this[0]\n}\n\n/**\n * Returns first element.\n *
@throws [NoSuchElementException] if the array is empty.\n */\npublic fun LongArray.first(): Long { \n  if
(isEmpty())\n    throw NoSuchElementException("Array is empty.")\n  return this[0]\n}\n\n/**\n * Returns first
element.\n * @throws [NoSuchElementException] if the array is empty.\n */\npublic fun FloatArray.first(): Float
{ \n  if (isEmpty())\n    throw NoSuchElementException("Array is empty.")\n  return this[0]\n}\n\n/**\n *
Returns first element.\n * @throws [NoSuchElementException] if the array is empty.\n */\npublic fun
DoubleArray.first(): Double { \n  if (isEmpty())\n    throw NoSuchElementException("Array is empty.")\n
return this[0]\n}\n\n/**\n * Returns first element.\n * @throws [NoSuchElementException] if the array is empty.\n
*/\npublic fun BooleanArray.first(): Boolean { \n  if (isEmpty())\n    throw NoSuchElementException("Array is
empty.")\n  return this[0]\n}\n\n/**\n * Returns first element.\n * @throws [NoSuchElementException] if the
array is empty.\n */\npublic fun CharArray.first(): Char { \n  if (isEmpty())\n    throw
NoSuchElementException("Array is empty.")\n  return this[0]\n}\n\n/**\n * Returns the first element matching
the given [predicate].\n * @throws [NoSuchElementException] if no such element is found.\n */\npublic inline fun
<T> Array<out T>.first(predicate: (T) -> Boolean): T { \n  for (element in this) if (predicate(element)) return
element\n  throw NoSuchElementException("Array contains no element matching the predicate.")\n}\n\n/**\n *
Returns the first element matching the given [predicate].\n * @throws [NoSuchElementException] if no such
element is found.\n */\npublic inline fun ByteArray.first(predicate: (Byte) -> Boolean): Byte { \n  for (element in
this) if (predicate(element)) return element\n  throw NoSuchElementException("Array contains no element
matching the predicate.")\n}\n\n/**\n * Returns the first element matching the given [predicate].\n * @throws
[NoSuchElementException] if no such element is found.\n */\npublic inline fun ShortArray.first(predicate: (Short) -
> Boolean): Short { \n  for (element in this) if (predicate(element)) return element\n  throw
NoSuchElementException("Array contains no element matching the predicate.")\n}\n\n/**\n * Returns the first
element matching the given [predicate].\n * @throws [NoSuchElementException] if no such element is found.\n
*/\npublic inline fun IntArray.first(predicate: (Int) -> Boolean): Int { \n  for (element in this) if (predicate(element))
return element\n  throw NoSuchElementException("Array contains no element matching the
predicate.")\n}\n\n/**\n * Returns the first element matching the given [predicate].\n * @throws
[NoSuchElementException] if no such element is found.\n */\npublic inline fun LongArray.first(predicate: (Long) -
> Boolean): Long { \n  for (element in this) if (predicate(element)) return element\n  throw
NoSuchElementException("Array contains no element matching the predicate.")\n}\n\n/**\n * Returns the first
element matching the given [predicate].\n * @throws [NoSuchElementException] if no such element is found.\n
*/\npublic inline fun FloatArray.first(predicate: (Float) -> Boolean): Float { \n  for (element in this) if
(predicate(element)) return element\n  throw NoSuchElementException("Array contains no element matching the
predicate.")\n}\n\n/**\n * Returns the first element matching the given [predicate].\n * @throws
[NoSuchElementException] if no such element is found.\n */\npublic inline fun DoubleArray.first(predicate:
(Double) -> Boolean): Double { \n  for (element in this) if (predicate(element)) return element\n  throw
NoSuchElementException("Array contains no element matching the predicate.")\n}\n\n/**\n * Returns the first
element matching the given [predicate].\n * @throws [NoSuchElementException] if no such element is found.\n
*/\npublic inline fun BooleanArray.first(predicate: (Boolean) -> Boolean): Boolean { \n  for (element in this) if
(predicate(element)) return element\n  throw NoSuchElementException("Array contains no element matching the
predicate.")\n}\n\n/**\n * Returns the first element matching the given [predicate].\n * @throws
[NoSuchElementException] if no such element is found.\n */\npublic inline fun CharArray.first(predicate: (Char) ->

```

Boolean): Char { \n for (element in this) if (predicate(element)) return element\n throw
NoSuchElementException("Array contains no element matching the predicate.")\n}\n\n/**\n * Returns the first
non-null value produced by [transform] function being applied to elements of this array in iteration order,\n * or
throws [NoSuchElementException] if no non-null value was produced.\n * \n * @sample
samples.collections.Collections.Transformations.firstNotNullOf\n

```
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Any> Array<out  

T>.firstNotNullOf(transform: (T) -> R?): R { \n return firstNotNullOfOrNull(transform) ?: throw  

NoSuchElementException("No element of the array was transformed to a non-null value.")\n}\n\n/**\n * Returns  

the first non-null value produced by [transform] function being applied to elements of this array in iteration order,\n * or `null` if no non-null value was produced.\n * \n * @sample  

samples.collections.Collections.Transformations.firstNotNullOf\n
```

```
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Any> Array<out  

T>.firstNotNullOfOrNull(transform: (T) -> R?): R? { \n for (element in this) { \n val result =  

transform(element)\n if (result != null) { \n return result\n } \n } \n return null\n}\n\n/**\n *  

Returns the first element, or `null` if the array is empty.\n */\npublic fun <T> Array<out T>.firstOrNull(): T? { \n  

return if (isEmpty()) null else this[0]\n}\n\n/**\n * Returns the first element, or `null` if the array is empty.\n */\npublic fun ByteArray.firstOrNull(): Byte? { \n return if (isEmpty()) null else this[0]\n}\n\n/**\n * Returns the  

first element, or `null` if the array is empty.\n */\npublic fun ShortArray.firstOrNull(): Short? { \n return if  

(isEmpty()) null else this[0]\n}\n\n/**\n * Returns the first element, or `null` if the array is empty.\n */\npublic fun  

IntArray.firstOrNull(): Int? { \n return if (isEmpty()) null else this[0]\n}\n\n/**\n * Returns the first element, or  

`null` if the array is empty.\n */\npublic fun LongArray.firstOrNull(): Long? { \n return if (isEmpty()) null else  

this[0]\n}\n\n/**\n * Returns the first element, or `null` if the array is empty.\n */\npublic fun  

FloatArray.firstOrNull(): Float? { \n return if (isEmpty()) null else this[0]\n}\n\n/**\n * Returns the first element,  

or `null` if the array is empty.\n */\npublic fun DoubleArray.firstOrNull(): Double? { \n return if (isEmpty()) null  

else this[0]\n}\n\n/**\n * Returns the first element, or `null` if the array is empty.\n */\npublic fun  

BooleanArray.firstOrNull(): Boolean? { \n return if (isEmpty()) null else this[0]\n}\n\n/**\n * Returns the first  

element, or `null` if the array is empty.\n */\npublic fun CharArray.firstOrNull(): Char? { \n return if (isEmpty())  

null else this[0]\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if element was not  

found.\n */\npublic inline fun <T> Array<out T>.firstOrNull(predicate: (T) -> Boolean): T? { \n for (element in  

this) if (predicate(element)) return element\n return null\n}\n\n/**\n * Returns the first element matching the given  

[predicate], or `null` if element was not found.\n */\npublic inline fun ByteArray.firstOrNull(predicate: (Byte) ->  

Boolean): Byte? { \n for (element in this) if (predicate(element)) return element\n return null\n}\n\n/**\n *  

Returns the first element matching the given [predicate], or `null` if element was not found.\n */\npublic inline fun  

ShortArray.firstOrNull(predicate: (Short) -> Boolean): Short? { \n for (element in this) if (predicate(element))  

return element\n return null\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if  

element was not found.\n */\npublic inline fun IntArray.firstOrNull(predicate: (Int) -> Boolean): Int? { \n for  

(element in this) if (predicate(element)) return element\n return null\n}\n\n/**\n * Returns the first element  

matching the given [predicate], or `null` if element was not found.\n */\npublic inline fun  

LongArray.firstOrNull(predicate: (Long) -> Boolean): Long? { \n for (element in this) if (predicate(element))  

return element\n return null\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if  

element was not found.\n */\npublic inline fun FloatArray.firstOrNull(predicate: (Float) -> Boolean): Float? { \n  

for (element in this) if (predicate(element)) return element\n return null\n}\n\n/**\n * Returns the first element  

matching the given [predicate], or `null` if element was not found.\n */\npublic inline fun  

DoubleArray.firstOrNull(predicate: (Double) -> Boolean): Double? { \n for (element in this) if  

(predicate(element)) return element\n return null\n}\n\n/**\n * Returns the first element matching the given  

[predicate], or `null` if element was not found.\n */\npublic inline fun BooleanArray.firstOrNull(predicate:  

(Boolean) -> Boolean): Boolean? { \n for (element in this) if (predicate(element)) return element\n return  

null\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if element was not found.\n
```

```

*\npublic inline fun CharArray.firstOrNull(predicate: (Char) -> Boolean): Char? {\n    for (element in this) if
(predicate(element)) return element\n    return null\n}\n\n/**\n * Returns an element at the given [index] or the
result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n
*\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.getOrNull(index: Int, defaultValue: (Int) ->
T): T {\n    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns
an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of
this array.\n
*\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.getOrNull(index: Int, defaultValue: (Int) ->
Byte): Byte {\n    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n *
Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of
bounds of this array.\n
*\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.getOrNull(index: Int,
defaultValue: (Int) -> Short): Short {\n    return if (index >= 0 && index <= lastIndex) get(index) else
defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue]
function if the [index] is out of bounds of this array.\n
*\n@kotlin.internal.InlineOnly\npublic inline fun
IntArray.getOrNull(index: Int, defaultValue: (Int) -> Int): Int {\n    return if (index >= 0 && index <=
lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of
calling the [defaultValue] function if the [index] is out of bounds of this array.\n
*\n@kotlin.internal.InlineOnly\npublic inline
fun LongArray.getOrNull(index: Int, defaultValue: (Int) -> Long): Long {\n    return if (index >= 0 && index <=
lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of
calling the [defaultValue] function if the [index] is out of bounds of this array.\n
*\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.getOrNull(index: Int, defaultValue: (Int) -> Float):
Float {\n    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns
an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of
this array.\n
*\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.getOrNull(index: Int, defaultValue:
(Int) -> Double): Double {\n    return if (index >= 0 && index <= lastIndex) get(index) else
defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue]
function if the [index] is out of bounds of this array.\n
*\n@kotlin.internal.InlineOnly\npublic inline fun
BooleanArray.getOrNull(index: Int, defaultValue: (Int) -> Boolean): Boolean {\n    return if (index >= 0 && index
<= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result
of calling the [defaultValue] function if the [index] is out of bounds of this array.\n
*\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.getOrNull(index: Int, defaultValue: (Int) -> Char):
Char {\n    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns
an element at the given [index] or `null` if the [index] is out of bounds of this array.\n
*\n * @sample
samples.collections.Collections.Elements.getOrNull\n
*\npublic fun <T> Array<out T>.getOrNull(index: Int): T?
{\n    return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns an element at the
given [index] or `null` if the [index] is out of bounds of this array.\n
*\n * @sample
samples.collections.Collections.Elements.getOrNull\n
*\npublic fun ByteArray.getOrNull(index: Int): Byte? {\n
return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns an element at the given
[index] or `null` if the [index] is out of bounds of this array.\n
*\n * @sample
samples.collections.Collections.Elements.getOrNull\n
*\npublic fun ShortArray.getOrNull(index: Int): Short? {\n
return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns an element at the given
[index] or `null` if the [index] is out of bounds of this array.\n
*\n * @sample
samples.collections.Collections.Elements.getOrNull\n
*\npublic fun IntArray.getOrNull(index: Int): Int? {\n
return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns an element at the given
[index] or `null` if the [index] is out of bounds of this array.\n
*\n * @sample
samples.collections.Collections.Elements.getOrNull\n
*\npublic fun LongArray.getOrNull(index: Int): Long? {\n
return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns an element at the given
[index] or `null` if the [index] is out of bounds of this array.\n
*\n * @sample
samples.collections.Collections.Elements.getOrNull\n
*\npublic fun FloatArray.getOrNull(index: Int): Float? {\n

```



```

(predicate(this[index])) {\n      return index\n    }\n  }\n  return -1\n}\n\n/**\n * Returns index of the first
element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun
IntArray.indexOfFirst(predicate: (Int) -> Boolean): Int {\n  for (index in indices) {\n    if (predicate(this[index]))
{\n      return index\n    }\n  }\n  return -1\n}\n\n/**\n * Returns index of the first element matching the
given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun
LongArray.indexOfFirst(predicate: (Long) -> Boolean): Int {\n  for (index in indices) {\n    if
(predicate(this[index])) {\n      return index\n    }\n  }\n  return -1\n}\n\n/**\n * Returns index of the first
element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun
FloatArray.indexOfFirst(predicate: (Float) -> Boolean): Int {\n  for (index in indices) {\n    if
(predicate(this[index])) {\n      return index\n    }\n  }\n  return -1\n}\n\n/**\n * Returns index of the first
element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun
DoubleArray.indexOfFirst(predicate: (Double) -> Boolean): Int {\n  for (index in indices) {\n    if
(predicate(this[index])) {\n      return index\n    }\n  }\n  return -1\n}\n\n/**\n * Returns index of the first
element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun
BooleanArray.indexOfFirst(predicate: (Boolean) -> Boolean): Int {\n  for (index in indices) {\n    if
(predicate(this[index])) {\n      return index\n    }\n  }\n  return -1\n}\n\n/**\n * Returns index of the first
element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun
CharArray.indexOfFirst(predicate: (Char) -> Boolean): Int {\n  for (index in indices) {\n    if
(predicate(this[index])) {\n      return index\n    }\n  }\n  return -1\n}\n\n/**\n * Returns index of the last
element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun <T>
Array<out T>.indexOfLast(predicate: (T) -> Boolean): Int {\n  for (index in indices.reversed()) {\n    if
(predicate(this[index])) {\n      return index\n    }\n  }\n  return -1\n}\n\n/**\n * Returns index of the last
element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun
ByteArray.indexOfLast(predicate: (Byte) -> Boolean): Int {\n  for (index in indices.reversed()) {\n    if
(predicate(this[index])) {\n      return index\n    }\n  }\n  return -1\n}\n\n/**\n * Returns index of the last
element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun
ShortArray.indexOfLast(predicate: (Short) -> Boolean): Int {\n  for (index in indices.reversed()) {\n    if
(predicate(this[index])) {\n      return index\n    }\n  }\n  return -1\n}\n\n/**\n * Returns index of the last
element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun
IntArray.indexOfLast(predicate: (Int) -> Boolean): Int {\n  for (index in indices.reversed()) {\n    if
(predicate(this[index])) {\n      return index\n    }\n  }\n  return -1\n}\n\n/**\n * Returns index of the last
element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun
LongArray.indexOfLast(predicate: (Long) -> Boolean): Int {\n  for (index in indices.reversed()) {\n    if
(predicate(this[index])) {\n      return index\n    }\n  }\n  return -1\n}\n\n/**\n * Returns index of the last
element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun
FloatArray.indexOfLast(predicate: (Float) -> Boolean): Int {\n  for (index in indices.reversed()) {\n    if
(predicate(this[index])) {\n      return index\n    }\n  }\n  return -1\n}\n\n/**\n * Returns index of the last
element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun
DoubleArray.indexOfLast(predicate: (Double) -> Boolean): Int {\n  for (index in indices.reversed()) {\n    if
(predicate(this[index])) {\n      return index\n    }\n  }\n  return -1\n}\n\n/**\n * Returns index of the last
element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun
BooleanArray.indexOfLast(predicate: (Boolean) -> Boolean): Int {\n  for (index in indices.reversed()) {\n    if
(predicate(this[index])) {\n      return index\n    }\n  }\n  return -1\n}\n\n/**\n * Returns index of the last
element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun
CharArray.indexOfLast(predicate: (Char) -> Boolean): Int {\n  for (index in indices.reversed()) {\n    if
(predicate(this[index])) {\n      return index\n    }\n  }\n  return -1\n}\n\n/**\n * Returns the last element.\n */\n
\n * @throws NoSuchElementException if the array is empty.\n */\n\n * @sample
samples.collections.Collections.Elements.last\n */\npublic fun <T> Array<out T>.last(): T {\n  if (isEmpty())\n

```

```

throw NoSuchElementException("Array is empty.")\n    return this[lastIndex]\n}\n\n/**\n * Returns the last element.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic fun ByteArray.last(): Byte {\n    if (isEmpty())\n        throw NoSuchElementException("Array is empty.")\n    return this[lastIndex]\n}\n\n/**\n * Returns the last element.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic fun ShortArray.last(): Short {\n    if (isEmpty())\n        throw NoSuchElementException("Array is empty.")\n    return this[lastIndex]\n}\n\n/**\n * Returns the last element.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic fun IntArray.last(): Int {\n    if (isEmpty())\n        throw NoSuchElementException("Array is empty.")\n    return this[lastIndex]\n}\n\n/**\n * Returns the last element.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic fun LongArray.last(): Long {\n    if (isEmpty())\n        throw NoSuchElementException("Array is empty.")\n    return this[lastIndex]\n}\n\n/**\n * Returns the last element.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic fun FloatArray.last(): Float {\n    if (isEmpty())\n        throw NoSuchElementException("Array is empty.")\n    return this[lastIndex]\n}\n\n/**\n * Returns the last element.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic fun DoubleArray.last(): Double {\n    if (isEmpty())\n        throw NoSuchElementException("Array is empty.")\n    return this[lastIndex]\n}\n\n/**\n * Returns the last element.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic fun BooleanArray.last(): Boolean {\n    if (isEmpty())\n        throw NoSuchElementException("Array is empty.")\n    return this[lastIndex]\n}\n\n/**\n * Returns the last element.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic fun CharArray.last(): Char {\n    if (isEmpty())\n        throw NoSuchElementException("Array is empty.")\n    return this[lastIndex]\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws NoSuchElementException if no such element is found.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic inline fun <T> Array<out T>.last(predicate: (T) -> Boolean): T {\n    for (index in this.indices.reversed()) {\n        val element = this[index]\n        if (predicate(element)) return element\n    }\n    throw NoSuchElementException("Array contains no element matching the predicate.")\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws NoSuchElementException if no such element is found.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic inline fun ByteArray.last(predicate: (Byte) -> Boolean): Byte {\n    for (index in this.indices.reversed()) {\n        val element = this[index]\n        if (predicate(element)) return element\n    }\n    throw NoSuchElementException("Array contains no element matching the predicate.")\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws NoSuchElementException if no such element is found.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic inline fun ShortArray.last(predicate: (Short) -> Boolean): Short {\n    for (index in this.indices.reversed()) {\n        val element = this[index]\n        if (predicate(element)) return element\n    }\n    throw NoSuchElementException("Array contains no element matching the predicate.")\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws NoSuchElementException if no such element is found.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic inline fun IntArray.last(predicate: (Int) -> Boolean): Int {\n    for (index in this.indices.reversed()) {\n        val element = this[index]\n        if (predicate(element)) return element\n    }\n    throw NoSuchElementException("Array contains no element matching the predicate.")\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws NoSuchElementException if no such element is found.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic inline fun LongArray.last(predicate: (Long) -> Boolean): Long {\n    for (index in this.indices.reversed()) {\n        val element = this[index]\n        if (predicate(element))

```

```

return element\n } throw NoSuchElementException("Array contains no element matching the
predicate.")\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws
NoSuchElementException if no such element is found.\n * \n * @sample
samples.collections.Collections.Elements.last\n */\npublic inline fun FloatArray.last(predicate: (Float) -> Boolean):
Float {\n for (index in this.indices.reversed()) {\n val element = this[index]\n if (predicate(element))
return element\n }\n } throw NoSuchElementException("Array contains no element matching the
predicate.")\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws
NoSuchElementException if no such element is found.\n * \n * @sample
samples.collections.Collections.Elements.last\n */\npublic inline fun DoubleArray.last(predicate: (Double) ->
Boolean): Double {\n for (index in this.indices.reversed()) {\n val element = this[index]\n if
(predicate(element)) return element\n }\n } throw NoSuchElementException("Array contains no element
matching the predicate.")\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws
NoSuchElementException if no such element is found.\n * \n * @sample
samples.collections.Collections.Elements.last\n */\npublic inline fun BooleanArray.last(predicate: (Boolean) ->
Boolean): Boolean {\n for (index in this.indices.reversed()) {\n val element = this[index]\n if
(predicate(element)) return element\n }\n } throw NoSuchElementException("Array contains no element
matching the predicate.")\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws
NoSuchElementException if no such element is found.\n * \n * @sample
samples.collections.Collections.Elements.last\n */\npublic inline fun CharArray.last(predicate: (Char) -> Boolean):
Char {\n for (index in this.indices.reversed()) {\n val element = this[index]\n if (predicate(element))
return element\n }\n } throw NoSuchElementException("Array contains no element matching the
predicate.")\n}\n\n/**\n * Returns last index of [element], or -1 if the array does not contain element.\n */\npublic
fun <@kotlin.internal.OnlyInputTypes T> Array<out T>.lastIndexOf(element: T): Int {\n if (element == null) {\n
for (index in indices.reversed()) {\n if (this[index] == null) {\n return index\n }\n }\n } else {\n
for (index in indices.reversed()) {\n if (element == this[index]) {\n return index\n }\n }\n }\n } return -1\n}\n\n/**\n * Returns last index of [element], or -1 if the array does not contain
element.\n */\npublic fun ByteArray.lastIndexOf(element: Byte): Int {\n for (index in indices.reversed()) {\n if
(element == this[index]) {\n return index\n }\n }\n } return -1\n}\n\n/**\n * Returns last index of
[element], or -1 if the array does not contain element.\n */\npublic fun ShortArray.lastIndexOf(element: Short): Int
{\n for (index in indices.reversed()) {\n if (element == this[index]) {\n return index\n }\n }\n } return -1\n}\n\n/**\n * Returns last index of [element], or -1 if the array does not contain element.\n */\npublic fun
IntArray.lastIndexOf(element: Int): Int {\n for (index in indices.reversed()) {\n if (element == this[index]) {\n
return index\n }\n }\n } return -1\n}\n\n/**\n * Returns last index of [element], or -1 if the array does not
contain element.\n */\npublic fun LongArray.lastIndexOf(element: Long): Int {\n for (index in indices.reversed())
{\n if (element == this[index]) {\n return index\n }\n }\n } return -1\n}\n\n/**\n * Returns last
index of [element], or -1 if the array does not contain element.\n */\n@Deprecated("The function has unclear
behavior when searching for NaN or zero values and will be removed soon. Use 'indexOfLast { it == element }'
instead to continue using this behavior, or '.asList().lastIndexOf(element: T)' to get the same search behavior as in a
list.", ReplaceWith("indexOfLast { it == element }"))\n@DeprecatedSinceKotlin(warningSince = "1.4",
errorSince = "1.6")\npublic fun FloatArray.lastIndexOf(element: Float): Int {\n for (index in indices.reversed())
{\n if (element == this[index]) {\n return index\n }\n }\n } return -1\n}\n\n/**\n * Returns last
index of [element], or -1 if the array does not contain element.\n */\n@Deprecated("The function has unclear
behavior when searching for NaN or zero values and will be removed soon. Use 'indexOfLast { it == element }'
instead to continue using this behavior, or '.asList().lastIndexOf(element: T)' to get the same search behavior as in a
list.", ReplaceWith("indexOfLast { it == element }"))\n@DeprecatedSinceKotlin(warningSince = "1.4",
errorSince = "1.6")\npublic fun DoubleArray.lastIndexOf(element: Double): Int {\n for (index in
indices.reversed()) {\n if (element == this[index]) {\n return index\n }\n }\n } return -1\n}\n\n/**\n * Returns last index of [element], or -1 if the array does not contain element.\n */\npublic fun

```

```

BooleanArray.lastIndexOf(element: Boolean): Int {\n  for (index in indices.reversed()) {\n    if (element ==
this[index]) {\n      return index\n    }\n  }\n  return -1\n}\n\n/**\n * Returns last index of [element], or -1 if
the array does not contain element.\n */\npublic fun CharArray.lastIndexOf(element: Char): Int {\n  for (index in
indices.reversed()) {\n    if (element == this[index]) {\n      return index\n    }\n  }\n  return -1\n}\n\n/**\n * Returns the last element, or `null` if the array is empty.\n */\n @sample
samples.collections.Collections.Elements.last\n\n/\npublic fun <T> Array<out T>.lastOrNull(): T? {\n  return if
(isEmpty()) null else this[size - 1]\n}\n\n/**\n * Returns the last element, or `null` if the array is empty.\n */\n @sample
samples.collections.Collections.Elements.last\n\n/\npublic fun ByteArray.lastOrNull(): Byte? {\n  return
if (isEmpty()) null else this[size - 1]\n}\n\n/**\n * Returns the last element, or `null` if the array is empty.\n */\n @sample
samples.collections.Collections.Elements.last\n\n/\npublic fun ShortArray.lastOrNull(): Short? {\n  return
if (isEmpty()) null else this[size - 1]\n}\n\n/**\n * Returns the last element, or `null` if the array is empty.\n */\n @sample
samples.collections.Collections.Elements.last\n\n/\npublic fun IntArray.lastOrNull(): Int? {\n  return
if (isEmpty()) null else this[size - 1]\n}\n\n/**\n * Returns the last element, or `null` if the array is empty.\n */\n @sample
samples.collections.Collections.Elements.last\n\n/\npublic fun LongArray.lastOrNull(): Long? {\n  return
if (isEmpty()) null else this[size - 1]\n}\n\n/**\n * Returns the last element, or `null` if the array is empty.\n */\n @sample
samples.collections.Collections.Elements.last\n\n/\npublic fun FloatArray.lastOrNull(): Float? {\n  return
if (isEmpty()) null else this[size - 1]\n}\n\n/**\n * Returns the last element, or `null` if the array is empty.\n */\n @sample
samples.collections.Collections.Elements.last\n\n/\npublic fun DoubleArray.lastOrNull(): Double? {\n  return
if (isEmpty()) null else this[size - 1]\n}\n\n/**\n * Returns the last element, or `null` if the array is empty.\n */\n @sample
samples.collections.Collections.Elements.last\n\n/\npublic fun BooleanArray.lastOrNull(): Boolean? {\n  return
if (isEmpty()) null else this[size - 1]\n}\n\n/**\n * Returns the last element, or `null` if the array is
empty.\n */\n @sample samples.collections.Collections.Elements.last\n\n/\npublic fun CharArray.lastOrNull():
Char? {\n  return if (isEmpty()) null else this[size - 1]\n}\n\n/**\n * Returns the last element matching the given
[predicate], or `null` if no such element was found.\n */\n @sample
samples.collections.Collections.Elements.last\n\n/\npublic inline fun <T> Array<out T>.lastOrNull(predicate: (T) ->
Boolean): T? {\n  for (index in this.indices.reversed()) {\n    val element = this[index]\n    if
(predicate(element)) return element\n  }\n  return null\n}\n\n/**\n * Returns the last element matching the given
[predicate], or `null` if no such element was found.\n */\n @sample
samples.collections.Collections.Elements.last\n\n/\npublic inline fun ByteArray.lastOrNull(predicate: (Byte) ->
Boolean): Byte? {\n  for (index in this.indices.reversed()) {\n    val element = this[index]\n    if
(predicate(element)) return element\n  }\n  return null\n}\n\n/**\n * Returns the last element matching the given
[predicate], or `null` if no such element was found.\n */\n @sample
samples.collections.Collections.Elements.last\n\n/\npublic inline fun ShortArray.lastOrNull(predicate: (Short) ->
Boolean): Short? {\n  for (index in this.indices.reversed()) {\n    val element = this[index]\n    if
(predicate(element)) return element\n  }\n  return null\n}\n\n/**\n * Returns the last element matching the given
[predicate], or `null` if no such element was found.\n */\n @sample
samples.collections.Collections.Elements.last\n\n/\npublic inline fun IntArray.lastOrNull(predicate: (Int) ->
Boolean): Int? {\n  for (index in this.indices.reversed()) {\n    val element = this[index]\n    if
(predicate(element)) return element\n  }\n  return null\n}\n\n/**\n * Returns the last element matching the given
[predicate], or `null` if no such element was found.\n */\n @sample
samples.collections.Collections.Elements.last\n\n/\npublic inline fun LongArray.lastOrNull(predicate: (Long) ->
Boolean): Long? {\n  for (index in this.indices.reversed()) {\n    val element = this[index]\n    if
(predicate(element)) return element\n  }\n  return null\n}\n\n/**\n * Returns the last element matching the given
[predicate], or `null` if no such element was found.\n */\n @sample
samples.collections.Collections.Elements.last\n\n/\npublic inline fun FloatArray.lastOrNull(predicate: (Float) ->
Boolean): Float? {\n  for (index in this.indices.reversed()) {\n    val element = this[index]\n    if
(predicate(element)) return element\n  }\n  return null\n}\n\n/**\n * Returns the last element matching the given
[predicate], or `null` if no such element was found.\n */\n @sample

```

```

samples.collections.Collections.Elements.last\n *\npublic inline fun DoubleArray.lastOrNull(predicate: (Double) ->
Boolean): Double? {\n for (index in this.indices.reversed()) {\n val element = this[index]\n if
(predicate(element)) return element\n }\n return null\n}\n\n**\n * Returns the last element matching the given
[predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.last\n *\npublic inline fun BooleanArray.lastOrNull(predicate: (Boolean)
-> Boolean): Boolean? {\n for (index in this.indices.reversed()) {\n val element = this[index]\n if
(predicate(element)) return element\n }\n return null\n}\n\n**\n * Returns the last element matching the given
[predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.last\n *\npublic inline fun CharArray.lastOrNull(predicate: (Char) ->
Boolean): Char? {\n for (index in this.indices.reversed()) {\n val element = this[index]\n if
(predicate(element)) return element\n }\n return null\n}\n\n**\n * Returns a random element from this array.\n
* \n * @throws NoSuchElementException if this array is empty.\n
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.random(): T {\n
return random(Random)\n}\n\n**\n * Returns a random element from this array.\n * \n * @throws
NoSuchElementException if this array is empty.\n *\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic
inline fun ByteArray.random(): Byte {\n return random(Random)\n}\n\n**\n * Returns a random element from
this array.\n * \n * @throws NoSuchElementException if this array is empty.\n
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.random(): Short {\n return
random(Random)\n}\n\n**\n * Returns a random element from this array.\n * \n * @throws
NoSuchElementException if this array is empty.\n *\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic
inline fun IntArray.random(): Int {\n return random(Random)\n}\n\n**\n * Returns a random element from this
array.\n * \n * @throws NoSuchElementException if this array is empty.\n
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.random(): Long {\n return
random(Random)\n}\n\n**\n * Returns a random element from this array.\n * \n * @throws
NoSuchElementException if this array is empty.\n *\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic
inline fun FloatArray.random(): Float {\n return random(Random)\n}\n\n**\n * Returns a random element from
this array.\n * \n * @throws NoSuchElementException if this array is empty.\n
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.random(): Double {\n
return random(Random)\n}\n\n**\n * Returns a random element from this array.\n * \n * @throws
NoSuchElementException if this array is empty.\n *\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic
inline fun BooleanArray.random(): Boolean {\n return random(Random)\n}\n\n**\n * Returns a random element
from this array.\n * \n * @throws NoSuchElementException if this array is empty.\n
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.random(): Char {\n return
random(Random)\n}\n\n**\n * Returns a random element from this array using the specified source of
randomness.\n * \n * @throws NoSuchElementException if this array is empty.\n *\n@SinceKotlin("1.3")\npublic
fun <T> Array<out T>.random(random: Random): T {\n if (isEmpty())\n throw
NoSuchElementException("Array is empty.")\n return get(random.nextInt(size))\n}\n\n**\n * Returns a random
element from this array using the specified source of randomness.\n * \n * @throws NoSuchElementException if
this array is empty.\n *\n@SinceKotlin("1.3")\npublic fun ByteArray.random(random: Random): Byte {\n if
(isEmpty())\n throw NoSuchElementException("Array is empty.")\n return
get(random.nextInt(size))\n}\n\n**\n * Returns a random element from this array using the specified source of
randomness.\n * \n * @throws NoSuchElementException if this array is empty.\n *\n@SinceKotlin("1.3")\npublic
fun ShortArray.random(random: Random): Short {\n if (isEmpty())\n throw
NoSuchElementException("Array is empty.")\n return get(random.nextInt(size))\n}\n\n**\n * Returns a random
element from this array using the specified source of randomness.\n * \n * @throws NoSuchElementException if
this array is empty.\n *\n@SinceKotlin("1.3")\npublic fun IntArray.random(random: Random): Int {\n if
(isEmpty())\n throw NoSuchElementException("Array is empty.")\n return
get(random.nextInt(size))\n}\n\n**\n * Returns a random element from this array using the specified source of

```

```

randomness.\n * \n * @throws NoSuchElementException if this array is empty.\n */\n@SinceKotlin("1.3")\npublic
fun LongArray.random(random: Random): Long {\n    if (isEmpty())\n        throw
NoSuchElementException("Array is empty.")\n    return get(random.nextInt(size))\n}\n\n/**\n * Returns a random
element from this array using the specified source of randomness.\n * \n * @throws NoSuchElementException if
this array is empty.\n */\n@SinceKotlin("1.3")\npublic fun FloatArray.random(random: Random): Float {\n    if
(isEmpty())\n        throw NoSuchElementException("Array is empty.")\n    return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness.\n * \n * @throws NoSuchElementException if this array is empty.\n */\n@SinceKotlin("1.3")\npublic
fun DoubleArray.random(random: Random): Double {\n    if (isEmpty())\n        throw
NoSuchElementException("Array is empty.")\n    return get(random.nextInt(size))\n}\n\n/**\n * Returns a random
element from this array using the specified source of randomness.\n * \n * @throws NoSuchElementException if
this array is empty.\n */\n@SinceKotlin("1.3")\npublic fun BooleanArray.random(random: Random): Boolean {\n
    if (isEmpty())\n        throw NoSuchElementException("Array is empty.")\n    return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness.\n * \n * @throws NoSuchElementException if this array is empty.\n */\n@SinceKotlin("1.3")\npublic
fun CharArray.random(random: Random): Char {\n    if (isEmpty())\n        throw NoSuchElementException("Array
is empty.")\n    return get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array, or `null` if
this array is empty.\n */\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npubli
c inline fun <T> Array<out T>.randomOrNull(): T? {\n    return randomOrNull(Random)\n}\n\n/**\n * Returns a
random element from this array, or `null` if this array is empty.\n */\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npubli
c inline fun ByteArray.randomOrNull(): Byte? {\n    return randomOrNull(Random)\n}\n\n/**\n * Returns a random
element from this array, or `null` if this array is empty.\n */\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npubli
c inline fun ShortArray.randomOrNull(): Short? {\n    return randomOrNull(Random)\n}\n\n/**\n * Returns a
random element from this array, or `null` if this array is empty.\n */\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npubli
c inline fun IntArray.randomOrNull(): Int? {\n    return randomOrNull(Random)\n}\n\n/**\n * Returns a random
element from this array, or `null` if this array is empty.\n */\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npubli
c inline fun LongArray.randomOrNull(): Long? {\n    return randomOrNull(Random)\n}\n\n/**\n * Returns a
random element from this array, or `null` if this array is empty.\n */\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npubli
c inline fun FloatArray.randomOrNull(): Float? {\n    return randomOrNull(Random)\n}\n\n/**\n * Returns a
random element from this array, or `null` if this array is empty.\n */\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npubli
c inline fun DoubleArray.randomOrNull(): Double? {\n    return randomOrNull(Random)\n}\n\n/**\n * Returns a
random element from this array, or `null` if this array is empty.\n */\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npubli
c inline fun BooleanArray.randomOrNull(): Boolean? {\n    return randomOrNull(Random)\n}\n\n/**\n * Returns a
random element from this array, or `null` if this array is empty.\n */\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npubli
c inline fun CharArray.randomOrNull(): Char? {\n    return randomOrNull(Random)\n}\n\n/**\n * Returns a
random element from this array using the specified source of randomness, or `null` if this array is empty.\n */\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun <T> Array<out
T>.randomOrNull(random: Random): T? {\n    if (isEmpty())\n        return null\n    return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of

```

randomness, or `null` if this array is empty.

```

*\/n@SinceKotlin("1.4")n@WasExperimental(ExperimentalStdlibApi::class)npublic fun
ByteArray.randomOrNull(random: Random): Byte? {n if (isEmpty())n return nulln return
get(random.nextInt(size))n}n/n/**n * Returns a random element from this array using the specified source of
randomness, or `null` if this array is empty.n
*\/n@SinceKotlin("1.4")n@WasExperimental(ExperimentalStdlibApi::class)npublic fun
ShortArray.randomOrNull(random: Random): Short? {n if (isEmpty())n return nulln return
get(random.nextInt(size))n}n/n/**n * Returns a random element from this array using the specified source of
randomness, or `null` if this array is empty.n
*\/n@SinceKotlin("1.4")n@WasExperimental(ExperimentalStdlibApi::class)npublic fun
IntArray.randomOrNull(random: Random): Int? {n if (isEmpty())n return nulln return
get(random.nextInt(size))n}n/n/**n * Returns a random element from this array using the specified source of
randomness, or `null` if this array is empty.n
*\/n@SinceKotlin("1.4")n@WasExperimental(ExperimentalStdlibApi::class)npublic fun
LongArray.randomOrNull(random: Random): Long? {n if (isEmpty())n return nulln return
get(random.nextInt(size))n}n/n/**n * Returns a random element from this array using the specified source of
randomness, or `null` if this array is empty.n
*\/n@SinceKotlin("1.4")n@WasExperimental(ExperimentalStdlibApi::class)npublic fun
FloatArray.randomOrNull(random: Random): Float? {n if (isEmpty())n return nulln return
get(random.nextInt(size))n}n/n/**n * Returns a random element from this array using the specified source of
randomness, or `null` if this array is empty.n
*\/n@SinceKotlin("1.4")n@WasExperimental(ExperimentalStdlibApi::class)npublic fun
DoubleArray.randomOrNull(random: Random): Double? {n if (isEmpty())n return nulln return
get(random.nextInt(size))n}n/n/**n * Returns a random element from this array using the specified source of
randomness, or `null` if this array is empty.n
*\/n@SinceKotlin("1.4")n@WasExperimental(ExperimentalStdlibApi::class)npublic fun
BooleanArray.randomOrNull(random: Random): Boolean? {n if (isEmpty())n return nulln return
get(random.nextInt(size))n}n/n/**n * Returns a random element from this array using the specified source of
randomness, or `null` if this array is empty.n
*\/n@SinceKotlin("1.4")n@WasExperimental(ExperimentalStdlibApi::class)npublic fun
CharArray.randomOrNull(random: Random): Char? {n if (isEmpty())n return nulln return
get(random.nextInt(size))n}n/n/**n * Returns the single element, or throws an exception if the array is empty or
has more than one element.n *\/npublic fun <T> Array<out T>.single(): T {n return when (size) {n 0 ->
throw NoSuchElementException("Array is empty.")n 1 -> this[0]n else -> throw
IllegalArgumentException("Array has more than one element.")n }}n/n/n/**n * Returns the single element, or
throws an exception if the array is empty or has more than one element.n *\/npublic fun ByteArray.single(): Byte
{n return when (size) {n 0 -> throw NoSuchElementException("Array is empty.")n 1 -> this[0]n
else -> throw IllegalArgumentException("Array has more than one element.")n }}n/n/n/**n * Returns the
single element, or throws an exception if the array is empty or has more than one element.n *\/npublic fun
ShortArray.single(): Short {n return when (size) {n 0 -> throw NoSuchElementException("Array is
empty.")n 1 -> this[0]n else -> throw IllegalArgumentException("Array has more than one element.")n
}}n/n/n/**n * Returns the single element, or throws an exception if the array is empty or has more than one
element.n *\/npublic fun IntArray.single(): Int {n return when (size) {n 0 -> throw
NoSuchElementException("Array is empty.")n 1 -> this[0]n else -> throw
IllegalArgumentException("Array has more than one element.")n }}n/n/n/**n * Returns the single element, or
throws an exception if the array is empty or has more than one element.n *\/npublic fun LongArray.single(): Long
{n return when (size) {n 0 -> throw NoSuchElementException("Array is empty.")n 1 -> this[0]n
else -> throw IllegalArgumentException("Array has more than one element.")n }}n/n/n/**n * Returns the

```



```

the predicate.}\n @Suppress(\"UNCHECKED_CAST\")\n return single as Float}\n}\n/**\n * Returns the
single element matching the given [predicate], or throws exception if there is no or more than one matching
element.\n */\npublic inline fun DoubleArray.single(predicate: (Double) -> Boolean): Double {\n var single:
Double? = null\n var found = false\n for (element in this) {\n if (predicate(element)) {\n if (found)
throw IllegalArgumentException(\"Array contains more than one matching element.}\n single = element\n
found = true\n }\n }\n if (!found) throw NoSuchElementException(\"Array contains no element
matching the predicate.}\n @Suppress(\"UNCHECKED_CAST\")\n return single as Double}\n}\n/**\n *
Returns the single element matching the given [predicate], or throws exception if there is no or more than one
matching element.\n */\npublic inline fun BooleanArray.single(predicate: (Boolean) -> Boolean): Boolean {\n var
single: Boolean? = null\n var found = false\n for (element in this) {\n if (predicate(element)) {\n if
(found) throw IllegalArgumentException(\"Array contains more than one matching element.}\n single =
element\n found = true\n }\n }\n if (!found) throw NoSuchElementException(\"Array contains no
element matching the predicate.}\n @Suppress(\"UNCHECKED_CAST\")\n return single as
Boolean}\n}\n/**\n * Returns the single element matching the given [predicate], or throws exception if there is no
or more than one matching element.\n */\npublic inline fun CharArray.single(predicate: (Char) -> Boolean): Char
{\n var single: Char? = null\n var found = false\n for (element in this) {\n if (predicate(element)) {\n
if (found) throw IllegalArgumentException(\"Array contains more than one matching element.}\n single =
element\n found = true\n }\n }\n if (!found) throw NoSuchElementException(\"Array contains no
element matching the predicate.}\n @Suppress(\"UNCHECKED_CAST\")\n return single as Char}\n}\n/**\n *
Returns single element, or `null` if the array is empty or has more than one element.\n */\npublic fun <T>
Array<out T>.singleOrNull(): T? {\n return if (size == 1) this[0] else null}\n}\n/**\n * Returns single element, or
`null` if the array is empty or has more than one element.\n */\npublic fun ByteArray.singleOrNull(): Byte? {\n
return if (size == 1) this[0] else null}\n}\n/**\n * Returns single element, or `null` if the array is empty or has more
than one element.\n */\npublic fun ShortArray.singleOrNull(): Short? {\n return if (size == 1) this[0] else
null}\n}\n/**\n * Returns single element, or `null` if the array is empty or has more than one element.\n */\npublic
fun IntArray.singleOrNull(): Int? {\n return if (size == 1) this[0] else null}\n}\n/**\n * Returns single element, or
`null` if the array is empty or has more than one element.\n */\npublic fun LongArray.singleOrNull(): Long? {\n
return if (size == 1) this[0] else null}\n}\n/**\n * Returns single element, or `null` if the array is empty or has more
than one element.\n */\npublic fun FloatArray.singleOrNull(): Float? {\n return if (size == 1) this[0] else
null}\n}\n/**\n * Returns single element, or `null` if the array is empty or has more than one element.\n */\npublic
fun DoubleArray.singleOrNull(): Double? {\n return if (size == 1) this[0] else null}\n}\n/**\n * Returns single
element, or `null` if the array is empty or has more than one element.\n */\npublic fun BooleanArray.singleOrNull():
Boolean? {\n return if (size == 1) this[0] else null}\n}\n/**\n * Returns single element, or `null` if the array is
empty or has more than one element.\n */\npublic fun CharArray.singleOrNull(): Char? {\n return if (size == 1)
this[0] else null}\n}\n/**\n * Returns the single element matching the given [predicate], or `null` if element was not
found or more than one element was found.\n */\npublic inline fun <T> Array<out T>.singleOrNull(predicate: (T) -
> Boolean): T? {\n var single: T? = null\n var found = false\n for (element in this) {\n if
(predicate(element)) {\n if (found) return null\n single = element\n found = true\n }\n }\n if (!found)
return null\n return single}\n}\n/**\n * Returns the single element matching the given [predicate], or
`null` if element was not found or more than one element was found.\n */\npublic inline fun
ByteArray.singleOrNull(predicate: (Byte) -> Boolean): Byte? {\n var single: Byte? = null\n var found = false\n
for (element in this) {\n if (predicate(element)) {\n if (found) return null\n single = element\n
found = true\n }\n }\n if (!found) return null\n return single}\n}\n/**\n * Returns the single element
matching the given [predicate], or `null` if element was not found or more than one element was found.\n */\npublic
inline fun ShortArray.singleOrNull(predicate: (Short) -> Boolean): Short? {\n var single: Short? = null\n var
found = false\n for (element in this) {\n if (predicate(element)) {\n if (found) return null\n single
= element\n found = true\n }\n }\n if (!found) return null\n return single}\n}\n/**\n * Returns the
single element matching the given [predicate], or `null` if element was not found or more than one element was

```

```

found.\n */\npublic inline fun IntArray.singleOrNull(predicate: (Int) -> Boolean): Int? {\n    var single: Int? = null\n    var found = false\n    for (element in this) {\n        if (predicate(element)) {\n            if (found) return null\n            single = element\n            found = true\n        }\n    }\n    if (!found) return null\n    return single\n}\n\n/**\n * Returns the single element matching the given [predicate], or `null` if element was not found or more than one element was found.\n */\npublic inline fun LongArray.singleOrNull(predicate: (Long) -> Boolean): Long? {\n    var single: Long? = null\n    var found = false\n    for (element in this) {\n        if (predicate(element)) {\n            if (found) return null\n            single = element\n            found = true\n        }\n    }\n    if (!found) return null\n    return single\n}\n\n/**\n * Returns the single element matching the given [predicate], or `null` if element was not found or more than one element was found.\n */\npublic inline fun FloatArray.singleOrNull(predicate: (Float) -> Boolean): Float? {\n    var single: Float? = null\n    var found = false\n    for (element in this) {\n        if (predicate(element)) {\n            if (found) return null\n            single = element\n            found = true\n        }\n    }\n    if (!found) return null\n    return single\n}\n\n/**\n * Returns the single element matching the given [predicate], or `null` if element was not found or more than one element was found.\n */\npublic inline fun DoubleArray.singleOrNull(predicate: (Double) -> Boolean): Double? {\n    var single: Double? = null\n    var found = false\n    for (element in this) {\n        if (predicate(element)) {\n            if (found) return null\n            single = element\n            found = true\n        }\n    }\n    if (!found) return null\n    return single\n}\n\n/**\n * Returns the single element matching the given [predicate], or `null` if element was not found or more than one element was found.\n */\npublic inline fun BooleanArray.singleOrNull(predicate: (Boolean) -> Boolean): Boolean? {\n    var single: Boolean? = null\n    var found = false\n    for (element in this) {\n        if (predicate(element)) {\n            if (found) return null\n            single = element\n            found = true\n        }\n    }\n    if (!found) return null\n    return single\n}\n\n/**\n * Returns the single element matching the given [predicate], or `null` if element was not found or more than one element was found.\n */\npublic inline fun CharArray.singleOrNull(predicate: (Char) -> Boolean): Char? {\n    var single: Char? = null\n    var found = false\n    for (element in this) {\n        if (predicate(element)) {\n            if (found) return null\n            single = element\n            found = true\n        }\n    }\n    if (!found) return null\n    return single\n}\n\n/**\n * Returns a list containing all elements except first [n] elements.\n * \n * @throws\n * IllegalArgumentException if [n] is negative.\n * \n * @sample\n * samples.collections.Collections.Transformations.drop\n */\npublic fun <T> Array<out T>.drop(n: Int): List<T> {\n    require(n >= 0) { "Requested element count $n is less than zero." }\n    return takeLast((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except first [n] elements.\n * \n * @throws\n * IllegalArgumentException if [n] is negative.\n * \n * @sample\n * samples.collections.Collections.Transformations.drop\n */\npublic fun ByteArray.drop(n: Int): List<Byte> {\n    require(n >= 0) { "Requested element count $n is less than zero." }\n    return takeLast((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except first [n] elements.\n * \n * @throws\n * IllegalArgumentException if [n] is negative.\n * \n * @sample\n * samples.collections.Collections.Transformations.drop\n */\npublic fun ShortArray.drop(n: Int): List<Short> {\n    require(n >= 0) { "Requested element count $n is less than zero." }\n    return takeLast((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except first [n] elements.\n * \n * @throws\n * IllegalArgumentException if [n] is negative.\n * \n * @sample\n * samples.collections.Collections.Transformations.drop\n */\npublic fun IntArray.drop(n: Int): List<Int> {\n    require(n >= 0) { "Requested element count $n is less than zero." }\n    return takeLast((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except first [n] elements.\n * \n * @throws\n * IllegalArgumentException if [n] is negative.\n * \n * @sample\n * samples.collections.Collections.Transformations.drop\n */\npublic fun LongArray.drop(n: Int): List<Long> {\n    require(n >= 0) { "Requested element count $n is less than zero." }\n    return takeLast((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except first [n] elements.\n * \n * @throws\n * IllegalArgumentException if [n] is negative.\n * \n * @sample\n * samples.collections.Collections.Transformations.drop\n */\npublic fun FloatArray.drop(n: Int): List<Float> {\n    require(n >= 0) { "Requested element count $n is less than zero." }\n    return takeLast((size -

```

n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except first [n] elements.\n * \n * @throws
 IllegalArgumentException if [n] is negative.\n * \n * @sample
 samples.collections.Collections.Transformations.drop\n *\npublic fun DoubleArray.drop(n: Int): List<Double> {\n
 require(n >= 0) { \"Requested element count \$n is less than zero.\" }\n return takeLast((size -
 n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except first [n] elements.\n * \n * @throws
 IllegalArgumentException if [n] is negative.\n * \n * @sample
 samples.collections.Collections.Transformations.drop\n *\npublic fun BooleanArray.drop(n: Int): List<Boolean>
 {\n require(n >= 0) { \"Requested element count \$n is less than zero.\" }\n return takeLast((size -
 n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except first [n] elements.\n * \n * @throws
 IllegalArgumentException if [n] is negative.\n * \n * @sample
 samples.collections.Collections.Transformations.drop\n *\npublic fun CharArray.drop(n: Int): List<Char> {\n
 require(n >= 0) { \"Requested element count \$n is less than zero.\" }\n return takeLast((size -
 n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws
 IllegalArgumentException if [n] is negative.\n * \n * @sample
 samples.collections.Collections.Transformations.drop\n *\npublic fun <T> Array<out T>.dropLast(n: Int): List<T>
 {\n require(n >= 0) { \"Requested element count \$n is less than zero.\" }\n return take((size -
 n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws
 IllegalArgumentException if [n] is negative.\n * \n * @sample
 samples.collections.Collections.Transformations.drop\n *\npublic fun ByteArray.dropLast(n: Int): List<Byte> {\n
 require(n >= 0) { \"Requested element count \$n is less than zero.\" }\n return take((size -
 n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws
 IllegalArgumentException if [n] is negative.\n * \n * @sample
 samples.collections.Collections.Transformations.drop\n *\npublic fun ShortArray.dropLast(n: Int): List<Short> {\n
 require(n >= 0) { \"Requested element count \$n is less than zero.\" }\n return take((size -
 n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws
 IllegalArgumentException if [n] is negative.\n * \n * @sample
 samples.collections.Collections.Transformations.drop\n *\npublic fun IntArray.dropLast(n: Int): List<Int> {\n
 require(n >= 0) { \"Requested element count \$n is less than zero.\" }\n return take((size -
 n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws
 IllegalArgumentException if [n] is negative.\n * \n * @sample
 samples.collections.Collections.Transformations.drop\n *\npublic fun LongArray.dropLast(n: Int): List<Long> {\n
 require(n >= 0) { \"Requested element count \$n is less than zero.\" }\n return take((size -
 n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws
 IllegalArgumentException if [n] is negative.\n * \n * @sample
 samples.collections.Collections.Transformations.drop\n *\npublic fun FloatArray.dropLast(n: Int): List<Float> {\n
 require(n >= 0) { \"Requested element count \$n is less than zero.\" }\n return take((size -
 n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws
 IllegalArgumentException if [n] is negative.\n * \n * @sample
 samples.collections.Collections.Transformations.drop\n *\npublic fun DoubleArray.dropLast(n: Int): List<Double>
 {\n require(n >= 0) { \"Requested element count \$n is less than zero.\" }\n return take((size -
 n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws
 IllegalArgumentException if [n] is negative.\n * \n * @sample
 samples.collections.Collections.Transformations.drop\n *\npublic fun BooleanArray.dropLast(n: Int):
 List<Boolean> {\n require(n >= 0) { \"Requested element count \$n is less than zero.\" }\n return take((size -
 n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws
 IllegalArgumentException if [n] is negative.\n * \n * @sample
 samples.collections.Collections.Transformations.drop\n *\npublic fun CharArray.dropLast(n: Int): List<Char> {\n
 require(n >= 0) { \"Requested element count \$n is less than zero.\" }\n return take((size -

```

n).coerceAtLeast(0))\n\n/**\n * Returns a list containing all elements except last elements that satisfy the given
[predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n * \npublic inline fun <T>
Array<out T>.dropLastWhile(predicate: (T) -> Boolean): List<T> {\n    for (index in lastIndex downTo 0) {\n        if
(!predicate(this[index])) {\n            return take(index + 1)\n        }\n    }\n    return emptyList()\n}\n\n/**\n * Returns
a list containing all elements except last elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n * \npublic inline fun ByteArray.dropLastWhile(predicate:
(Byte) -> Boolean): List<Byte> {\n    for (index in lastIndex downTo 0) {\n        if (!predicate(this[index])) {\n
        return take(index + 1)\n        }\n    }\n    return emptyList()\n}\n\n/**\n * Returns a list containing all elements
except last elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n * \npublic inline fun ShortArray.dropLastWhile(predicate:
(Short) -> Boolean): List<Short> {\n    for (index in lastIndex downTo 0) {\n        if (!predicate(this[index])) {\n
        return take(index + 1)\n        }\n    }\n    return emptyList()\n}\n\n/**\n * Returns a list containing all elements
except last elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n * \npublic inline fun IntArray.dropLastWhile(predicate:
(Int) -> Boolean): List<Int> {\n    for (index in lastIndex downTo 0) {\n        if (!predicate(this[index])) {\n
        return take(index + 1)\n        }\n    }\n    return emptyList()\n}\n\n/**\n * Returns a list containing all elements
except last elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n * \npublic inline fun LongArray.dropLastWhile(predicate:
(Long) -> Boolean): List<Long> {\n    for (index in lastIndex downTo 0) {\n        if (!predicate(this[index])) {\n
        return take(index + 1)\n        }\n    }\n    return emptyList()\n}\n\n/**\n * Returns a list containing all elements
except last elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n * \npublic inline fun FloatArray.dropLastWhile(predicate:
(Float) -> Boolean): List<Float> {\n    for (index in lastIndex downTo 0) {\n        if (!predicate(this[index])) {\n
        return take(index + 1)\n        }\n    }\n    return emptyList()\n}\n\n/**\n * Returns a list containing all elements
except last elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n * \npublic inline fun DoubleArray.dropLastWhile(predicate:
(Double) -> Boolean): List<Double> {\n    for (index in lastIndex downTo 0) {\n        if (!predicate(this[index])) {\n
        return take(index + 1)\n        }\n    }\n    return emptyList()\n}\n\n/**\n * Returns a list containing all elements
except last elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n * \npublic inline fun
BooleanArray.dropLastWhile(predicate: (Boolean) -> Boolean): List<Boolean> {\n    for (index in lastIndex
downTo 0) {\n        if (!predicate(this[index])) {\n            return take(index + 1)\n        }\n    }\n    return
emptyList()\n}\n\n/**\n * Returns a list containing all elements except last elements that satisfy the given
[predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n * \npublic inline fun
CharArray.dropLastWhile(predicate: (Char) -> Boolean): List<Char> {\n    for (index in lastIndex downTo 0) {\n
    if (!predicate(this[index])) {\n        return take(index + 1)\n    }\n}\n    return emptyList()\n}\n\n/**\n *
Returns a list containing all elements except first elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n * \npublic inline fun <T> Array<out
T>.dropWhile(predicate: (T) -> Boolean): List<T> {\n    var yielding = false\n    val list = ArrayList<T>()\n    for
(item in this)\n        if (yielding)\n            list.add(item)\n        else if (!predicate(item)) {\n            list.add(item)\n
            yielding = true\n        }\n    return list\n}\n\n/**\n * Returns a list containing all elements except first elements that
satisfy the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n * \npublic
inline fun ByteArray.dropWhile(predicate: (Byte) -> Boolean): List<Byte> {\n    var yielding = false\n    val list =
ArrayList<Byte>()\n    for (item in this)\n        if (yielding)\n            list.add(item)\n        else if (!predicate(item)) {\n
            list.add(item)\n            yielding = true\n        }\n    return list\n}\n\n/**\n * Returns a list containing all
elements except first elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n * \npublic inline fun ShortArray.dropWhile(predicate:
(Short) -> Boolean): List<Short> {\n    var yielding = false\n    val list = ArrayList<Short>()\n    for (item in this)\n

```

```

    if (yielding)\n        list.add(item)\n    else if (!predicate(item)) {\n        list.add(item)\n        yielding =
true\n    }\n    return list\n}\n\n/**\n * Returns a list containing all elements except first elements that satisfy the
given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic inline fun
IntArray.dropWhile(predicate: (Int) -> Boolean): List<Int> {\n    var yielding = false\n    val list =
ArrayList<Int>()\n    for (item in this)\n        if (yielding)\n            list.add(item)\n        else if (!predicate(item)) {\n
            list.add(item)\n            yielding = true\n        }\n    return list\n}\n\n/**\n * Returns a list containing all elements
except first elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n */\npublic inline fun LongArray.dropWhile(predicate:
(Long) -> Boolean): List<Long> {\n    var yielding = false\n    val list = ArrayList<Long>()\n    for (item in this)\n
        if (yielding)\n            list.add(item)\n        else if (!predicate(item)) {\n            list.add(item)\n
            yielding =
true\n        }\n    return list\n}\n\n/**\n * Returns a list containing all elements except first elements that satisfy the
given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic inline fun
FloatArray.dropWhile(predicate: (Float) -> Boolean): List<Float> {\n    var yielding = false\n    val list =
ArrayList<Float>()\n    for (item in this)\n        if (yielding)\n            list.add(item)\n        else if (!predicate(item))
{\n            list.add(item)\n            yielding = true\n        }\n    return list\n}\n\n/**\n * Returns a list containing all
elements except first elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n */\npublic inline fun DoubleArray.dropWhile(predicate:
(Double) -> Boolean): List<Double> {\n    var yielding = false\n    val list = ArrayList<Double>()\n    for (item in
this)\n        if (yielding)\n            list.add(item)\n        else if (!predicate(item)) {\n            list.add(item)\n
            yielding =
true\n        }\n    return list\n}\n\n/**\n * Returns a list containing all elements except first elements that
satisfy the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic
inline fun BooleanArray.dropWhile(predicate: (Boolean) -> Boolean): List<Boolean> {\n    var yielding = false\n
    val list = ArrayList<Boolean>()\n    for (item in this)\n        if (yielding)\n            list.add(item)\n        else if
(!predicate(item)) {\n            list.add(item)\n            yielding = true\n        }\n    return list\n}\n\n/**\n * Returns a list
containing all elements except first elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n */\npublic inline fun CharArray.dropWhile(predicate:
(Char) -> Boolean): List<Char> {\n    var yielding = false\n    val list = ArrayList<Char>()\n    for (item in this)\n
        if (yielding)\n            list.add(item)\n        else if (!predicate(item)) {\n            list.add(item)\n
            yielding =
true\n        }\n    return list\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * \n *
@sample samples.collections.Collections.Filtering.filter\n */\npublic inline fun <T> Array<out
T>.filter(predicate: (T) -> Boolean): List<T> {\n    return filterTo(ArrayList<T>(), predicate)\n}\n\n/**\n * Returns
a list containing only elements matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n */\npublic inline fun ByteArray.filter(predicate: (Byte) -> Boolean):
List<Byte> {\n    return filterTo(ArrayList<Byte>(), predicate)\n}\n\n/**\n * Returns a list containing only elements
matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n */\npublic inline
fun ShortArray.filter(predicate: (Short) -> Boolean): List<Short> {\n    return filterTo(ArrayList<Short>(),
predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n */\npublic inline fun IntArray.filter(predicate: (Int) -> Boolean):
List<Int> {\n    return filterTo(ArrayList<Int>(), predicate)\n}\n\n/**\n * Returns a list containing only elements
matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n */\npublic inline
fun LongArray.filter(predicate: (Long) -> Boolean): List<Long> {\n    return filterTo(ArrayList<Long>(),
predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n */\npublic inline fun FloatArray.filter(predicate: (Float) ->
Boolean): List<Float> {\n    return filterTo(ArrayList<Float>(), predicate)\n}\n\n/**\n * Returns a list containing
only elements matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n
*/\npublic inline fun DoubleArray.filter(predicate: (Double) -> Boolean): List<Double> {\n    return
filterTo(ArrayList<Double>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given
[predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n */\npublic inline fun

```

```

BooleanArray.filter(predicate: (Boolean) -> Boolean): List<Boolean> {\n  return filterTo(ArrayList<Boolean>(),
predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n */\npublic inline fun CharArray.filter(predicate: (Char) ->
Boolean): List<Char> {\n  return filterTo(ArrayList<Char>(), predicate)\n}\n\n/**\n * Returns a list containing
only elements matching the given [predicate].\n * @param [predicate] function that takes the index of an element
and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexed\n */\npublic inline fun <T> Array<out
T>.filterIndexed(predicate: (index: Int, T) -> Boolean): List<T> {\n  return filterIndexedTo(ArrayList<T>(),
predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * @param
[predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate
evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexed\n */\npublic
inline fun ByteArray.filterIndexed(predicate: (index: Int, Byte) -> Boolean): List<Byte> {\n  return
filterIndexedTo(ArrayList<Byte>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the
given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself\n * and
returns the result of predicate evaluation on the element.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexed\n */\npublic inline fun ShortArray.filterIndexed(predicate:
(index: Int, Short) -> Boolean): List<Short> {\n  return filterIndexedTo(ArrayList<Short>(), predicate)\n}\n\n/**\n
* Returns a list containing only elements matching the given [predicate].\n * @param [predicate] function that takes
the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n
* @sample samples.collections.Collections.Filtering.filterIndexed\n */\npublic inline fun
IntArray.filterIndexed(predicate: (index: Int, Int) -> Boolean): List<Int> {\n  return
filterIndexedTo(ArrayList<Int>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the
given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself\n * and
returns the result of predicate evaluation on the element.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexed\n */\npublic inline fun LongArray.filterIndexed(predicate:
(index: Int, Long) -> Boolean): List<Long> {\n  return filterIndexedTo(ArrayList<Long>(), predicate)\n}\n\n/**\n
* Returns a list containing only elements matching the given [predicate].\n * @param [predicate] function that takes
the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n
* @sample samples.collections.Collections.Filtering.filterIndexed\n */\npublic inline fun
FloatArray.filterIndexed(predicate: (index: Int, Float) -> Boolean): List<Float> {\n  return
filterIndexedTo(ArrayList<Float>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the
given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself\n * and
returns the result of predicate evaluation on the element.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexed\n */\npublic inline fun DoubleArray.filterIndexed(predicate:
(index: Int, Double) -> Boolean): List<Double> {\n  return filterIndexedTo(ArrayList<Double>(),
predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * @param
[predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate
evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexed\n */\npublic
inline fun BooleanArray.filterIndexed(predicate: (index: Int, Boolean) -> Boolean): List<Boolean> {\n  return
filterIndexedTo(ArrayList<Boolean>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching
the given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself\n *
and returns the result of predicate evaluation on the element.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexed\n */\npublic inline fun CharArray.filterIndexed(predicate:
(index: Int, Char) -> Boolean): List<Char> {\n  return filterIndexedTo(ArrayList<Char>(), predicate)\n}\n\n/**\n
* Appends all elements matching the given [predicate] to the given [destination].\n * @param [predicate] function that
takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the
element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexedTo\n */\npublic inline fun <T, C :
MutableCollection<in T>> Array<out T>.filterIndexedTo(destination: C, predicate: (index: Int, T) -> Boolean): C

```

```

{\n  forEachIndexed { index, element ->\n    if (predicate(index, element)) destination.add(element)\n  }\n  return destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexedTo\n */\npublic inline fun <C : MutableCollection<in Byte>> ByteArray.filterIndexedTo(destination: C, predicate: (index: Int, Byte) -> Boolean): C {\n  forEachIndexed { index, element ->\n    if (predicate(index, element)) destination.add(element)\n  }\n  return destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexedTo\n */\npublic inline fun <C : MutableCollection<in Short>> ShortArray.filterIndexedTo(destination: C, predicate: (index: Int, Short) -> Boolean): C {\n  forEachIndexed { index, element ->\n    if (predicate(index, element)) destination.add(element)\n  }\n  return destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexedTo\n */\npublic inline fun <C : MutableCollection<in Int>> IntArray.filterIndexedTo(destination: C, predicate: (index: Int, Int) -> Boolean): C {\n  forEachIndexed { index, element ->\n    if (predicate(index, element)) destination.add(element)\n  }\n  return destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexedTo\n */\npublic inline fun <C : MutableCollection<in Long>> LongArray.filterIndexedTo(destination: C, predicate: (index: Int, Long) -> Boolean): C {\n  forEachIndexed { index, element ->\n    if (predicate(index, element)) destination.add(element)\n  }\n  return destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexedTo\n */\npublic inline fun <C : MutableCollection<in Float>> FloatArray.filterIndexedTo(destination: C, predicate: (index: Int, Float) -> Boolean): C {\n  forEachIndexed { index, element ->\n    if (predicate(index, element)) destination.add(element)\n  }\n  return destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexedTo\n */\npublic inline fun <C : MutableCollection<in Double>> DoubleArray.filterIndexedTo(destination: C, predicate: (index: Int, Double) -> Boolean): C {\n  forEachIndexed { index, element ->\n    if (predicate(index, element)) destination.add(element)\n  }\n  return destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexedTo\n */\npublic inline fun <C : MutableCollection<in Boolean>> BooleanArray.filterIndexedTo(destination: C, predicate: (index: Int, Boolean) -> Boolean): C {\n  forEachIndexed { index, element ->\n    if (predicate(index, element)) destination.add(element)\n  }\n  return destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexedTo\n */\npublic inline fun <C : MutableCollection<in Char>> CharArray.filterIndexedTo(destination: C, predicate: (index: Int, Char) -> Boolean): C {\n  forEachIndexed { index, element ->\n    if (predicate(index, element)) destination.add(element)\n  }\n  return destination\n}\n\n/**\n * Returns a list containing all elements that are instances of specified type parameter R.\n * \n * @sample samples.collections.Collections.Filtering.filterIsInstance\n */\npublic inline fun <reified R> Array<*>.filterIsInstance(): List<@kotlin.internal.NoInfer R> {\n  return

```

```

filterIsInstanceTo(ArrayList<R>())\n\n/**\n * Appends all elements that are instances of specified type
parameter R to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterIsInstanceTo\n *\npublic inline fun <reified R, C :
MutableCollection<in R>> Array<*>.filterIsInstanceTo(destination: C): C {\n  for (element in this) if (element is
R) destination.add(element)\n  return destination\n}\n\n/**\n * Returns a list containing all elements not matching
the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n *\npublic inline fun <T>
Array<out T>.filterNot(predicate: (T) -> Boolean): List<T> {\n  return filterNotTo(ArrayList<T>(),
predicate)\n}\n\n/**\n * Returns a list containing all elements not matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n *\npublic inline fun ByteArray.filterNot(predicate: (Byte) ->
Boolean): List<Byte> {\n  return filterNotTo(ArrayList<Byte>(), predicate)\n}\n\n/**\n * Returns a list containing
all elements not matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n
*\npublic inline fun ShortArray.filterNot(predicate: (Short) -> Boolean): List<Short> {\n  return
filterNotTo(ArrayList<Short>(), predicate)\n}\n\n/**\n * Returns a list containing all elements not matching the
given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n *\npublic inline fun
IntArray.filterNot(predicate: (Int) -> Boolean): List<Int> {\n  return filterNotTo(ArrayList<Int>(),
predicate)\n}\n\n/**\n * Returns a list containing all elements not matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n *\npublic inline fun LongArray.filterNot(predicate: (Long) ->
Boolean): List<Long> {\n  return filterNotTo(ArrayList<Long>(), predicate)\n}\n\n/**\n * Returns a list
containing all elements not matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n *\npublic inline fun FloatArray.filterNot(predicate: (Float) ->
Boolean): List<Float> {\n  return filterNotTo(ArrayList<Float>(), predicate)\n}\n\n/**\n * Returns a list
containing all elements not matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n *\npublic inline fun DoubleArray.filterNot(predicate: (Double) ->
Boolean): List<Double> {\n  return filterNotTo(ArrayList<Double>(), predicate)\n}\n\n/**\n * Returns a list
containing all elements not matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n *\npublic inline fun BooleanArray.filterNot(predicate: (Boolean) -
> Boolean): List<Boolean> {\n  return filterNotTo(ArrayList<Boolean>(), predicate)\n}\n\n/**\n * Returns a list
containing all elements not matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n *\npublic inline fun CharArray.filterNot(predicate: (Char) ->
Boolean): List<Char> {\n  return filterNotTo(ArrayList<Char>(), predicate)\n}\n\n/**\n * Returns a list containing
all elements that are not `null`.\n * \n * @sample samples.collections.Collections.Filtering.filterNotNull\n *\npublic
fun <T : Any> Array<out T?>.filterNotNull(): List<T> {\n  return filterNotNullTo(ArrayList<T>())\n}\n\n/**\n *
Appends all elements that are not `null` to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterNotNullTo\n *\npublic fun <C : MutableCollection<in T>, T : Any>
Array<out T?>.filterNotNullTo(destination: C): C {\n  for (element in this) if (element != null)
destination.add(element)\n  return destination\n}\n\n/**\n * Appends all elements not matching the given
[predicate] to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n
*\npublic inline fun <T, C : MutableCollection<in T>> Array<out T>.filterNotTo(destination: C, predicate: (T) ->
Boolean): C {\n  for (element in this) if (!predicate(element)) destination.add(element)\n  return
destination\n}\n\n/**\n * Appends all elements not matching the given [predicate] to the given [destination].\n * \n *
@sample samples.collections.Collections.Filtering.filterTo\n *\npublic inline fun <C : MutableCollection<in
Byte>> ByteArray.filterNotTo(destination: C, predicate: (Byte) -> Boolean): C {\n  for (element in this) if
(!predicate(element)) destination.add(element)\n  return destination\n}\n\n/**\n * Appends all elements not
matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n *\npublic inline fun <C : MutableCollection<in Short>>
ShortArray.filterNotTo(destination: C, predicate: (Short) -> Boolean): C {\n  for (element in this) if
(!predicate(element)) destination.add(element)\n  return destination\n}\n\n/**\n * Appends all elements not
matching the given [predicate] to the given [destination].\n * \n * @sample

```

```

samples.collections.Collections.Filtering.filterTo\n *^\npublic inline fun <C : MutableCollection<in Int>>
IntArray.filterNotTo(destination: C, predicate: (Int) -> Boolean): C {\n  for (element in this) if
(!predicate(element)) destination.add(element)\n  return destination\n}\n\n/**\n * Appends all elements not
matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n *^\npublic inline fun <C : MutableCollection<in Long>>
LongArray.filterNotTo(destination: C, predicate: (Long) -> Boolean): C {\n  for (element in this) if
(!predicate(element)) destination.add(element)\n  return destination\n}\n\n/**\n * Appends all elements not
matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n *^\npublic inline fun <C : MutableCollection<in Float>>
FloatArray.filterNotTo(destination: C, predicate: (Float) -> Boolean): C {\n  for (element in this) if
(!predicate(element)) destination.add(element)\n  return destination\n}\n\n/**\n * Appends all elements not
matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n *^\npublic inline fun <C : MutableCollection<in Double>>
DoubleArray.filterNotTo(destination: C, predicate: (Double) -> Boolean): C {\n  for (element in this) if
(!predicate(element)) destination.add(element)\n  return destination\n}\n\n/**\n * Appends all elements not
matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n *^\npublic inline fun <C : MutableCollection<in Boolean>>
BooleanArray.filterNotTo(destination: C, predicate: (Boolean) -> Boolean): C {\n  for (element in this) if
(!predicate(element)) destination.add(element)\n  return destination\n}\n\n/**\n * Appends all elements not
matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n *^\npublic inline fun <C : MutableCollection<in Char>>
CharArray.filterNotTo(destination: C, predicate: (Char) -> Boolean): C {\n  for (element in this) if
(!predicate(element)) destination.add(element)\n  return destination\n}\n\n/**\n * Appends all elements matching
the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n *^\npublic inline fun <T, C : MutableCollection<in T>> Array<out T>.filterTo(destination: C, predicate: (T) ->
Boolean): C {\n  for (element in this) if (predicate(element)) destination.add(element)\n  return
destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n * \n *
@sample
samples.collections.Collections.Filtering.filterTo\n *^\npublic inline fun <C : MutableCollection<in
Byte>> ByteArray.filterTo(destination: C, predicate: (Byte) -> Boolean): C {\n  for (element in this) if
(predicate(element)) destination.add(element)\n  return destination\n}\n\n/**\n * Appends all elements matching
the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n *^\npublic inline fun <C : MutableCollection<in Short>> ShortArray.filterTo(destination: C, predicate: (Short) ->
Boolean): C {\n  for (element in this) if (predicate(element)) destination.add(element)\n  return
destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n * \n *
@sample
samples.collections.Collections.Filtering.filterTo\n *^\npublic inline fun <C : MutableCollection<in Int>>
IntArray.filterTo(destination: C, predicate: (Int) -> Boolean): C {\n  for (element in this) if (predicate(element))
destination.add(element)\n  return destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to
the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n *^\npublic inline fun
<C : MutableCollection<in Long>> LongArray.filterTo(destination: C, predicate: (Long) -> Boolean): C {\n  for
(element in this) if (predicate(element)) destination.add(element)\n  return destination\n}\n\n/**\n * Appends all
elements matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n *^\npublic inline fun <C : MutableCollection<in Float>>
FloatArray.filterTo(destination: C, predicate: (Float) -> Boolean): C {\n  for (element in this) if
(predicate(element)) destination.add(element)\n  return destination\n}\n\n/**\n * Appends all elements matching
the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n *^\npublic inline fun <C : MutableCollection<in Double>>
DoubleArray.filterTo(destination: C, predicate: (Double)
-> Boolean): C {\n  for (element in this) if (predicate(element)) destination.add(element)\n  return
destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n * \n *

```

```

@sample samples.collections.Collections.Filtering.filterTo\n *^\npublic inline fun <C : MutableCollection<in
Boolean>> BooleanArray.filterTo(destination: C, predicate: (Boolean) -> Boolean): C {\n  for (element in this) if
(predicate(element)) destination.add(element)\n  return destination\n}\n\n/**\n * Appends all elements matching
the given [predicate] to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n
*^\npublic inline fun <C : MutableCollection<in Char>> CharArray.filterTo(destination: C, predicate: (Char) ->
Boolean): C {\n  for (element in this) if (predicate(element)) destination.add(element)\n  return
destination\n}\n\n/**\n * Returns a list containing elements at indices in the specified [indices] range.\n *^\npublic
fun <T> Array<out T>.slice(indices: IntRange): List<T> {\n  if (indices.isEmpty()) return listOf()\n  return
copyOfRange(indices.start, indices.endInclusive + 1).asList()\n}\n\n/**\n * Returns a list containing elements at
indices in the specified [indices] range.\n *^\npublic fun ByteArray.slice(indices: IntRange): List<Byte> {\n  if
(indices.isEmpty()) return listOf()\n  return copyOfRange(indices.start, indices.endInclusive +
1).asList()\n}\n\n/**\n * Returns a list containing elements at indices in the specified [indices] range.\n *^\npublic
fun ShortArray.slice(indices: IntRange): List<Short> {\n  if (indices.isEmpty()) return listOf()\n  return
copyOfRange(indices.start, indices.endInclusive + 1).asList()\n}\n\n/**\n * Returns a list containing elements at
indices in the specified [indices] range.\n *^\npublic fun IntArray.slice(indices: IntRange): List<Int> {\n  if
(indices.isEmpty()) return listOf()\n  return copyOfRange(indices.start, indices.endInclusive +
1).asList()\n}\n\n/**\n * Returns a list containing elements at indices in the specified [indices] range.\n *^\npublic
fun LongArray.slice(indices: IntRange): List<Long> {\n  if (indices.isEmpty()) return listOf()\n  return
copyOfRange(indices.start, indices.endInclusive + 1).asList()\n}\n\n/**\n * Returns a list containing elements at
indices in the specified [indices] range.\n *^\npublic fun FloatArray.slice(indices: IntRange): List<Float> {\n  if
(indices.isEmpty()) return listOf()\n  return copyOfRange(indices.start, indices.endInclusive +
1).asList()\n}\n\n/**\n * Returns a list containing elements at indices in the specified [indices] range.\n *^\npublic
fun DoubleArray.slice(indices: IntRange): List<Double> {\n  if (indices.isEmpty()) return listOf()\n  return
copyOfRange(indices.start, indices.endInclusive + 1).asList()\n}\n\n/**\n * Returns a list containing elements at
indices in the specified [indices] range.\n *^\npublic fun BooleanArray.slice(indices: IntRange): List<Boolean> {\n  if
(indices.isEmpty()) return listOf()\n  return copyOfRange(indices.start, indices.endInclusive +
1).asList()\n}\n\n/**\n * Returns a list containing elements at indices in the specified [indices] range.\n *^\npublic
fun CharArray.slice(indices: IntRange): List<Char> {\n  if (indices.isEmpty()) return listOf()\n  return
copyOfRange(indices.start, indices.endInclusive + 1).asList()\n}\n\n/**\n * Returns a list containing elements at
specified [indices].\n *^\npublic fun <T> Array<out T>.slice(indices: Iterable<Int>): List<T> {\n  val size =
indices.collectionSizeOrDefault(10)\n  if (size == 0) return emptyList()\n  val list = ArrayList<T>(size)\n  for
(index in indices) {\n    list.add(get(index))\n  }\n  return list\n}\n\n/**\n * Returns a list containing elements at
specified [indices].\n *^\npublic fun ByteArray.slice(indices: Iterable<Int>): List<Byte> {\n  val size =
indices.collectionSizeOrDefault(10)\n  if (size == 0) return emptyList()\n  val list = ArrayList<Byte>(size)\n  for
(index in indices) {\n    list.add(get(index))\n  }\n  return list\n}\n\n/**\n * Returns a list containing elements at
specified [indices].\n *^\npublic fun ShortArray.slice(indices: Iterable<Int>): List<Short> {\n  val size =
indices.collectionSizeOrDefault(10)\n  if (size == 0) return emptyList()\n  val list = ArrayList<Short>(size)\n
for (index in indices) {\n    list.add(get(index))\n  }\n  return list\n}\n\n/**\n * Returns a list containing
elements at specified [indices].\n *^\npublic fun IntArray.slice(indices: Iterable<Int>): List<Int> {\n  val size =
indices.collectionSizeOrDefault(10)\n  if (size == 0) return emptyList()\n  val list = ArrayList<Int>(size)\n
for (index in indices) {\n    list.add(get(index))\n  }\n  return list\n}\n\n/**\n * Returns a list containing
elements at specified [indices].\n *^\npublic fun LongArray.slice(indices: Iterable<Int>): List<Long> {\n  val size =
indices.collectionSizeOrDefault(10)\n  if (size == 0) return emptyList()\n  val list = ArrayList<Long>(size)\n
for (index in indices) {\n    list.add(get(index))\n  }\n  return list\n}\n\n/**\n * Returns a list containing
elements at specified [indices].\n *^\npublic fun FloatArray.slice(indices: Iterable<Int>): List<Float> {\n  val size =
indices.collectionSizeOrDefault(10)\n  if (size == 0) return emptyList()\n  val list = ArrayList<Float>(size)\n
for (index in indices) {\n    list.add(get(index))\n  }\n  return list\n}\n\n/**\n * Returns a list containing
elements at specified [indices].\n *^\npublic fun DoubleArray.slice(indices: Iterable<Int>): List<Double> {\n  val

```

```

size = indices.collectionSizeOrDefault(10)\n  if (size == 0) return emptyList()\n  val list =
ArrayList<Double>(size)\n  for (index in indices) {\n    list.add(get(index))\n  }\n  return list\n}\n\n/**\n *
Returns a list containing elements at specified [indices].\n */\npublic fun BooleanArray.slice(indices: Iterable<Int>):
List<Boolean> {\n  val size = indices.collectionSizeOrDefault(10)\n  if (size == 0) return emptyList()\n  val list
= ArrayList<Boolean>(size)\n  for (index in indices) {\n    list.add(get(index))\n  }\n  return list\n}\n\n/**\n *
Returns a list containing elements at specified [indices].\n */\npublic fun CharArray.slice(indices: Iterable<Int>):
List<Char> {\n  val size = indices.collectionSizeOrDefault(10)\n  if (size == 0) return emptyList()\n  val list =
ArrayList<Char>(size)\n  for (index in indices) {\n    list.add(get(index))\n  }\n  return list\n}\n\n/**\n *
Returns an array containing elements of this array at specified [indices].\n */\npublic fun <T>
Array<T>.sliceArray(indices: Collection<Int>): Array<T> {\n  val result = arrayOfNulls(this, indices.size)\n  var
targetIndex = 0\n  for (sourceIndex in indices) {\n    result[targetIndex++] = this[sourceIndex]\n  }\n  return
result\n}\n\n/**\n * Returns an array containing elements of this array at specified [indices].\n */\npublic fun
ByteArray.sliceArray(indices: Collection<Int>): ByteArray {\n  val result = ByteArray(indices.size)\n  var
targetIndex = 0\n  for (sourceIndex in indices) {\n    result[targetIndex++] = this[sourceIndex]\n  }\n  return
result\n}\n\n/**\n * Returns an array containing elements of this array at specified [indices].\n */\npublic fun
ShortArray.sliceArray(indices: Collection<Int>): ShortArray {\n  val result = ShortArray(indices.size)\n  var
targetIndex = 0\n  for (sourceIndex in indices) {\n    result[targetIndex++] = this[sourceIndex]\n  }\n  return
result\n}\n\n/**\n * Returns an array containing elements of this array at specified [indices].\n */\npublic fun
IntArray.sliceArray(indices: Collection<Int>): IntArray {\n  val result = IntArray(indices.size)\n  var targetIndex
= 0\n  for (sourceIndex in indices) {\n    result[targetIndex++] = this[sourceIndex]\n  }\n  return
result\n}\n\n/**\n * Returns an array containing elements of this array at specified [indices].\n */\npublic fun
LongArray.sliceArray(indices: Collection<Int>): LongArray {\n  val result = LongArray(indices.size)\n  var
targetIndex = 0\n  for (sourceIndex in indices) {\n    result[targetIndex++] = this[sourceIndex]\n  }\n  return
result\n}\n\n/**\n * Returns an array containing elements of this array at specified [indices].\n */\npublic fun
FloatArray.sliceArray(indices: Collection<Int>): FloatArray {\n  val result = FloatArray(indices.size)\n  var
targetIndex = 0\n  for (sourceIndex in indices) {\n    result[targetIndex++] = this[sourceIndex]\n  }\n  return
result\n}\n\n/**\n * Returns an array containing elements of this array at specified [indices].\n */\npublic fun
DoubleArray.sliceArray(indices: Collection<Int>): DoubleArray {\n  val result = DoubleArray(indices.size)\n
var targetIndex = 0\n  for (sourceIndex in indices) {\n    result[targetIndex++] = this[sourceIndex]\n  }\n
return result\n}\n\n/**\n * Returns an array containing elements of this array at specified [indices].\n */\npublic fun
BooleanArray.sliceArray(indices: Collection<Int>): BooleanArray {\n  val result = BooleanArray(indices.size)\n
var targetIndex = 0\n  for (sourceIndex in indices) {\n    result[targetIndex++] = this[sourceIndex]\n  }\n
return result\n}\n\n/**\n * Returns an array containing elements of this array at specified [indices].\n */\npublic fun
CharArray.sliceArray(indices: Collection<Int>): CharArray {\n  val result = CharArray(indices.size)\n  var
targetIndex = 0\n  for (sourceIndex in indices) {\n    result[targetIndex++] = this[sourceIndex]\n  }\n  return
result\n}\n\n/**\n * Returns an array containing elements at indices in the specified [indices] range.\n */\npublic fun
<T> Array<T>.sliceArray(indices: IntRange): Array<T> {\n  if (indices.isEmpty()) return copyOfRange(0, 0)\n
return copyOfRange(indices.start, indices.endInclusive + 1)\n}\n\n/**\n * Returns an array containing elements at
indices in the specified [indices] range.\n */\npublic fun ByteArray.sliceArray(indices: IntRange): ByteArray {\n  if
(indices.isEmpty()) return ByteArray(0)\n  return copyOfRange(indices.start, indices.endInclusive + 1)\n}\n\n/**\n *
Returns an array containing elements at indices in the specified [indices] range.\n */\npublic fun
ShortArray.sliceArray(indices: IntRange): ShortArray {\n  if (indices.isEmpty()) return ShortArray(0)\n  return
copyOfRange(indices.start, indices.endInclusive + 1)\n}\n\n/**\n * Returns an array containing elements at indices
in the specified [indices] range.\n */\npublic fun IntArray.sliceArray(indices: IntRange): IntArray {\n  if
(indices.isEmpty()) return IntArray(0)\n  return copyOfRange(indices.start, indices.endInclusive + 1)\n}\n\n/**\n *
Returns an array containing elements at indices in the specified [indices] range.\n */\npublic fun
LongArray.sliceArray(indices: IntRange): LongArray {\n  if (indices.isEmpty()) return LongArray(0)\n  return
copyOfRange(indices.start, indices.endInclusive + 1)\n}\n\n/**\n * Returns an array containing elements at indices

```

```

in the specified [indices] range.\n *\npublic fun FloatArray.sliceArray(indices: IntRange): FloatArray {\n  if
(indices.isEmpty()) return FloatArray(0)\n  return copyOfRange(indices.start, indices.endInclusive + 1)\n}\n\n/**\n * Returns an array containing elements at indices in the specified [indices] range.\n *\npublic fun
DoubleArray.sliceArray(indices: IntRange): DoubleArray {\n  if (indices.isEmpty()) return DoubleArray(0)\n
return copyOfRange(indices.start, indices.endInclusive + 1)\n}\n\n/**\n * Returns an array containing elements at
indices in the specified [indices] range.\n *\npublic fun BooleanArray.sliceArray(indices: IntRange): BooleanArray
{\n  if (indices.isEmpty()) return BooleanArray(0)\n  return copyOfRange(indices.start, indices.endInclusive +
1)\n}\n\n/**\n * Returns an array containing elements at indices in the specified [indices] range.\n *\npublic fun
CharArray.sliceArray(indices: IntRange): CharArray {\n  if (indices.isEmpty()) return CharArray(0)\n  return
copyOfRange(indices.start, indices.endInclusive + 1)\n}\n\n/**\n * Returns a list containing first [n] elements.\n *\n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n *\npublic fun <T> Array<out T>.take(n: Int): List<T> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n  if (n == 0) return emptyList()\n  if (n >=
size) return toList()\n  if (n == 1) return listOf(this[0])\n  var count = 0\n  val list = ArrayList<T>(n)\n  for
(item in this) {\n    list.add(item)\n    if (++count == n)\n      break\n  }\n  return list\n}\n\n/**\n * Returns
a list containing first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n *\npublic fun ByteArray.take(n: Int): List<Byte> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n  if (n == 0) return emptyList()\n  if (n >=
size) return toList()\n  if (n == 1) return listOf(this[0])\n  var count = 0\n  val list = ArrayList<Byte>(n)\n  for
(item in this) {\n    list.add(item)\n    if (++count == n)\n      break\n  }\n  return list\n}\n\n/**\n * Returns
a list containing first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n *\npublic fun ShortArray.take(n: Int): List<Short> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n  if (n == 0) return emptyList()\n  if (n >=
size) return toList()\n  if (n == 1) return listOf(this[0])\n  var count = 0\n  val list = ArrayList<Short>(n)\n  for
(item in this) {\n    list.add(item)\n    if (++count == n)\n      break\n  }\n  return list\n}\n\n/**\n * Returns
a list containing first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n *\npublic fun IntArray.take(n: Int): List<Int> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n  if (n == 0) return emptyList()\n  if (n >=
size) return toList()\n  if (n == 1) return listOf(this[0])\n  var count = 0\n  val list = ArrayList<Int>(n)\n  for
(item in this) {\n    list.add(item)\n    if (++count == n)\n      break\n  }\n  return list\n}\n\n/**\n * Returns
a list containing first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n *\npublic fun LongArray.take(n: Int): List<Long> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n  if (n == 0) return emptyList()\n  if (n >=
size) return toList()\n  if (n == 1) return listOf(this[0])\n  var count = 0\n  val list = ArrayList<Long>(n)\n  for
(item in this) {\n    list.add(item)\n    if (++count == n)\n      break\n  }\n  return list\n}\n\n/**\n * Returns
a list containing first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n *\npublic fun FloatArray.take(n: Int): List<Float> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n  if (n == 0) return emptyList()\n  if (n >=
size) return toList()\n  if (n == 1) return listOf(this[0])\n  var count = 0\n  val list = ArrayList<Float>(n)\n  for
(item in this) {\n    list.add(item)\n    if (++count == n)\n      break\n  }\n  return list\n}\n\n/**\n * Returns
a list containing first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n *\npublic fun DoubleArray.take(n: Int): List<Double> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n  if (n == 0) return emptyList()\n  if (n >=
size) return toList()\n  if (n == 1) return listOf(this[0])\n  var count = 0\n  val list = ArrayList<Double>(n)\n
for (item in this) {\n  list.add(item)\n  if (++count == n)\n    break\n }\n  return list\n}\n\n/**\n *
Returns a list containing first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n *
@sample samples.collections.Collections.Transformations.take\n *\npublic fun BooleanArray.take(n: Int):
List<Boolean> {\n  require(n >= 0) { \"Requested element count $n is less than zero.\" }\n  if (n == 0) return

```

```

emptyList()\n  if (n >= size) return toList()\n  if (n == 1) return listOf(this[0])\n  var count = 0\n  val list =
ArrayList<Boolean>()\n  for (item in this) {\n    list.add(item)\n    if (++count == n)\n      break\n  }\n  return list\n}\n\n/**\n * Returns a list containing first [n] elements.\n * \n * @throws IllegalArgumentException if
[n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.take\n */\npublic fun
CharArray.take(n: Int): List<Char> {\n  require(n >= 0) { \"Requested element count $n is less than zero.\" }\n  if
(n == 0) return emptyList()\n  if (n >= size) return toList()\n  if (n == 1) return listOf(this[0])\n  var count = 0\n  val list = ArrayList<Char>()\n  for (item in this) {\n    list.add(item)\n    if (++count == n)\n      break\n  }\n  return list\n}\n\n/**\n * Returns a list containing last [n] elements.\n * \n * @throws
IllegalArgument\nException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n */\npublic fun <T> Array<out T>.takeLast(n: Int): List<T>
{\n  require(n >= 0) { \"Requested element count $n is less than zero.\" }\n  if (n == 0) return emptyList()\n  val
size = size\n  if (n >= size) return toList()\n  if (n == 1) return listOf(this[size - 1])\n  val list =
ArrayList<T>()\n  for (index in size - n until size)\n    list.add(this[index])\n  return list\n}\n\n/**\n * Returns
a list containing last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n */\npublic fun ByteArray.takeLast(n: Int): List<Byte> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n  if (n == 0) return emptyList()\n  val size =
size\n  if (n >= size) return toList()\n  if (n == 1) return listOf(this[size - 1])\n  val list = ArrayList<Byte>()\n
for (index in size - n until size)\n  list.add(this[index])\n  return list\n}\n\n/**\n * Returns a list containing last
[n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n */\npublic fun ShortArray.takeLast(n: Int): List<Short> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n  if (n == 0) return emptyList()\n  val size =
size\n  if (n >= size) return toList()\n  if (n == 1) return listOf(this[size - 1])\n  val list = ArrayList<Short>()\n
for (index in size - n until size)\n  list.add(this[index])\n  return list\n}\n\n/**\n * Returns a list containing last
[n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n */\npublic fun IntArray.takeLast(n: Int): List<Int> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n  if (n == 0) return emptyList()\n  val size =
size\n  if (n >= size) return toList()\n  if (n == 1) return listOf(this[size - 1])\n  val list = ArrayList<Int>()\n
for (index in size - n until size)\n  list.add(this[index])\n  return list\n}\n\n/**\n * Returns a list containing last
[n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n */\npublic fun LongArray.takeLast(n: Int): List<Long> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n  if (n == 0) return emptyList()\n  val size =
size\n  if (n >= size) return toList()\n  if (n == 1) return listOf(this[size - 1])\n  val list = ArrayList<Long>()\n
for (index in size - n until size)\n  list.add(this[index])\n  return list\n}\n\n/**\n * Returns a list containing last
[n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n */\npublic fun FloatArray.takeLast(n: Int): List<Float> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n  if (n == 0) return emptyList()\n  val size =
size\n  if (n >= size) return toList()\n  if (n == 1) return listOf(this[size - 1])\n  val list = ArrayList<Float>()\n
for (index in size - n until size)\n  list.add(this[index])\n  return list\n}\n\n/**\n * Returns a list containing last
[n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n */\npublic fun DoubleArray.takeLast(n: Int): List<Double>
{\n  require(n >= 0) { \"Requested element count $n is less than zero.\" }\n  if (n == 0) return emptyList()\n  val
size = size\n  if (n >= size) return toList()\n  if (n == 1) return listOf(this[size - 1])\n  val list =
ArrayList<Double>()\n  for (index in size - n until size)\n    list.add(this[index])\n  return list\n}\n\n/**\n * Returns a list containing last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n */\npublic fun BooleanArray.takeLast(n: Int):
List<Boolean> {\n  require(n >= 0) { \"Requested element count $n is less than zero.\" }\n  if (n == 0) return
emptyList()\n  val size = size\n  if (n >= size) return toList()\n  if (n == 1) return listOf(this[size - 1])\n  val list
= ArrayList<Boolean>()\n  for (index in size - n until size)\n    list.add(this[index])\n  return list\n}\n\n/**\n *

```

Returns a list containing last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.take\n * \n public fun CharArray.takeLast(n: Int): List<Char> {\n require(n >= 0) { \"Requested element count \$n is less than zero.\" }\n if (n == 0) return emptyList()\n val size = size\n if (n >= size) return toList()\n if (n == 1) return listOf(this[size - 1])\n val list = ArrayList<Char>(n)\n for (index in size - n until size)\n list.add(this[index])\n return list\n}\n\n/**\n * Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n * \n public inline fun <T> Array<out T>.takeLastWhile(predicate: (T) -> Boolean): List<T> {\n for (index in lastIndex downTo 0) {\n if (!predicate(this[index])) {\n return drop(index + 1)\n }\n }\n return toList()\n}\n\n/**\n * Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n * \n public inline fun ByteArray.takeLastWhile(predicate: (Byte) -> Boolean): List<Byte> {\n for (index in lastIndex downTo 0) {\n if (!predicate(this[index])) {\n return drop(index + 1)\n }\n }\n return toList()\n}\n\n/**\n * Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n * \n public inline fun ShortArray.takeLastWhile(predicate: (Short) -> Boolean): List<Short> {\n for (index in lastIndex downTo 0) {\n if (!predicate(this[index])) {\n return drop(index + 1)\n }\n }\n return toList()\n}\n\n/**\n * Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n * \n public inline fun IntArray.takeLastWhile(predicate: (Int) -> Boolean): List<Int> {\n for (index in lastIndex downTo 0) {\n if (!predicate(this[index])) {\n return drop(index + 1)\n }\n }\n return toList()\n}\n\n/**\n * Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n * \n public inline fun LongArray.takeLastWhile(predicate: (Long) -> Boolean): List<Long> {\n for (index in lastIndex downTo 0) {\n if (!predicate(this[index])) {\n return drop(index + 1)\n }\n }\n return toList()\n}\n\n/**\n * Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n * \n public inline fun FloatArray.takeLastWhile(predicate: (Float) -> Boolean): List<Float> {\n for (index in lastIndex downTo 0) {\n if (!predicate(this[index])) {\n return drop(index + 1)\n }\n }\n return toList()\n}\n\n/**\n * Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n * \n public inline fun DoubleArray.takeLastWhile(predicate: (Double) -> Boolean): List<Double> {\n for (index in lastIndex downTo 0) {\n if (!predicate(this[index])) {\n return drop(index + 1)\n }\n }\n return toList()\n}\n\n/**\n * Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n * \n public inline fun BooleanArray.takeLastWhile(predicate: (Boolean) -> Boolean): List<Boolean> {\n for (index in lastIndex downTo 0) {\n if (!predicate(this[index])) {\n return drop(index + 1)\n }\n }\n return toList()\n}\n\n/**\n * Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n * \n public inline fun CharArray.takeLastWhile(predicate: (Char) -> Boolean): List<Char> {\n for (index in lastIndex downTo 0) {\n if (!predicate(this[index])) {\n return drop(index + 1)\n }\n }\n return toList()\n}\n\n/**\n * Returns a list containing first elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n * \n public inline fun <T> Array<out T>.takeWhile(predicate: (T) -> Boolean): List<T> {\n val list = ArrayList<T>()\n for (item in this) {\n if (!predicate(item))\n break\n list.add(item)\n }\n return list\n}\n\n/**\n * Returns a list containing first elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n * \n public inline fun ByteArray.takeWhile(predicate: (Byte) -> Boolean): List<Byte> {\n val list = ArrayList<Byte>()\n for (item in this) {\n if (!predicate(item))\n break\n list.add(item)\n }\n return list\n}\n\n/**\n * Returns a list containing first elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n * \n public inline fun ShortArray.takeWhile(predicate: (Short) -> Boolean): List<Short> {\n val list = ArrayList<Short>()\n for (item in this) {\n if (!predicate(item))\n break\n list.add(item)\n }\n return list\n}\n\n/**\n * Returns a

```

list containing first elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n *\npublic inline fun IntArray.takeWhile(predicate: (Int) ->
Boolean): List<Int> {\n    val list = ArrayList<Int>()\n    for (item in this) {\n        if (!predicate(item))\n        break\n        list.add(item)\n    }\n    return list\n}\n\n/**\n * Returns a list containing first elements satisfying the
given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n *\npublic inline fun LongArray.takeWhile(predicate: (Long) -> Boolean): List<Long> {\n    val list = ArrayList<Long>()\n    for (item
in this) {\n        if (!predicate(item))\n            break\n        list.add(item)\n    }\n    return list\n}\n\n/**\n * Returns a
list containing first elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n *\npublic inline fun FloatArray.takeWhile(predicate:
(Float) -> Boolean): List<Float> {\n    val list = ArrayList<Float>()\n    for (item in this) {\n        if
(!predicate(item))\n            break\n        list.add(item)\n    }\n    return list\n}\n\n/**\n * Returns a list containing first
elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n
*\npublic inline fun DoubleArray.takeWhile(predicate: (Double) -> Boolean): List<Double> {\n    val list =
ArrayList<Double>()\n    for (item in this) {\n        if (!predicate(item))\n            break\n        list.add(item)\n    }\n
return list\n}\n\n/**\n * Returns a list containing first elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n *\npublic inline fun BooleanArray.takeWhile(predicate:
(Boolean) -> Boolean): List<Boolean> {\n    val list = ArrayList<Boolean>()\n    for (item in this) {\n        if
(!predicate(item))\n            break\n        list.add(item)\n    }\n    return list\n}\n\n/**\n * Returns a list containing first
elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n
*\npublic inline fun CharArray.takeWhile(predicate: (Char) -> Boolean): List<Char> {\n    val list =
ArrayList<Char>()\n    for (item in this) {\n        if (!predicate(item))\n            break\n        list.add(item)\n    }\n
return list\n}\n\n/**\n * Reverses elements in the array in-place.\n *\npublic fun <T> Array<T>.reverse(): Unit {\n
val midPoint = (size / 2) - 1\n    if (midPoint < 0) return\n    var reverseIndex = lastIndex\n    for (index in
0..midPoint) {\n        val tmp = this[index]\n        this[index] = this[reverseIndex]\n        this[reverseIndex] = tmp\n
        reverseIndex--\n    }\n}\n\n/**\n * Reverses elements in the array in-place.\n *\npublic fun ByteArray.reverse():
Unit {\n    val midPoint = (size / 2) - 1\n    if (midPoint < 0) return\n    var reverseIndex = lastIndex\n    for (index in
0..midPoint) {\n        val tmp = this[index]\n        this[index] = this[reverseIndex]\n        this[reverseIndex] = tmp\n
        reverseIndex--\n    }\n}\n\n/**\n * Reverses elements in the array in-place.\n *\npublic fun ShortArray.reverse():
Unit {\n    val midPoint = (size / 2) - 1\n    if (midPoint < 0) return\n    var reverseIndex = lastIndex\n    for (index in
0..midPoint) {\n        val tmp = this[index]\n        this[index] = this[reverseIndex]\n        this[reverseIndex] = tmp\n
        reverseIndex--\n    }\n}\n\n/**\n * Reverses elements in the array in-place.\n *\npublic fun IntArray.reverse():
Unit {\n    val midPoint = (size / 2) - 1\n    if (midPoint < 0) return\n    var reverseIndex = lastIndex\n    for (index in
0..midPoint) {\n        val tmp = this[index]\n        this[index] = this[reverseIndex]\n        this[reverseIndex] = tmp\n
        reverseIndex--\n    }\n}\n\n/**\n * Reverses elements in the array in-place.\n *\npublic fun LongArray.reverse():
Unit {\n    val midPoint = (size / 2) - 1\n    if (midPoint < 0) return\n    var reverseIndex = lastIndex\n    for (index in
0..midPoint) {\n        val tmp = this[index]\n        this[index] = this[reverseIndex]\n        this[reverseIndex] = tmp\n
        reverseIndex--\n    }\n}\n\n/**\n * Reverses elements in the array in-place.\n *\npublic fun FloatArray.reverse():
Unit {\n    val midPoint = (size / 2) - 1\n    if (midPoint < 0) return\n    var reverseIndex = lastIndex\n    for (index in
0..midPoint) {\n        val tmp = this[index]\n        this[index] = this[reverseIndex]\n        this[reverseIndex] = tmp\n
        reverseIndex--\n    }\n}\n\n/**\n * Reverses elements in the array in-place.\n *\npublic fun DoubleArray.reverse(): Unit {\n
val midPoint = (size / 2) - 1\n    if (midPoint < 0) return\n    var reverseIndex =
lastIndex\n    for (index in 0..midPoint) {\n        val tmp = this[index]\n        this[index] = this[reverseIndex]\n
        this[reverseIndex] = tmp\n        reverseIndex--\n    }\n}\n\n/**\n * Reverses elements in the array in-place.\n
*\npublic fun BooleanArray.reverse(): Unit {\n    val midPoint = (size / 2) - 1\n    if (midPoint < 0) return\n    var
reverseIndex = lastIndex\n    for (index in 0..midPoint) {\n        val tmp = this[index]\n        this[index] =
this[reverseIndex]\n        this[reverseIndex] = tmp\n        reverseIndex--\n    }\n}\n\n/**\n * Reverses elements in the
array in-place.\n *\npublic fun CharArray.reverse(): Unit {\n    val midPoint = (size / 2) - 1\n    if (midPoint < 0)
return\n    var reverseIndex = lastIndex\n    for (index in 0..midPoint) {\n        val tmp = this[index]\n        this[index]
=

```

```

= this[reverseIndex]\n      this[reverseIndex] = tmp\n      reverseIndex--\n    }\n}\n\n/**\n * Reverses elements of
the array in the specified range in-place.\n * \n * @param fromIndex the start of the range (inclusive) to reverse.\n *
@param toIndex the end of the range (exclusive) to reverse.\n * \n * @throws IndexOutOfBoundsException if
[fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws
IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n @SinceKotlin("1.4")\npublic fun <T>
Array<T>.reverse(fromIndex: Int, toIndex: Int): Unit {\n    AbstractList.checkRangeIndexes(fromIndex, toIndex,
size)\n    val midPoint = (fromIndex + toIndex) / 2\n    if (fromIndex == midPoint) return\n    var reverseIndex =
toIndex - 1\n    for (index in fromIndex until midPoint) {\n        val tmp = this[index]\n        this[index] =
this[reverseIndex]\n        this[reverseIndex] = tmp\n        reverseIndex--\n    }\n}\n\n/**\n * Reverses elements of the
array in the specified range in-place.\n * \n * @param fromIndex the start of the range (inclusive) to reverse.\n *
@param toIndex the end of the range (exclusive) to reverse.\n * \n * @throws IndexOutOfBoundsException if
[fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws
IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n @SinceKotlin("1.4")\npublic fun
ByteArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n    AbstractList.checkRangeIndexes(fromIndex, toIndex,
size)\n    val midPoint = (fromIndex + toIndex) / 2\n    if (fromIndex == midPoint) return\n    var reverseIndex =
toIndex - 1\n    for (index in fromIndex until midPoint) {\n        val tmp = this[index]\n        this[index] =
this[reverseIndex]\n        this[reverseIndex] = tmp\n        reverseIndex--\n    }\n}\n\n/**\n * Reverses elements of the
array in the specified range in-place.\n * \n * @param fromIndex the start of the range (inclusive) to reverse.\n *
@param toIndex the end of the range (exclusive) to reverse.\n * \n * @throws IndexOutOfBoundsException if
[fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws
IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n @SinceKotlin("1.4")\npublic fun
ShortArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n    AbstractList.checkRangeIndexes(fromIndex, toIndex,
size)\n    val midPoint = (fromIndex + toIndex) / 2\n    if (fromIndex == midPoint) return\n    var reverseIndex =
toIndex - 1\n    for (index in fromIndex until midPoint) {\n        val tmp = this[index]\n        this[index] =
this[reverseIndex]\n        this[reverseIndex] = tmp\n        reverseIndex--\n    }\n}\n\n/**\n * Reverses elements of the
array in the specified range in-place.\n * \n * @param fromIndex the start of the range (inclusive) to reverse.\n *
@param toIndex the end of the range (exclusive) to reverse.\n * \n * @throws IndexOutOfBoundsException if
[fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws
IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n @SinceKotlin("1.4")\npublic fun
IntArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n    AbstractList.checkRangeIndexes(fromIndex, toIndex,
size)\n    val midPoint = (fromIndex + toIndex) / 2\n    if (fromIndex == midPoint) return\n    var reverseIndex =
toIndex - 1\n    for (index in fromIndex until midPoint) {\n        val tmp = this[index]\n        this[index] =
this[reverseIndex]\n        this[reverseIndex] = tmp\n        reverseIndex--\n    }\n}\n\n/**\n * Reverses elements of the
array in the specified range in-place.\n * \n * @param fromIndex the start of the range (inclusive) to reverse.\n *
@param toIndex the end of the range (exclusive) to reverse.\n * \n * @throws IndexOutOfBoundsException if
[fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws
IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n @SinceKotlin("1.4")\npublic fun
LongArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n    AbstractList.checkRangeIndexes(fromIndex, toIndex,
size)\n    val midPoint = (fromIndex + toIndex) / 2\n    if (fromIndex == midPoint) return\n    var reverseIndex =
toIndex - 1\n    for (index in fromIndex until midPoint) {\n        val tmp = this[index]\n        this[index] =
this[reverseIndex]\n        this[reverseIndex] = tmp\n        reverseIndex--\n    }\n}\n\n/**\n * Reverses elements of the
array in the specified range in-place.\n * \n * @param fromIndex the start of the range (inclusive) to reverse.\n *
@param toIndex the end of the range (exclusive) to reverse.\n * \n * @throws IndexOutOfBoundsException if
[fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws
IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n @SinceKotlin("1.4")\npublic fun
FloatArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n    AbstractList.checkRangeIndexes(fromIndex, toIndex,
size)\n    val midPoint = (fromIndex + toIndex) / 2\n    if (fromIndex == midPoint) return\n    var reverseIndex =
toIndex - 1\n    for (index in fromIndex until midPoint) {\n        val tmp = this[index]\n        this[index] =

```

```

this[reverseIndex]\n    this[reverseIndex] = tmp\n    reverseIndex--\n    }\n}\n\n/**\n * Reverses elements of the
array in the specified range in-place.\n * \n * @param fromIndex the start of the range (inclusive) to reverse.\n *
@param toIndex the end of the range (exclusive) to reverse.\n * \n * @throws IndexOutOfBoundsException if
[fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws
IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n\n@SinceKotlin("1.4")\npublic fun
DoubleArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n    AbstractList.checkRangeIndexes(fromIndex, toIndex,
size)\n    val midPoint = (fromIndex + toIndex) / 2\n    if (fromIndex == midPoint) return\n    var reverseIndex =
toIndex - 1\n    for (index in fromIndex until midPoint) {\n        val tmp = this[index]\n        this[index] =
this[reverseIndex]\n        this[reverseIndex] = tmp\n        reverseIndex--\n    }\n}\n\n/**\n * Reverses elements of the
array in the specified range in-place.\n * \n * @param fromIndex the start of the range (inclusive) to reverse.\n *
@param toIndex the end of the range (exclusive) to reverse.\n * \n * @throws IndexOutOfBoundsException if
[fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws
IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n\n@SinceKotlin("1.4")\npublic fun
BooleanArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n    AbstractList.checkRangeIndexes(fromIndex,
toIndex, size)\n    val midPoint = (fromIndex + toIndex) / 2\n    if (fromIndex == midPoint) return\n    var
reverseIndex = toIndex - 1\n    for (index in fromIndex until midPoint) {\n        val tmp = this[index]\n
this[index] = this[reverseIndex]\n        this[reverseIndex] = tmp\n        reverseIndex--\n    }\n}\n\n/**\n * Reverses
elements of the array in the specified range in-place.\n * \n * @param fromIndex the start of the range (inclusive) to
reverse.\n * \n * @param toIndex the end of the range (exclusive) to reverse.\n * \n * @throws
IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n *
@throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n\n@SinceKotlin("1.4")\npublic
fun CharArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n    AbstractList.checkRangeIndexes(fromIndex,
toIndex, size)\n    val midPoint = (fromIndex + toIndex) / 2\n    if (fromIndex == midPoint) return\n    var
reverseIndex = toIndex - 1\n    for (index in fromIndex until midPoint) {\n        val tmp = this[index]\n
this[index] = this[reverseIndex]\n        this[reverseIndex] = tmp\n        reverseIndex--\n    }\n}\n\n/**\n * Returns a
list with elements in reversed order.\n * \n\npublic fun <T> Array<out T>.reversed(): List<T> {\n    if (isEmpty())
return emptyList()\n    val list = toMutableList()\n    list.reverse()\n    return list\n}\n\n/**\n * Returns a list with
elements in reversed order.\n * \n\npublic fun ByteArray.reversed(): List<Byte> {\n    if (isEmpty()) return
emptyList()\n    val list = toMutableList()\n    list.reverse()\n    return list\n}\n\n/**\n * Returns a list with elements
in reversed order.\n * \n\npublic fun ShortArray.reversed(): List<Short> {\n    if (isEmpty()) return emptyList()\n
val list = toMutableList()\n    list.reverse()\n    return list\n}\n\n/**\n * Returns a list with elements in reversed
order.\n * \n\npublic fun IntArray.reversed(): List<Int> {\n    if (isEmpty()) return emptyList()\n    val list =
toMutableList()\n    list.reverse()\n    return list\n}\n\n/**\n * Returns a list with elements in reversed order.\n *
\n\npublic fun LongArray.reversed(): List<Long> {\n    if (isEmpty()) return emptyList()\n    val list =
toMutableList()\n    list.reverse()\n    return list\n}\n\n/**\n * Returns a list with elements in reversed order.\n *
\n\npublic fun FloatArray.reversed(): List<Float> {\n    if (isEmpty()) return emptyList()\n    val list =
toMutableList()\n    list.reverse()\n    return list\n}\n\n/**\n * Returns a list with elements in reversed order.\n *
\n\npublic fun DoubleArray.reversed(): List<Double> {\n    if (isEmpty()) return emptyList()\n    val list =
toMutableList()\n    list.reverse()\n    return list\n}\n\n/**\n * Returns a list with elements in reversed order.\n *
\n\npublic fun BooleanArray.reversed(): List<Boolean> {\n    if (isEmpty()) return emptyList()\n    val list =
toMutableList()\n    list.reverse()\n    return list\n}\n\n/**\n * Returns a list with elements in reversed order.\n *
\n\npublic fun CharArray.reversed(): List<Char> {\n    if (isEmpty()) return emptyList()\n    val list =
toMutableList()\n    list.reverse()\n    return list\n}\n\n/**\n * Returns an array with elements of this array in
reversed order.\n * \n\npublic fun <T> Array<T>.reversedArray(): Array<T> {\n    if (isEmpty()) return this\n    val
result = arrayOfNulls(this, size)\n    val lastIndex = lastIndex\n    for (i in 0..lastIndex)\n        result[lastIndex - i] = this[i]\n    return
result\n}\n\n/**\n * Returns an array with elements of this array in reversed order.\n * \n\npublic fun
ByteArray.reversedArray(): ByteArray {\n    if (isEmpty()) return this\n    val result = ByteArray(size)\n    val
lastIndex = lastIndex\n    for (i in 0..lastIndex)\n        result[lastIndex - i] = this[i]\n    return result\n}\n\n/**

```

```

Returns an array with elements of this array in reversed order.\n */\npublic fun ShortArray.reversedArray():
ShortArray {\n if (isEmpty()) return this\n val result = ShortArray(size)\n val lastIndex = lastIndex\n for (i in
0..lastIndex)\n result[lastIndex - i] = this[i]\n return result\n}\n\n/**\n * Returns an array with elements of this
array in reversed order.\n */\npublic fun IntArray.reversedArray(): IntArray {\n if (isEmpty()) return this\n val
result = IntArray(size)\n val lastIndex = lastIndex\n for (i in 0..lastIndex)\n result[lastIndex - i] = this[i]\n
return result\n}\n\n/**\n * Returns an array with elements of this array in reversed order.\n */\npublic fun
LongArray.reversedArray(): LongArray {\n if (isEmpty()) return this\n val result = LongArray(size)\n val
lastIndex = lastIndex\n for (i in 0..lastIndex)\n result[lastIndex - i] = this[i]\n return result\n}\n\n/**\n *
Returns an array with elements of this array in reversed order.\n */\npublic fun FloatArray.reversedArray():
FloatArray {\n if (isEmpty()) return this\n val result = FloatArray(size)\n val lastIndex = lastIndex\n for (i in
0..lastIndex)\n result[lastIndex - i] = this[i]\n return result\n}\n\n/**\n * Returns an array with elements of this
array in reversed order.\n */\npublic fun DoubleArray.reversedArray(): DoubleArray {\n if (isEmpty()) return
this\n val result = DoubleArray(size)\n val lastIndex = lastIndex\n for (i in 0..lastIndex)\n result[lastIndex -
i] = this[i]\n return result\n}\n\n/**\n * Returns an array with elements of this array in reversed order.\n */\npublic
fun BooleanArray.reversedArray(): BooleanArray {\n if (isEmpty()) return this\n val result =
BooleanArray(size)\n val lastIndex = lastIndex\n for (i in 0..lastIndex)\n result[lastIndex - i] = this[i]\n
return result\n}\n\n/**\n * Returns an array with elements of this array in reversed order.\n */\npublic fun
CharArray.reversedArray(): CharArray {\n if (isEmpty()) return this\n val result = CharArray(size)\n val
lastIndex = lastIndex\n for (i in 0..lastIndex)\n result[lastIndex - i] = this[i]\n return result\n}\n\n/**\n *
Randomly shuffles elements in this array in-place.\n */\n@SinceKotlin("1.4")\npublic fun <T>
Array<T>.shuffle(): Unit {\n shuffle(Random)\n}\n\n/**\n * Randomly shuffles elements in this array in-place.\n */\n@SinceKotlin("1.4")\npublic fun ByteArray.shuffle(): Unit {\n shuffle(Random)\n}\n\n/**\n * Randomly
shuffles elements in this array in-place.\n */\n@SinceKotlin("1.4")\npublic fun ShortArray.shuffle(): Unit {\n
shuffle(Random)\n}\n\n/**\n * Randomly shuffles elements in this array in-place.\n */\n@SinceKotlin("1.4")\npublic fun IntArray.shuffle(): Unit {\n shuffle(Random)\n}\n\n/**\n * Randomly
shuffles elements in this array in-place.\n */\n@SinceKotlin("1.4")\npublic fun LongArray.shuffle(): Unit {\n
shuffle(Random)\n}\n\n/**\n * Randomly shuffles elements in this array in-place.\n */\n@SinceKotlin("1.4")\npublic fun FloatArray.shuffle(): Unit {\n shuffle(Random)\n}\n\n/**\n * Randomly
shuffles elements in this array in-place.\n */\n@SinceKotlin("1.4")\npublic fun DoubleArray.shuffle(): Unit {\n
shuffle(Random)\n}\n\n/**\n * Randomly shuffles elements in this array in-place.\n */\n@SinceKotlin("1.4")\npublic fun BooleanArray.shuffle(): Unit {\n shuffle(Random)\n}\n\n/**\n * Randomly
shuffles elements in this array in-place using the specified [random]
instance as the source of randomness.\n */\n * See:
https://en.wikipedia.org/wiki/Fisher%20%80%93Yates\_shuffle#The\_modern\_algorithm\n
*/\n@SinceKotlin("1.4")\npublic fun <T> Array<T>.shuffle(random: Random): Unit {\n for (i in lastIndex
downTo 1) {\n val j = random.nextInt(i + 1)\n val copy = this[i]\n this[i] = this[j]\n this[j] = copy\n
}\n}\n\n/**\n * Randomly shuffles elements in this array in-place using the specified [random] instance as the
source of randomness.\n */\n * See:
https://en.wikipedia.org/wiki/Fisher%20%80%93Yates\_shuffle#The\_modern\_algorithm\n
*/\n@SinceKotlin("1.4")\npublic fun ByteArray.shuffle(random: Random): Unit {\n for (i in lastIndex downTo
1) {\n val j = random.nextInt(i + 1)\n val copy = this[i]\n this[i] = this[j]\n this[j] = copy\n
}\n}\n\n/**\n * Randomly shuffles elements in this array in-place using the specified [random] instance as the
source of randomness.\n */\n * See:
https://en.wikipedia.org/wiki/Fisher%20%80%93Yates\_shuffle#The\_modern\_algorithm\n
*/\n@SinceKotlin("1.4")\npublic fun ShortArray.shuffle(random: Random): Unit {\n for (i in lastIndex downTo
1) {\n val j = random.nextInt(i + 1)\n val copy = this[i]\n this[i] = this[j]\n this[j] = copy\n
}\n}\n\n/**\n * Randomly shuffles elements in this array in-place using the specified [random] instance as the

```


1) `{\n sort()\n reverse()\n }{\n}\n\n/**\n * Sorts elements in the array in-place descending according to their natural sort order.\n */\npublic fun DoubleArray.sortDescending(): Unit {\n if (size > 1) {\n sort()\n reverse()\n }\n}\n\n/**\n * Sorts elements in the array in-place descending according to their natural sort order.\n */\npublic fun CharArray.sortDescending(): Unit {\n if (size > 1) {\n sort()\n reverse()\n }\n}\n\n/**\n * Returns a list of all elements sorted according to their natural sort order.\n */\n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n */\npublic fun <T : Comparable<T>> Array<out T>.sorted(): List<T> {\n return sortedArray().asList()\n}\n\n/**\n * Returns a list of all elements sorted according to their natural sort order.\n */\npublic fun ByteArray.sorted(): List<Byte> {\n return toTypedArray().apply { sort() }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to their natural sort order.\n */\npublic fun ShortArray.sorted(): List<Short> {\n return toTypedArray().apply { sort() }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to their natural sort order.\n */\npublic fun IntArray.sorted(): List<Int> {\n return toTypedArray().apply { sort() }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to their natural sort order.\n */\npublic fun LongArray.sorted(): List<Long> {\n return toTypedArray().apply { sort() }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to their natural sort order.\n */\npublic fun FloatArray.sorted(): List<Float> {\n return toTypedArray().apply { sort() }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to their natural sort order.\n */\npublic fun DoubleArray.sorted(): List<Double> {\n return toTypedArray().apply { sort() }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to their natural sort order.\n */\npublic fun CharArray.sorted(): List<Char> {\n return toTypedArray().apply { sort() }.asList()\n}\n\n/**\n * Returns an array with all elements of this array sorted according to their natural sort order.\n */\n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n */\npublic fun <T : Comparable<T>> Array<T>.sortedArray(): Array<T> {\n if (isEmpty()) return this\n return this.copyOf().apply { sort() }\n}\n\n/**\n * Returns an array with all elements of this array sorted according to their natural sort order.\n */\npublic fun ByteArray.sortedArray(): ByteArray {\n if (isEmpty()) return this\n return this.copyOf().apply { sort() }\n}\n\n/**\n * Returns an array with all elements of this array sorted according to their natural sort order.\n */\npublic fun ShortArray.sortedArray(): ShortArray {\n if (isEmpty()) return this\n return this.copyOf().apply { sort() }\n}\n\n/**\n * Returns an array with all elements of this array sorted according to their natural sort order.\n */\npublic fun IntArray.sortedArray(): IntArray {\n if (isEmpty()) return this\n return this.copyOf().apply { sort() }\n}\n\n/**\n * Returns an array with all elements of this array sorted according to their natural sort order.\n */\npublic fun LongArray.sortedArray(): LongArray {\n if (isEmpty()) return this\n return this.copyOf().apply { sort() }\n}\n\n/**\n * Returns an array with all elements of this array sorted according to their natural sort order.\n */\npublic fun FloatArray.sortedArray(): FloatArray {\n if (isEmpty()) return this\n return this.copyOf().apply { sort() }\n}\n\n/**\n * Returns an array with all elements of this array sorted according to their natural sort order.\n */\npublic fun DoubleArray.sortedArray(): DoubleArray {\n if (isEmpty()) return this\n return this.copyOf().apply { sort() }\n}\n\n/**\n * Returns an array with all elements of this array sorted according to their natural sort order.\n */\npublic fun CharArray.sortedArray(): CharArray {\n if (isEmpty()) return this\n return this.copyOf().apply { sort() }\n}\n\n/**\n * Returns an array with all elements of this array sorted descending according to their natural sort order.\n */\n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n */\npublic fun <T : Comparable<T>> Array<T>.sortedArrayDescending(): Array<T> {\n if (isEmpty()) return this\n return this.copyOf().apply { sortWith(reverseOrder()) }\n}\n\n/**\n * Returns an array with all elements of this array sorted descending according to their natural sort order.\n */\npublic fun ByteArray.sortedArrayDescending(): ByteArray {\n if (isEmpty()) return this\n return this.copyOf().apply { sortDescending() }\n}\n\n/**\n * Returns an array with all elements of this array sorted descending according to their natural sort order.\n */\npublic fun ShortArray.sortedArrayDescending(): ShortArray {\n if (isEmpty()) return this\n return this.copyOf().apply { sortDescending() }\n}\n\n/**\n * Returns an array with all elements of this array sorted descending according to their natural sort order.\n */\npublic fun IntArray.sortedArrayDescending(): IntArray {\n if (isEmpty()) return this\n return this.copyOf().apply { sortDescending() }\n}\n\n/**\n * Returns an array with all elements of this array sorted descending according to their natural sort order.\n */\npublic fun LongArray.sortedArrayDescending():`

```

LongArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sortDescending() }
}
/**
 * Returns an array with all elements of this array sorted descending according to their natural sort order.
 */
public fun FloatArray.sortedArrayDescending(): FloatArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sortDescending() }
}
/**
 * Returns an array with all elements of this array sorted descending according to their natural sort order.
 */
public fun DoubleArray.sortedArrayDescending(): DoubleArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sortDescending() }
}
/**
 * Returns an array with all elements of this array sorted descending according to their natural sort order.
 */
public fun CharArray.sortedArrayDescending(): CharArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sortDescending() }
}
/**
 * Returns an array with all elements of this array sorted according the specified [comparator].
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 */
public fun <T> Array<out T>.sortedArrayWith(comparator: Comparator<in T>): Array<out T> {
    if (isEmpty()) return this
    return this.copyOf().apply { sortWith(comparator) }
}
/**
 * Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 */
@sample samples.collections.Collections.Sorting.sortedBy
public inline fun <T, R : Comparable<R>> Array<out T>.sortedBy(crossinline selector: (T) -> R?): List<T> {
    return sortedWith(compareBy(selector))
}
/**
 * Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.
 */
@sample samples.collections.Collections.Sorting.sortedBy
public inline fun <R : Comparable<R>> ByteArray.sortedBy(crossinline selector: (Byte) -> R?): List<Byte> {
    return sortedWith(compareBy(selector))
}
/**
 * Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.
 */
@sample samples.collections.Collections.Sorting.sortedBy
public inline fun <R : Comparable<R>> ShortArray.sortedBy(crossinline selector: (Short) -> R?): List<Short> {
    return sortedWith(compareBy(selector))
}
/**
 * Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.
 */
@sample samples.collections.Collections.Sorting.sortedBy
public inline fun <R : Comparable<R>> IntArray.sortedBy(crossinline selector: (Int) -> R?): List<Int> {
    return sortedWith(compareBy(selector))
}
/**
 * Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.
 */
@sample samples.collections.Collections.Sorting.sortedBy
public inline fun <R : Comparable<R>> LongArray.sortedBy(crossinline selector: (Long) -> R?): List<Long> {
    return sortedWith(compareBy(selector))
}
/**
 * Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.
 */
@sample samples.collections.Collections.Sorting.sortedBy
public inline fun <R : Comparable<R>> FloatArray.sortedBy(crossinline selector: (Float) -> R?): List<Float> {
    return sortedWith(compareBy(selector))
}
/**
 * Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.
 */
@sample samples.collections.Collections.Sorting.sortedBy
public inline fun <R : Comparable<R>> DoubleArray.sortedBy(crossinline selector: (Double) -> R?): List<Double> {
    return sortedWith(compareBy(selector))
}
/**
 * Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.
 */
@sample samples.collections.Collections.Sorting.sortedBy
public inline fun <R : Comparable<R>> BooleanArray.sortedBy(crossinline selector: (Boolean) -> R?): List<Boolean> {
    return sortedWith(compareBy(selector))
}
/**
 * Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.
 */
@sample samples.collections.Collections.Sorting.sortedBy
public inline fun <R : Comparable<R>> CharArray.sortedBy(crossinline selector: (Char) -> R?): List<Char> {
    return

```

sortedWith(compareBy(selector))\n}\n\n/**\n * Returns a list of all elements sorted descending according to natural sort order of the value returned by specified [selector] function.\n * \n * The sort is `_stable_`. It means that equal elements preserve their order relative to each other after sorting.\n */\npublic inline fun <T, R : Comparable<R>> Array<out T>.sortedByDescending(crossinline selector: (T) -> R?): List<T> {\n return sortedWith(compareByDescending(selector))\n}\n\n/**\n * Returns a list of all elements sorted descending according to natural sort order of the value returned by specified [selector] function.\n */\npublic inline fun <R : Comparable<R>> ByteArray.sortedByDescending(crossinline selector: (Byte) -> R?): List<Byte> {\n return sortedWith(compareByDescending(selector))\n}\n\n/**\n * Returns a list of all elements sorted descending according to natural sort order of the value returned by specified [selector] function.\n */\npublic inline fun <R : Comparable<R>> ShortArray.sortedByDescending(crossinline selector: (Short) -> R?): List<Short> {\n return sortedWith(compareByDescending(selector))\n}\n\n/**\n * Returns a list of all elements sorted descending according to natural sort order of the value returned by specified [selector] function.\n */\npublic inline fun <R : Comparable<R>> IntArray.sortedByDescending(crossinline selector: (Int) -> R?): List<Int> {\n return sortedWith(compareByDescending(selector))\n}\n\n/**\n * Returns a list of all elements sorted descending according to natural sort order of the value returned by specified [selector] function.\n */\npublic inline fun <R : Comparable<R>> LongArray.sortedByDescending(crossinline selector: (Long) -> R?): List<Long> {\n return sortedWith(compareByDescending(selector))\n}\n\n/**\n * Returns a list of all elements sorted descending according to natural sort order of the value returned by specified [selector] function.\n */\npublic inline fun <R : Comparable<R>> FloatArray.sortedByDescending(crossinline selector: (Float) -> R?): List<Float> {\n return sortedWith(compareByDescending(selector))\n}\n\n/**\n * Returns a list of all elements sorted descending according to natural sort order of the value returned by specified [selector] function.\n */\npublic inline fun <R : Comparable<R>> DoubleArray.sortedByDescending(crossinline selector: (Double) -> R?): List<Double> {\n return sortedWith(compareByDescending(selector))\n}\n\n/**\n * Returns a list of all elements sorted descending according to natural sort order of the value returned by specified [selector] function.\n */\npublic inline fun <R : Comparable<R>> BooleanArray.sortedByDescending(crossinline selector: (Boolean) -> R?): List<Boolean> {\n return sortedWith(compareByDescending(selector))\n}\n\n/**\n * Returns a list of all elements sorted descending according to natural sort order of the value returned by specified [selector] function.\n */\npublic inline fun <R : Comparable<R>> CharArray.sortedByDescending(crossinline selector: (Char) -> R?): List<Char> {\n return sortedWith(compareByDescending(selector))\n}\n\n/**\n * Returns a list of all elements sorted descending according to their natural sort order.\n * \n * The sort is `_stable_`. It means that equal elements preserve their order relative to each other after sorting.\n */\npublic fun <T : Comparable<T>> Array<out T>.sortedDescending(): List<T> {\n return sortedWith(reverseOrder())\n}\n\n/**\n * Returns a list of all elements sorted descending according to their natural sort order.\n */\npublic fun ByteArray.sortedDescending(): List<Byte> {\n return copyOf().apply { sort() }.reversed()\n}\n\n/**\n * Returns a list of all elements sorted descending according to their natural sort order.\n */\npublic fun ShortArray.sortedDescending(): List<Short> {\n return copyOf().apply { sort() }.reversed()\n}\n\n/**\n * Returns a list of all elements sorted descending according to their natural sort order.\n */\npublic fun IntArray.sortedDescending(): List<Int> {\n return copyOf().apply { sort() }.reversed()\n}\n\n/**\n * Returns a list of all elements sorted descending according to their natural sort order.\n */\npublic fun LongArray.sortedDescending(): List<Long> {\n return copyOf().apply { sort() }.reversed()\n}\n\n/**\n * Returns a list of all elements sorted descending according to their natural sort order.\n */\npublic fun FloatArray.sortedDescending(): List<Float> {\n return copyOf().apply { sort() }.reversed()\n}\n\n/**\n * Returns a list of all elements sorted descending according to their natural sort order.\n */\npublic fun DoubleArray.sortedDescending(): List<Double> {\n return copyOf().apply { sort() }.reversed()\n}\n\n/**\n * Returns a list of all elements sorted descending according to their natural sort order.\n */\npublic fun CharArray.sortedDescending(): List<Char> {\n return copyOf().apply { sort() }.reversed()\n}\n\n/**\n * Returns a list of all elements sorted according to the specified [comparator].\n * \n * The sort is `_stable_`. It means that equal elements preserve their order relative to each other after sorting.\n */\npublic fun <T> Array<out T>.sortedWith(comparator: Comparator<in T>): List<T> {\n return

sortedArrayWith(comparator).asList()\n\n/**\n * Returns a list of all elements sorted according to the specified [comparator].\n * \n\npublic fun ByteArray.sortedWith(comparator: Comparator<in Byte>): List<Byte> {\n return toTypedArray().apply { sortWith(comparator) }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to the specified [comparator].\n * \n\npublic fun ShortArray.sortedWith(comparator: Comparator<in Short>): List<Short> {\n return toTypedArray().apply { sortWith(comparator) }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to the specified [comparator].\n * \n\npublic fun IntArray.sortedWith(comparator: Comparator<in Int>): List<Int> {\n return toTypedArray().apply { sortWith(comparator) }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to the specified [comparator].\n * \n\npublic fun LongArray.sortedWith(comparator: Comparator<in Long>): List<Long> {\n return toTypedArray().apply { sortWith(comparator) }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to the specified [comparator].\n * \n\npublic fun FloatArray.sortedWith(comparator: Comparator<in Float>): List<Float> {\n return toTypedArray().apply { sortWith(comparator) }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to the specified [comparator].\n * \n\npublic fun DoubleArray.sortedWith(comparator: Comparator<in Double>): List<Double> {\n return toTypedArray().apply { sortWith(comparator) }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to the specified [comparator].\n * \n\npublic fun BooleanArray.sortedWith(comparator: Comparator<in Boolean>): List<Boolean> {\n return toTypedArray().apply { sortWith(comparator) }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to the specified [comparator].\n * \n\npublic fun CharArray.sortedWith(comparator: Comparator<in Char>): List<Char> {\n return toTypedArray().apply { sortWith(comparator) }.asList()\n}\n\n/**\n * Returns a [List] that wraps the original array.\n * \n\npublic expect fun <T> Array<out T>.asList(): List<T>\n\n/**\n * Returns a [List] that wraps the original array.\n * \n\npublic expect fun ByteArray.asList(): List<Byte>\n\n/**\n * Returns a [List] that wraps the original array.\n * \n\npublic expect fun ShortArray.asList(): List<Short>\n\n/**\n * Returns a [List] that wraps the original array.\n * \n\npublic expect fun IntArray.asList(): List<Int>\n\n/**\n * Returns a [List] that wraps the original array.\n * \n\npublic expect fun LongArray.asList(): List<Long>\n\n/**\n * Returns a [List] that wraps the original array.\n * \n\npublic expect fun FloatArray.asList(): List<Float>\n\n/**\n * Returns a [List] that wraps the original array.\n * \n\npublic expect fun DoubleArray.asList(): List<Double>\n\n/**\n * Returns a [List] that wraps the original array.\n * \n\npublic expect fun BooleanArray.asList(): List<Boolean>\n\n/**\n * Returns a [List] that wraps the original array.\n * \n\npublic expect fun CharArray.asList(): List<Char>\n\n/**\n * Returns `true` if the two specified arrays are *deeply* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n\n * If two corresponding elements are nested arrays, they are also compared deeply.\n * If any of arrays contains itself on any nesting level the behavior is undefined.\n * \n\n * The elements of other types are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n * \n\n@SinceKotlin("1.1")\n@kotlin.internal.LowPriorityInOverloadResolution\npublic expect infix fun <T> Array<out T>.contentDeepEquals(other: Array<out T>): Boolean\n\n/**\n * Returns `true` if the two specified arrays are *deeply* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n\n * The specified arrays are also considered deeply equal if both are `null`.\n * \n\n * If two corresponding elements are nested arrays, they are also compared deeply.\n * If any of arrays contains itself on any nesting level the behavior is undefined.\n * \n\n * The elements of other types are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n * \n\n@SinceKotlin("1.4")\npublic expect infix fun <T> Array<out T>?.contentDeepEquals(other: Array<out T>?): Boolean\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n * \n\n * Nested arrays are treated as lists too.\n * \n\n * If any of arrays contains itself on any nesting level the behavior is undefined.\n * \n\n@SinceKotlin("1.1")\n@kotlin.internal.LowPriorityInOverloadResolution\npublic expect fun <T> Array<out T>.contentDeepHashCode(): Int\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n * \n\n * Nested arrays are treated as lists too.\n * \n\n * If any of arrays contains itself on any nesting level the behavior is undefined.\n * \n\n@SinceKotlin("1.4")\npublic expect fun <T> Array<out T>?.contentDeepHashCode(): Int\n\n/**\n * Returns a string representation of the contents of this array as if it is a [List].\n * \n\n * Nested arrays are treated as lists too.\n * \n\n * If any of arrays contains itself on any nesting level that

reference\n * is rendered as `\"[...]\"` to prevent recursion.\n * \n * @sample
samples.collections.Arrays.ContentOperations.contentDeepToString\n

`*\n@SinceKotlin(\"1.1\")\n@kotlin.internal.LowPriorityInOverloadResolution\npublic expect fun <T> Array<out T>.contentDeepToString(): String\n\n/**\n * Returns a string representation of the contents of this array as if it is a [List].\n * Nested arrays are treated as lists too.\n * \n * If any of arrays contains itself on any nesting level that reference\n * is rendered as \"[...]\" to prevent recursion.\n * \n * @sample
samples.collections.Arrays.ContentOperations.contentDeepToString\n * \n@SinceKotlin(\"1.4\")\npublic expect fun <T> Array<out T>?.contentDeepToString(): String\n\n/**\n * Returns true if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that NaN is equal to itself and -0.0 is not equal to 0.0.\n * \n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic expect infix fun <T> Array<out T>.contentEquals(other: Array<out T>): Boolean\n\n/**\n * Returns true if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that NaN is equal to itself and -0.0 is not equal to 0.0.\n * \n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic expect infix fun ByteArray.contentEquals(other: ByteArray): Boolean\n\n/**\n * Returns true if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that NaN is equal to itself and -0.0 is not equal to 0.0.\n * \n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic expect infix fun ShortArray.contentEquals(other: ShortArray): Boolean\n\n/**\n * Returns true if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that NaN is equal to itself and -0.0 is not equal to 0.0.\n * \n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic expect infix fun IntArray.contentEquals(other: IntArray): Boolean\n\n/**\n * Returns true if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that NaN is equal to itself and -0.0 is not equal to 0.0.\n * \n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic expect infix fun LongArray.contentEquals(other: LongArray): Boolean\n\n/**\n * Returns true if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that NaN is equal to itself and -0.0 is not equal to 0.0.\n * \n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic expect infix fun FloatArray.contentEquals(other: FloatArray): Boolean\n\n/**\n * Returns true if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that NaN is equal to itself and -0.0 is not equal to 0.0.\n * \n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic expect infix fun DoubleArray.contentEquals(other: DoubleArray): Boolean\n\n/**\n * Returns true if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that NaN is equal to itself and -0.0 is not equal to 0.0.\n * \n`

```

*\n@Deprecated("\nUse Kotlin compiler 1.4 to avoid deprecation
warning.\n")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic expect infix fun
BooleanArray.contentEquals(other: BooleanArray): Boolean\n\n**\n * Returns `true` if the two specified arrays are
*structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n
* The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it
means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n *\n@Deprecated("\nUse Kotlin compiler 1.4 to
avoid deprecation warning.\n")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic
expect infix fun CharArray.contentEquals(other: CharArray): Boolean\n\n**\n * Returns `true` if the two specified
arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same
order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point
numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n *\n@SinceKotlin("1.4")\npublic
expect infix fun <T> Array<out T>?.contentEquals(other: Array<out T>?): Boolean\n\n**\n * Returns `true` if the
two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements
in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For
floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n
*\n@SinceKotlin("1.4")\npublic expect infix fun ByteArray?.contentEquals(other: ByteArray?): Boolean\n\n**\n
* Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number
of the same elements in the same order.\n * \n * The elements are compared for equality with the
[equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not
equal to `0.0`.\n *\n@SinceKotlin("1.4")\npublic expect infix fun ShortArray?.contentEquals(other: ShortArray?):
Boolean\n\n**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain
the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the
[equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not
equal to `0.0`.\n *\n@SinceKotlin("1.4")\npublic expect infix fun IntArray?.contentEquals(other: IntArray?):
Boolean\n\n**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain
the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the
[equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not
equal to `0.0`.\n *\n@SinceKotlin("1.4")\npublic expect infix fun LongArray?.contentEquals(other: LongArray?):
Boolean\n\n**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain
the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the
[equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not
equal to `0.0`.\n *\n@SinceKotlin("1.4")\npublic expect infix fun FloatArray?.contentEquals(other: FloatArray?):
Boolean\n\n**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain
the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the
[equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not
equal to `0.0`.\n *\n@SinceKotlin("1.4")\npublic expect infix fun DoubleArray?.contentEquals(other:
DoubleArray?): Boolean\n\n**\n * Returns `true` if the two specified arrays are *structurally* equal to one
another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are
compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is
equal to itself and `-0.0` is not equal to `0.0`.\n *\n@SinceKotlin("1.4")\npublic expect infix fun
BooleanArray?.contentEquals(other: BooleanArray?): Boolean\n\n**\n * Returns `true` if the two specified arrays
are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n *
*\n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers
it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n *\n@SinceKotlin("1.4")\npublic expect
infix fun CharArray?.contentEquals(other: CharArray?): Boolean\n\n**\n * Returns a hash code based on the
contents of this array as if it is [List].\n *\n@Deprecated("\nUse Kotlin compiler 1.4 to avoid deprecation
warning.\n")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic expect fun <T>
Array<out T>.contentHashCode(): Int\n\n**\n * Returns a hash code based on the contents of this array as if it is

```

```

[List].\n *\n@Deprecated(\nUse Kotlin compiler 1.4 to avoid deprecation
warning.\n)\n@SinceKotlin(\n1.1\n)\n@DeprecatedSinceKotlin(hiddenSince = \n1.4\n)\npublic expect fun
ByteArray.contentHashCode(): Int\n\n**\n * Returns a hash code based on the contents of this array as if it is
[List].\n *\n@Deprecated(\nUse Kotlin compiler 1.4 to avoid deprecation
warning.\n)\n@SinceKotlin(\n1.1\n)\n@DeprecatedSinceKotlin(hiddenSince = \n1.4\n)\npublic expect fun
ShortArray.contentHashCode(): Int\n\n**\n * Returns a hash code based on the contents of this array as if it is
[List].\n *\n@Deprecated(\nUse Kotlin compiler 1.4 to avoid deprecation
warning.\n)\n@SinceKotlin(\n1.1\n)\n@DeprecatedSinceKotlin(hiddenSince = \n1.4\n)\npublic expect fun
IntArray.contentHashCode(): Int\n\n**\n * Returns a hash code based on the contents of this array as if it is
[List].\n *\n@Deprecated(\nUse Kotlin compiler 1.4 to avoid deprecation
warning.\n)\n@SinceKotlin(\n1.1\n)\n@DeprecatedSinceKotlin(hiddenSince = \n1.4\n)\npublic expect fun
LongArray.contentHashCode(): Int\n\n**\n * Returns a hash code based on the contents of this array as if it is
[List].\n *\n@Deprecated(\nUse Kotlin compiler 1.4 to avoid deprecation
warning.\n)\n@SinceKotlin(\n1.1\n)\n@DeprecatedSinceKotlin(hiddenSince = \n1.4\n)\npublic expect fun
FloatArray.contentHashCode(): Int\n\n**\n * Returns a hash code based on the contents of this array as if it is
[List].\n *\n@Deprecated(\nUse Kotlin compiler 1.4 to avoid deprecation
warning.\n)\n@SinceKotlin(\n1.1\n)\n@DeprecatedSinceKotlin(hiddenSince = \n1.4\n)\npublic expect fun
DoubleArray.contentHashCode(): Int\n\n**\n * Returns a hash code based on the contents of this array as if it is
[List].\n *\n@Deprecated(\nUse Kotlin compiler 1.4 to avoid deprecation
warning.\n)\n@SinceKotlin(\n1.1\n)\n@DeprecatedSinceKotlin(hiddenSince = \n1.4\n)\npublic expect fun
BooleanArray.contentHashCode(): Int\n\n**\n * Returns a hash code based on the contents of this array as if it is
[List].\n *\n@Deprecated(\nUse Kotlin compiler 1.4 to avoid deprecation
warning.\n)\n@SinceKotlin(\n1.1\n)\n@DeprecatedSinceKotlin(hiddenSince = \n1.4\n)\npublic expect fun
CharArray.contentHashCode(): Int\n\n**\n * Returns a hash code based on the contents of this array as if it is
[List].\n *\n@SinceKotlin(\n1.4\n)\npublic expect fun <T> Array<out T>?.contentHashCode(): Int\n\n**\n *
Returns a hash code based on the contents of this array as if it is [List].\n *\n@SinceKotlin(\n1.4\n)\npublic expect
fun ByteArray?.contentHashCode(): Int\n\n**\n * Returns a hash code based on the contents of this array as if it is
[List].\n *\n@SinceKotlin(\n1.4\n)\npublic expect fun ShortArray?.contentHashCode(): Int\n\n**\n * Returns a
hash code based on the contents of this array as if it is [List].\n *\n@SinceKotlin(\n1.4\n)\npublic expect fun
IntArray?.contentHashCode(): Int\n\n**\n * Returns a hash code based on the contents of this array as if it is
[List].\n *\n@SinceKotlin(\n1.4\n)\npublic expect fun LongArray?.contentHashCode(): Int\n\n**\n * Returns a
hash code based on the contents of this array as if it is [List].\n *\n@SinceKotlin(\n1.4\n)\npublic expect fun
FloatArray?.contentHashCode(): Int\n\n**\n * Returns a hash code based on the contents of this array as if it is
[List].\n *\n@SinceKotlin(\n1.4\n)\npublic expect fun DoubleArray?.contentHashCode(): Int\n\n**\n * Returns a
hash code based on the contents of this array as if it is [List].\n *\n@SinceKotlin(\n1.4\n)\npublic expect fun
BooleanArray?.contentHashCode(): Int\n\n**\n * Returns a hash code based on the contents of this array as if it is
[List].\n *\n@SinceKotlin(\n1.4\n)\npublic expect fun CharArray?.contentHashCode(): Int\n\n**\n * Returns a
string representation of the contents of the specified array as if it is [List].\n *\n \n * @sample
samples.collections.Arrays.ContentOperations.contentToString\n *\n@Deprecated(\nUse Kotlin compiler 1.4 to
avoid deprecation warning.\n)\n@SinceKotlin(\n1.1\n)\n@DeprecatedSinceKotlin(hiddenSince = \n1.4\n)\npublic
expect fun <T> Array<out T>.contentToString(): String\n\n**\n * Returns a string representation of the contents of
the specified array as if it is [List].\n *\n \n * @sample
samples.collections.Arrays.ContentOperations.contentToString\n *\n@Deprecated(\nUse Kotlin compiler 1.4 to
avoid deprecation warning.\n)\n@SinceKotlin(\n1.1\n)\n@DeprecatedSinceKotlin(hiddenSince = \n1.4\n)\npublic
expect fun ByteArray.contentToString(): String\n\n**\n * Returns a string representation of the contents of the
specified array as if it is [List].\n *\n \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n
*\n@Deprecated(\nUse Kotlin compiler 1.4 to avoid deprecation
warning.\n)\n@SinceKotlin(\n1.1\n)\n@DeprecatedSinceKotlin(hiddenSince = \n1.4\n)\npublic expect fun

```

ShortArray.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n *\n @Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n @SinceKotlin("1.1")\n @DeprecatedSinceKotlin(hiddenSince = "1.4")\n public expect fun

IntArray.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n *\n @Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n @SinceKotlin("1.1")\n @DeprecatedSinceKotlin(hiddenSince = "1.4")\n public expect fun

LongArray.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n *\n @Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n @SinceKotlin("1.1")\n @DeprecatedSinceKotlin(hiddenSince = "1.4")\n public expect fun

FloatArray.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n *\n @Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n @SinceKotlin("1.1")\n @DeprecatedSinceKotlin(hiddenSince = "1.4")\n public expect fun

DoubleArray.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n *\n @Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n @SinceKotlin("1.1")\n @DeprecatedSinceKotlin(hiddenSince = "1.4")\n public expect fun

BooleanArray.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n *\n @Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n @SinceKotlin("1.1")\n @DeprecatedSinceKotlin(hiddenSince = "1.4")\n public expect fun

CharArray.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n *\n @SinceKotlin("1.4")\n public expect fun

<T> Array<out T>?.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n *\n @SinceKotlin("1.4")\n public expect fun

ByteArray?.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n *\n @SinceKotlin("1.4")\n public expect fun

ShortArray?.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n *\n @SinceKotlin("1.4")\n public expect fun

IntArray?.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n *\n @SinceKotlin("1.4")\n public expect fun

LongArray?.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n *\n @SinceKotlin("1.4")\n public expect fun

FloatArray?.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n *\n @SinceKotlin("1.4")\n public expect fun

DoubleArray?.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n *\n @SinceKotlin("1.4")\n public expect fun

BooleanArray?.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n *\n @SinceKotlin("1.4")\n public expect fun

CharArray?.contentToString(): String\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the

[destination] and even specify the subrange so that it overlaps with the destination range.

`@param destination` the array to copy to.

`@param destinationOffset` the position in the [destination] array to copy to, 0 by default.

`@param startIndex` the beginning (inclusive) of the subrange to copy, 0 by default.

`@param endIndex` the end (exclusive) of the subrange to copy, size of this array by default.

`@throws IndexOutOfBoundsException` or `[IllegalArgumentException]` when [startIndex] or [endIndex] is out of range of this array indices or when ``startIndex > endIndex``.

`@throws IndexOutOfBoundsException` when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],

or when that index is out of the [destination] array indices range.

`@return` the [destination] array.

```

*\/n@SinceKotlin("1.3")npublic expect fun <T> Array<out T>.copyInto(destination: Array<T>, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): Array<T>
n/n/**n * Copies this array or its subrange into the [destination] array and returns that array.
n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.
n * @param destination the array to copy to.
n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.
n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.
n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],
n * or when that index is out of the [destination] array indices range.
n * @return the [destination] array.
n *\/n@SinceKotlin("1.3")npublic expect fun ByteArray.copyInto(destination: ByteArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): ByteArray
n/n/**n * Copies this array or its subrange into the [destination] array and returns that array.
n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.
n * @param destination the array to copy to.
n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.
n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.
n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],
n * or when that index is out of the [destination] array indices range.
n * @return the [destination] array.
n *\/n@SinceKotlin("1.3")npublic expect fun ShortArray.copyInto(destination: ShortArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): ShortArray
n/n/**n * Copies this array or its subrange into the [destination] array and returns that array.
n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.
n * @param destination the array to copy to.
n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.
n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.
n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],
n * or when that index is out of the [destination] array indices range.
n * @return the [destination] array.
n *\/n@SinceKotlin("1.3")npublic expect fun IntArray.copyInto(destination: IntArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): IntArray
n/n/**n * Copies this array or its subrange into the [destination] array and returns that array.
n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.
n * @param destination the array to copy to.
n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.
n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.
n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array

```


Array<T>\n\n/**\n * Returns new array which is a copy of the original array.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.copyOfOf\n * \n\npublic expect fun ByteArray.copyOfOf(): ByteArray\n\n/**\n * Returns new array which is a copy of the original array.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.copyOfOf\n * \n\npublic expect fun ShortArray.copyOfOf(): ShortArray\n\n/**\n * Returns new array which is a copy of the original array.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.copyOfOf\n * \n\npublic expect fun IntArray.copyOfOf(): IntArray\n\n/**\n * Returns new array which is a copy of the original array.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.copyOfOf\n * \n\npublic expect fun LongArray.copyOfOf(): LongArray\n\n/**\n * Returns new array which is a copy of the original array.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.copyOfOf\n * \n\npublic expect fun FloatArray.copyOfOf(): FloatArray\n\n/**\n * Returns new array which is a copy of the original array.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.copyOfOf\n * \n\npublic expect fun DoubleArray.copyOfOf(): DoubleArray\n\n/**\n * Returns new array which is a copy of the original array.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.copyOfOf\n * \n\npublic expect fun BooleanArray.copyOfOf(): BooleanArray\n\n/**\n * Returns new array which is a copy of the original array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with zero values if necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n * \n\npublic expect fun ByteArray.copyOfOf(newSize: Int): ByteArray\n\n/**\n * Returns new array which is a copy of the original array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with zero values if necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n * \n\npublic expect fun ShortArray.copyOfOf(newSize: Int): ShortArray\n\n/**\n * Returns new array which is a copy of the original array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with zero values if necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n * \n\npublic expect fun IntArray.copyOfOf(newSize: Int): IntArray\n\n/**\n * Returns new array which is a copy of the original array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with zero values if necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n * \n\npublic expect fun LongArray.copyOfOf(newSize: Int): LongArray\n\n/**\n * Returns new array which is a copy of the original array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with zero values if necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n * \n\npublic expect fun FloatArray.copyOfOf(newSize: Int): FloatArray\n\n/**\n * Returns new array which is a copy of the original array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with zero values if necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n * \n\npublic expect fun DoubleArray.copyOfOf(newSize: Int): DoubleArray\n\n/**\n * Returns new array which is a copy of the original array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with

[fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws
 IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\npublic expect fun
 DoubleArray.copyOfRange(fromIndex: Int, toIndex: Int): DoubleArray\n\n/**\n * Returns a new array which is a
 copy of the specified range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to
 copy.\n * @param toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException
 if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws
 IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\npublic expect fun
 BooleanArray.copyOfRange(fromIndex: Int, toIndex: Int): BooleanArray\n\n/**\n * Returns a new array which is a
 copy of the specified range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to
 copy.\n * @param toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException
 if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws
 IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\npublic expect fun
 CharArray.copyOfRange(fromIndex: Int, toIndex: Int): CharArray\n\n/**\n * Fills this array or its subrange with the
 specified [element] value.\n * \n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * \n *
 @param toIndex the end of the range (exclusive) to fill, size of this array by default.\n * \n * @throws
 IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n *
 @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n *
 @SinceKotlin("1.3")\npublic expect fun
 <T> Array<T>.fill(element: T, fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Fills this array or
 its subrange with the specified [element] value.\n * \n * @param fromIndex the start of the range (inclusive) to fill,
 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by default.\n * \n *
 @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this
 array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n\n*
 @SinceKotlin("1.3")\npublic expect fun
 ByteArray.fill(element: Byte, fromIndex: Int = 0, toIndex: Int = size):
 Unit\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param fromIndex the
 start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of
 this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is
 greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n\n*
 @SinceKotlin("1.3")\npublic expect fun
 ShortArray.fill(element: Short, fromIndex: Int = 0, toIndex: Int =
 size): Unit\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param fromIndex
 the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill,
 size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex]
 is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than
 [toIndex].\n */\n\n*
 @SinceKotlin("1.3")\npublic expect fun
 IntArray.fill(element: Int, fromIndex: Int = 0, toIndex: Int
 = size): Unit\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param
 fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive)
 to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero
 or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater
 than [toIndex].\n */\n\n*
 @SinceKotlin("1.3")\npublic expect fun
 LongArray.fill(element: Long, fromIndex: Int = 0,
 toIndex: Int = size): Unit\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n *
 @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range
 (exclusive) to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less
 than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is
 greater than [toIndex].\n */\n\n*
 @SinceKotlin("1.3")\npublic expect fun
 FloatArray.fill(element: Float, fromIndex: Int = 0,
 toIndex: Int = size): Unit\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n *
 @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range
 (exclusive) to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less
 than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is
 greater than [toIndex].\n */\n\n*
 @SinceKotlin("1.3")\npublic expect fun
 DoubleArray.fill(element: Double,

```

fromIndex: Int = 0, toIndex: Int = size): Unit\n\n**\n * Fills this array or its subrange with the specified [element]
value.\n * \n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end
of the range (exclusive) to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if
[fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws
IllegalArgumentException if [fromIndex] is greater than [toIndex].\n *\n@SinceKotlin("1.3")\npublic expect fun
BooleanArray.fill(element: Boolean, fromIndex: Int = 0, toIndex: Int = size): Unit\n\n**\n * Fills this array or its
subrange with the specified [element] value.\n * \n * @param fromIndex the start of the range (inclusive) to fill, 0
by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by default.\n * \n *
@throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this
array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*\n@SinceKotlin("1.3")\npublic expect fun CharArray.fill(element: Char, fromIndex: Int = 0, toIndex: Int = size):
Unit\n\n**\n * Returns the range of valid indices for the array.\n *\npublic val <T> Array<out T>.indices:
IntRange\n    get() = IntRange(0, lastIndex)\n\n**\n * Returns the range of valid indices for the array.\n *\npublic
val ByteArray.indices: IntRange\n    get() = IntRange(0, lastIndex)\n\n**\n * Returns the range of valid indices for
the array.\n *\npublic val ShortArray.indices: IntRange\n    get() = IntRange(0, lastIndex)\n\n**\n * Returns the
range of valid indices for the array.\n *\npublic val IntArray.indices: IntRange\n    get() = IntRange(0,
lastIndex)\n\n**\n * Returns the range of valid indices for the array.\n *\npublic val LongArray.indices:
IntRange\n    get() = IntRange(0, lastIndex)\n\n**\n * Returns the range of valid indices for the array.\n *\npublic
val FloatArray.indices: IntRange\n    get() = IntRange(0, lastIndex)\n\n**\n * Returns the range of valid indices for
the array.\n *\npublic val DoubleArray.indices: IntRange\n    get() = IntRange(0, lastIndex)\n\n**\n * Returns the
range of valid indices for the array.\n *\npublic val BooleanArray.indices: IntRange\n    get() = IntRange(0,
lastIndex)\n\n**\n * Returns the range of valid indices for the array.\n *\npublic val CharArray.indices: IntRange\n
    get() = IntRange(0, lastIndex)\n\n**\n * Returns `true` if the array is empty.\n
*\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.isEmpty(): Boolean {\n    return size ==
0\n}\n\n**\n * Returns `true` if the array is empty.\n *\n@kotlin.internal.InlineOnly\npublic inline fun
ByteArray.isEmpty(): Boolean {\n    return size == 0\n}\n\n**\n * Returns `true` if the array is empty.\n
*\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.isEmpty(): Boolean {\n    return size == 0\n}\n\n**\n
* Returns `true` if the array is empty.\n *\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.isEmpty():
Boolean {\n    return size == 0\n}\n\n**\n * Returns `true` if the array is empty.\n
*\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.isEmpty(): Boolean {\n    return size == 0\n}\n\n**\n
* Returns `true` if the array is empty.\n *\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.isEmpty():
Boolean {\n    return size == 0\n}\n\n**\n * Returns `true` if the array is empty.\n
*\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.isEmpty(): Boolean {\n    return size ==
0\n}\n\n**\n * Returns `true` if the array is empty.\n *\n@kotlin.internal.InlineOnly\npublic inline fun
BooleanArray.isEmpty(): Boolean {\n    return size == 0\n}\n\n**\n * Returns `true` if the array is empty.\n
*\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.isEmpty(): Boolean {\n    return size == 0\n}\n\n**\n
* Returns `true` if the array is not empty.\n *\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out
T>.isNotEmpty(): Boolean {\n    return !isEmpty()\n}\n\n**\n * Returns `true` if the array is not empty.\n
*\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.isNotEmpty(): Boolean {\n    return
!isEmpty()\n}\n\n**\n * Returns `true` if the array is not empty.\n *\n@kotlin.internal.InlineOnly\npublic inline
fun ShortArray.isNotEmpty(): Boolean {\n    return !isEmpty()\n}\n\n**\n * Returns `true` if the array is not
empty.\n *\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.isNotEmpty(): Boolean {\n    return
!isEmpty()\n}\n\n**\n * Returns `true` if the array is not empty.\n *\n@kotlin.internal.InlineOnly\npublic inline
fun LongArray.isNotEmpty(): Boolean {\n    return !isEmpty()\n}\n\n**\n * Returns `true` if the array is not
empty.\n *\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.isNotEmpty(): Boolean {\n    return
!isEmpty()\n}\n\n**\n * Returns `true` if the array is not empty.\n *\n@kotlin.internal.InlineOnly\npublic inline
fun DoubleArray.isNotEmpty(): Boolean {\n    return !isEmpty()\n}\n\n**\n * Returns `true` if the array is not
empty.\n *\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.isNotEmpty(): Boolean {\n    return

```

```

isEmpty()\n\n**\n * Returns `true` if the array is not empty.\n *\n@kotlin.internal.InlineOnly\npublic inline
fun CharArray.isNotEmpty(): Boolean {\n    return !isEmpty()\n}\n\n**\n * Returns the last valid index for the
array.\n *\npublic val <T> Array<out T>.lastIndex: Int\n    get() = size - 1\n\n**\n * Returns the last valid index
for the array.\n *\npublic val ByteArray.lastIndex: Int\n    get() = size - 1\n\n**\n * Returns the last valid index for
the array.\n *\npublic val ShortArray.lastIndex: Int\n    get() = size - 1\n\n**\n * Returns the last valid index for the
array.\n *\npublic val IntArray.lastIndex: Int\n    get() = size - 1\n\n**\n * Returns the last valid index for the
array.\n *\npublic val LongArray.lastIndex: Int\n    get() = size - 1\n\n**\n * Returns the last valid index for the
array.\n *\npublic val FloatArray.lastIndex: Int\n    get() = size - 1\n\n**\n * Returns the last valid index for the
array.\n *\npublic val DoubleArray.lastIndex: Int\n    get() = size - 1\n\n**\n * Returns the last valid index for the
array.\n *\npublic val BooleanArray.lastIndex: Int\n    get() = size - 1\n\n**\n * Returns the last valid index for the
array.\n *\npublic val CharArray.lastIndex: Int\n    get() = size - 1\n\n**\n * Returns an array containing all
elements of the original array and then the given [element].\n
*\n@Suppress("NO_ACTUAL_FOR_EXPECT")\npublic expect operator fun <T> Array<T>.plus(element: T):
Array<T>\n\n**\n * Returns an array containing all elements of the original array and then the given [element].\n
*\npublic expect operator fun ByteArray.plus(element: Byte): ByteArray\n\n**\n * Returns an array containing all
elements of the original array and then the given [element].\n *\npublic expect operator fun
ShortArray.plus(element: Short): ShortArray\n\n**\n * Returns an array containing all elements of the original
array and then the given [element].\n *\npublic expect operator fun IntArray.plus(element: Int): IntArray\n\n**\n *
Returns an array containing all elements of the original array and then the given [element].\n *\npublic expect
operator fun LongArray.plus(element: Long): LongArray\n\n**\n * Returns an array containing all elements of the
original array and then the given [element].\n *\npublic expect operator fun FloatArray.plus(element: Float):
FloatArray\n\n**\n * Returns an array containing all elements of the original array and then the given [element].\n
*\npublic expect operator fun DoubleArray.plus(element: Double): DoubleArray\n\n**\n * Returns an array
containing all elements of the original array and then the given [element].\n *\npublic expect operator fun
BooleanArray.plus(element: Boolean): BooleanArray\n\n**\n * Returns an array containing all elements of the
original array and then the given [element].\n *\npublic expect operator fun CharArray.plus(element: Char):
CharArray\n\n**\n * Returns an array containing all elements of the original array and then all elements of the
given [elements] collection.\n *\n@Suppress("NO_ACTUAL_FOR_EXPECT")\npublic expect operator fun <T>
Array<T>.plus(elements: Collection<T>): Array<T>\n\n**\n * Returns an array containing all elements of the
original array and then all elements of the given [elements] collection.\n *\npublic expect operator fun
ByteArray.plus(elements: Collection<Byte>): ByteArray\n\n**\n * Returns an array containing all elements of the
original array and then all elements of the given [elements] collection.\n *\npublic expect operator fun
ShortArray.plus(elements: Collection<Short>): ShortArray\n\n**\n * Returns an array containing all elements of
the original array and then all elements of the given [elements] collection.\n *\npublic expect operator fun
IntArray.plus(elements: Collection<Int>): IntArray\n\n**\n * Returns an array containing all elements of the
original array and then all elements of the given [elements] collection.\n *\npublic expect operator fun
LongArray.plus(elements: Collection<Long>): LongArray\n\n**\n * Returns an array containing all elements of the
original array and then all elements of the given [elements] collection.\n *\npublic expect operator fun
FloatArray.plus(elements: Collection<Float>): FloatArray\n\n**\n * Returns an array containing all elements of the
original array and then all elements of the given [elements] collection.\n *\npublic expect operator fun
DoubleArray.plus(elements: Collection<Double>): DoubleArray\n\n**\n * Returns an array containing all elements
of the original array and then all elements of the given [elements] collection.\n *\npublic expect operator fun
BooleanArray.plus(elements: Collection<Boolean>): BooleanArray\n\n**\n * Returns an array containing all
elements of the original array and then all elements of the given [elements] collection.\n *\npublic expect operator
fun CharArray.plus(elements: Collection<Char>): CharArray\n\n**\n * Returns an array containing all elements of
the original array and then all elements of the given [elements] array.\n
*\n@Suppress("NO_ACTUAL_FOR_EXPECT")\npublic expect operator fun <T> Array<T>.plus(elements:
Array<out T>): Array<T>\n\n**\n * Returns an array containing all elements of the original array and then all

```

elements of the given [elements] array.\n */\npublic expect operator fun ByteArray.plus(elements: ByteArray): ByteArray\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\npublic expect operator fun ShortArray.plus(elements: ShortArray): ShortArray\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\npublic expect operator fun IntArray.plus(elements: IntArray): IntArray\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\npublic expect operator fun LongArray.plus(elements: LongArray): LongArray\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\npublic expect operator fun FloatArray.plus(elements: FloatArray): FloatArray\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\npublic expect operator fun DoubleArray.plus(elements: DoubleArray): DoubleArray\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\npublic expect operator fun BooleanArray.plus(elements: BooleanArray): BooleanArray\n\n/**\n * Returns an array containing all elements of the original array and then the given [element].\n */\n@Suppress("NO_ACTUAL_FOR_EXPECT")\npublic expect fun <T> Array<T>.plusElement(element: T): Array<T>\n\n/**\n * Sorts the array in-place.\n * \n * @sample samples.collections.Arrays.Sorting.sortArray\n */\npublic expect fun IntArray.sort(): Unit\n\n/**\n * Sorts the array in-place.\n * \n * @sample samples.collections.Arrays.Sorting.sortArray\n */\npublic expect fun LongArray.sort(): Unit\n\n/**\n * Sorts the array in-place.\n * \n * @sample samples.collections.Arrays.Sorting.sortArray\n */\npublic expect fun ByteArray.sort(): Unit\n\n/**\n * Sorts the array in-place.\n * \n * @sample samples.collections.Arrays.Sorting.sortArray\n */\npublic expect fun ShortArray.sort(): Unit\n\n/**\n * Sorts the array in-place.\n * \n * @sample samples.collections.Arrays.Sorting.sortArray\n */\npublic expect fun DoubleArray.sort(): Unit\n\n/**\n * Sorts the array in-place.\n * \n * @sample samples.collections.Arrays.Sorting.sortArray\n */\npublic expect fun FloatArray.sort(): Unit\n\n/**\n * Sorts the array in-place according to the natural order of its elements.\n * \n * The sort is `_stable_`. It means that equal elements preserve their order relative to each other after sorting.\n * \n * @sample samples.collections.Arrays.Sorting.sortArrayOfComparable\n */\npublic expect fun <T : Comparable<T>> Array<out T>.sort(): Unit\n\n/**\n * Sorts a range in the array in-place.\n * \n * The sort is `_stable_`. It means that equal elements preserve their order relative to each other after sorting.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample samples.collections.Arrays.Sorting.sortRangeOfArrayOfComparable\n */\n@SinceKotlin("1.4")\npublic expect fun <T : Comparable<T>> Array<out T>.sort(fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample samples.collections.Arrays.Sorting.sortRangeOfArray\n */\n@SinceKotlin("1.4")\npublic expect fun ByteArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample samples.collections.Arrays.Sorting.sortRangeOfArray\n */\n@SinceKotlin("1.4")\npublic expect fun ShortArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample samples.collections.Arrays.Sorting.sortRangeOfArray\n */\n@SinceKotlin("1.4")\npublic expect fun DoubleArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample samples.collections.Arrays.Sorting.sortRangeOfArray\n */\n@SinceKotlin("1.4")\npublic expect fun FloatArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample samples.collections.Arrays.Sorting.sortRangeOfArray\n */\n@SinceKotlin("1.4")\npublic expect fun BooleanArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample samples.collections.Arrays.Sorting.sortRangeOfArray\n */\n@SinceKotlin("1.4")\npublic expect fun CharArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample samples.collections.Arrays.Sorting.sortRangeOfArray\n */\n@SinceKotlin("1.4")\npublic expect fun CharArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit


```

fromIndex the start of the range (inclusive) to sort.\n * @param toIndex the end of the range (exclusive) to sort.\n *
\n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of
this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*\n@SinceKotlin("1.4")\npublic fun IntArray.sortDescending(fromIndex: Int, toIndex: Int): Unit {\n
sort(fromIndex, toIndex)\n    reverse(fromIndex, toIndex)\n}\n\n/**\n * Sorts elements of the array in the specified
range in-place.\n * The elements are sorted descending according to their natural sort order.\n * \n * @param
fromIndex the start of the range (inclusive) to sort.\n * @param toIndex the end of the range (exclusive) to sort.\n *
\n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of
this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*\n@SinceKotlin("1.4")\npublic fun LongArray.sortDescending(fromIndex: Int, toIndex: Int): Unit {\n
sort(fromIndex, toIndex)\n    reverse(fromIndex, toIndex)\n}\n\n/**\n * Sorts elements of the array in the specified
range in-place.\n * The elements are sorted descending according to their natural sort order.\n * \n * @param
fromIndex the start of the range (inclusive) to sort.\n * @param toIndex the end of the range (exclusive) to sort.\n *
\n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of
this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*\n@SinceKotlin("1.4")\npublic fun FloatArray.sortDescending(fromIndex: Int, toIndex: Int): Unit {\n
sort(fromIndex, toIndex)\n    reverse(fromIndex, toIndex)\n}\n\n/**\n * Sorts elements of the array in the specified
range in-place.\n * The elements are sorted descending according to their natural sort order.\n * \n * @param
fromIndex the start of the range (inclusive) to sort.\n * @param toIndex the end of the range (exclusive) to sort.\n *
\n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of
this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*\n@SinceKotlin("1.4")\npublic fun DoubleArray.sortDescending(fromIndex: Int, toIndex: Int): Unit {\n
sort(fromIndex, toIndex)\n    reverse(fromIndex, toIndex)\n}\n\n/**\n * Sorts elements of the array in the specified
range in-place.\n * The elements are sorted descending according to their natural sort order.\n * \n * @param
fromIndex the start of the range (inclusive) to sort.\n * @param toIndex the end of the range (exclusive) to sort.\n *
\n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of
this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*\n@SinceKotlin("1.4")\npublic fun CharArray.sortDescending(fromIndex: Int, toIndex: Int): Unit {\n
sort(fromIndex, toIndex)\n    reverse(fromIndex, toIndex)\n}\n\n/**\n * Sorts the array in-place according to the
order specified by the given [comparator].\n * \n * The sort is _stable_. It means that equal elements preserve their
order relative to each other after sorting.\n */\npublic expect fun <T> Array<out T>.sortWith(comparator:
Comparator<in T>): Unit\n\n/**\n * Sorts a range in the array in-place with the given [comparator].\n * \n * The
sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n * \n *
@param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range
(exclusive) to sort, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is
less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if
[fromIndex] is greater than [toIndex].\n */\npublic expect fun <T> Array<out T>.sortWith(comparator:
Comparator<in T>, fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Returns an array of Boolean containing
all of the elements of this generic array.\n */\npublic fun Array<out Boolean>.toBooleanArray(): BooleanArray {\n
return BooleanArray(size) { index -> this[index] }\n}\n\n/**\n * Returns an array of Byte containing all of the
elements of this generic array.\n */\npublic fun Array<out Byte>.toByteArray(): ByteArray {\n    return
ByteArray(size) { index -> this[index] }\n}\n\n/**\n * Returns an array of Char containing all of the elements of this
generic array.\n */\npublic fun Array<out Char>.toCharArray(): CharArray {\n    return CharArray(size) { index ->
this[index] }\n}\n\n/**\n * Returns an array of Double containing all of the elements of this generic array.\n
*\n */\npublic fun Array<out Double>.toDoubleArray(): DoubleArray {\n    return DoubleArray(size) { index ->
this[index] }\n}\n\n/**\n * Returns an array of Float containing all of the elements of this generic array.\n
*\n */\npublic fun Array<out Float>.toFloatArray(): FloatArray {\n    return FloatArray(size) { index -> this[index] }\n}\n\n/**\n *
Returns an array of Int containing all of the elements of this generic array.\n */\npublic fun Array<out

```



```

map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitives\n *^\npublic inline fun <K, V>
FloatArray.associate(transform: (Float) -> Pair<K, V>): Map<K, V> {\n    val capacity =
mapCapacity(size).coerceAtLeast(16)\n    return associateTo(LinkedHashMap<K, V>(capacity),
transform)\n}\n\n/**\n * Returns a [Map] containing key-value pairs provided by [transform] function\n * applied to
elements of the given array.\n * \n * If any of two pairs would have the same key the last one gets added to the
map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitives\n *^\npublic inline fun <K, V>
DoubleArray.associate(transform: (Double) -> Pair<K, V>): Map<K, V> {\n    val capacity =
mapCapacity(size).coerceAtLeast(16)\n    return associateTo(LinkedHashMap<K, V>(capacity),
transform)\n}\n\n/**\n * Returns a [Map] containing key-value pairs provided by [transform] function\n * applied to
elements of the given array.\n * \n * If any of two pairs would have the same key the last one gets added to the
map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitives\n *^\npublic inline fun <K, V>
BooleanArray.associate(transform: (Boolean) -> Pair<K, V>): Map<K, V> {\n    val capacity =
mapCapacity(size).coerceAtLeast(16)\n    return associateTo(LinkedHashMap<K, V>(capacity),
transform)\n}\n\n/**\n * Returns a [Map] containing key-value pairs provided by [transform] function\n * applied to
elements of the given array.\n * \n * If any of two pairs would have the same key the last one gets added to the
map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitives\n *^\npublic inline fun <K, V>
CharArray.associate(transform: (Char) -> Pair<K, V>): Map<K, V> {\n    val capacity =
mapCapacity(size).coerceAtLeast(16)\n    return associateTo(LinkedHashMap<K, V>(capacity),
transform)\n}\n\n/**\n * Returns a [Map] containing the elements from the given array indexed by the key\n *
returned from [keySelector] function applied to each element.\n * \n * If any two elements would have the same key
returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration
order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy\n *^\npublic inline fun <T, K>
Array<out T>.associateBy(keySelector: (T) -> K): Map<K, T> {\n    val capacity =
mapCapacity(size).coerceAtLeast(16)\n    return associateByTo(LinkedHashMap<K, T>(capacity),
keySelector)\n}\n\n/**\n * Returns a [Map] containing the elements from the given array indexed by the key\n *
returned from [keySelector] function applied to each element.\n * \n * If any two elements would have the same key
returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration
order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy\n *^\npublic inline fun <K>
ByteArray.associateBy(keySelector: (Byte) -> K): Map<K, Byte> {\n    val capacity =
mapCapacity(size).coerceAtLeast(16)\n    return associateByTo(LinkedHashMap<K, Byte>(capacity),
keySelector)\n}\n\n/**\n * Returns a [Map] containing the elements from the given array indexed by the key\n *
returned from [keySelector] function applied to each element.\n * \n * If any two elements would have the same key
returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration
order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy\n *^\npublic inline fun <K>
ShortArray.associateBy(keySelector: (Short) -> K): Map<K, Short> {\n    val capacity =
mapCapacity(size).coerceAtLeast(16)\n    return associateByTo(LinkedHashMap<K, Short>(capacity),
keySelector)\n}\n\n/**\n * Returns a [Map] containing the elements from the given array indexed by the key\n *
returned from [keySelector] function applied to each element.\n * \n * If any two elements would have the same key
returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration
order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy\n *^\npublic inline fun <K>

```

```

IntArray.associateBy(keySelector: (Int) -> K): Map<K, Int> {\n  val capacity =
mapCapacity(size).coerceAtLeast(16)\n  return associateByTo(LinkedHashMap<K, Int>(capacity),
keySelector)\n}\n\n/**\n * Returns a [Map] containing the elements from the given array indexed by the key\n *
returned from [keySelector] function applied to each element.\n * \n * If any two elements would have the same key
returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration
order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy\n */\npublic inline fun <K>
LongArray.associateBy(keySelector: (Long) -> K): Map<K, Long> {\n  val capacity =
mapCapacity(size).coerceAtLeast(16)\n  return associateByTo(LinkedHashMap<K, Long>(capacity),
keySelector)\n}\n\n/**\n * Returns a [Map] containing the elements from the given array indexed by the key\n *
returned from [keySelector] function applied to each element.\n * \n * If any two elements would have the same key
returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration
order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy\n */\npublic inline fun <K>
FloatArray.associateBy(keySelector: (Float) -> K): Map<K, Float> {\n  val capacity =
mapCapacity(size).coerceAtLeast(16)\n  return associateByTo(LinkedHashMap<K, Float>(capacity),
keySelector)\n}\n\n/**\n * Returns a [Map] containing the elements from the given array indexed by the key\n *
returned from [keySelector] function applied to each element.\n * \n * If any two elements would have the same key
returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration
order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy\n */\npublic inline fun <K>
DoubleArray.associateBy(keySelector: (Double) -> K): Map<K, Double> {\n  val capacity =
mapCapacity(size).coerceAtLeast(16)\n  return associateByTo(LinkedHashMap<K, Double>(capacity),
keySelector)\n}\n\n/**\n * Returns a [Map] containing the elements from the given array indexed by the key\n *
returned from [keySelector] function applied to each element.\n * \n * If any two elements would have the same key
returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration
order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy\n */\npublic inline fun <K>
BooleanArray.associateBy(keySelector: (Boolean) -> K): Map<K, Boolean> {\n  val capacity =
mapCapacity(size).coerceAtLeast(16)\n  return associateByTo(LinkedHashMap<K, Boolean>(capacity),
keySelector)\n}\n\n/**\n * Returns a [Map] containing the elements from the given array indexed by the key\n *
returned from [keySelector] function applied to each element.\n * \n * If any two elements would have the same key
returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration
order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy\n */\npublic inline fun <K>
CharArray.associateBy(keySelector: (Char) -> K): Map<K, Char> {\n  val capacity =
mapCapacity(size).coerceAtLeast(16)\n  return associateByTo(LinkedHashMap<K, Char>(capacity),
keySelector)\n}\n\n/**\n * Returns a [Map] containing the values provided by [valueTransform] and indexed by
[keySelector] functions applied to elements of the given array.\n * \n * If any two elements would have the same key
returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration
order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform\n */\npublic inline
fun <T, K, V> Array<out T>.associateBy(keySelector: (T) -> K, valueTransform: (T) -> V): Map<K, V> {\n  val
capacity = mapCapacity(size).coerceAtLeast(16)\n  return associateByTo(LinkedHashMap<K, V>(capacity),
keySelector, valueTransform)\n}\n\n/**\n * Returns a [Map] containing the values provided by [valueTransform]
and indexed by [keySelector] functions applied to elements of the given array.\n * \n * If any two elements would
have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves
the entry iteration order of the original array.\n * \n * @sample

```

```
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform\n *\npublic inline  
fun <K, V> ByteArray.associateBy(keySelector: (Byte) -> K, valueTransform: (Byte) -> V): Map<K, V> {\n    val  
    capacity = mapCapacity(size).coerceAtLeast(16)\n    return associateByTo(LinkedHashMap<K, V>(capacity),  
    keySelector, valueTransform)\n}\n\n/**\n * Returns a [Map] containing the values provided by [valueTransform]  
and indexed by [keySelector] functions applied to elements of the given array.\n * \n * If any two elements would  
have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves  
the entry iteration order of the original array.\n * \n * @sample
```

```
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform\n *\npublic inline  
fun <K, V> ShortArray.associateBy(keySelector: (Short) -> K, valueTransform: (Short) -> V): Map<K, V> {\n    val  
    capacity = mapCapacity(size).coerceAtLeast(16)\n    return associateByTo(LinkedHashMap<K, V>(capacity),  
    keySelector, valueTransform)\n}\n\n/**\n * Returns a [Map] containing the values provided by [valueTransform]  
and indexed by [keySelector] functions applied to elements of the given array.\n * \n * If any two elements would  
have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves  
the entry iteration order of the original array.\n * \n * @sample
```

```
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform\n *\npublic inline  
fun <K, V> IntArray.associateBy(keySelector: (Int) -> K, valueTransform: (Int) -> V): Map<K, V> {\n    val  
    capacity = mapCapacity(size).coerceAtLeast(16)\n    return associateByTo(LinkedHashMap<K, V>(capacity),  
    keySelector, valueTransform)\n}\n\n/**\n * Returns a [Map] containing the values provided by [valueTransform]  
and indexed by [keySelector] functions applied to elements of the given array.\n * \n * If any two elements would  
have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves  
the entry iteration order of the original array.\n * \n * @sample
```

```
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform\n *\npublic inline  
fun <K, V> LongArray.associateBy(keySelector: (Long) -> K, valueTransform: (Long) -> V): Map<K, V> {\n    val  
    capacity = mapCapacity(size).coerceAtLeast(16)\n    return associateByTo(LinkedHashMap<K, V>(capacity),  
    keySelector, valueTransform)\n}\n\n/**\n * Returns a [Map] containing the values provided by [valueTransform]  
and indexed by [keySelector] functions applied to elements of the given array.\n * \n * If any two elements would  
have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves  
the entry iteration order of the original array.\n * \n * @sample
```

```
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform\n *\npublic inline  
fun <K, V> FloatArray.associateBy(keySelector: (Float) -> K, valueTransform: (Float) -> V): Map<K, V> {\n    val  
    capacity = mapCapacity(size).coerceAtLeast(16)\n    return associateByTo(LinkedHashMap<K, V>(capacity),  
    keySelector, valueTransform)\n}\n\n/**\n * Returns a [Map] containing the values provided by [valueTransform]  
and indexed by [keySelector] functions applied to elements of the given array.\n * \n * If any two elements would  
have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves  
the entry iteration order of the original array.\n * \n * @sample
```

```
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform\n *\npublic inline  
fun <K, V> DoubleArray.associateBy(keySelector: (Double) -> K, valueTransform: (Double) -> V): Map<K, V>  
{\n    val  
    capacity = mapCapacity(size).coerceAtLeast(16)\n    return associateByTo(LinkedHashMap<K,  
    V>(capacity), keySelector, valueTransform)\n}\n\n/**\n * Returns a [Map] containing the values provided by  
[valueTransform] and indexed by [keySelector] functions applied to elements of the given array.\n * \n * If any two  
elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The  
returned map preserves the entry iteration order of the original array.\n * \n * @sample
```

```
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform\n *\npublic inline  
fun <K, V> BooleanArray.associateBy(keySelector: (Boolean) -> K, valueTransform: (Boolean) -> V): Map<K, V>  
{\n    val  
    capacity = mapCapacity(size).coerceAtLeast(16)\n    return associateByTo(LinkedHashMap<K,  
    V>(capacity), keySelector, valueTransform)\n}\n\n/**\n * Returns a [Map] containing the values provided by  
[valueTransform] and indexed by [keySelector] functions applied to elements of the given array.\n * \n * If any two  
elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The
```

```

returned map preserves the entry iteration order of the original array.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform\n * \n * public inline
fun <K, V> CharArray.associateBy(keySelector: (Char) -> K, valueTransform: (Char) -> V): Map<K, V> {\n  val
capacity = mapCapacity(size).coerceAtLeast(16)\n  return associateByTo(LinkedHashMap<K, V>(capacity),
keySelector, valueTransform)\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value
pairs,\n * where key is provided by the [keySelector] function applied to each element of the given array\n * and
value is the element itself.\n * \n * If any two elements would have the same key returned by [keySelector] the last
one gets added to the map.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo\n * \n * public inline fun <T, K, M :
MutableMap<in K, in T>> Array<out T>.associateByTo(destination: M, keySelector: (T) -> K): M {\n  for
(element in this) {\n    destination.put(keySelector(element), element)\n  }\n  return destination\n}\n\n/**\n *
Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the
[keySelector] function applied to each element of the given array\n * and value is the element itself.\n * \n * If any
two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n *
@sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo\n * \n * public inline fun <K,
M : MutableMap<in K, in Byte>> ByteArray.associateByTo(destination: M, keySelector: (Byte) -> K): M {\n  for
(element in this) {\n    destination.put(keySelector(element), element)\n  }\n  return destination\n}\n\n/**\n *
Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the
[keySelector] function applied to each element of the given array\n * and value is the element itself.\n * \n * If any
two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n *
@sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo\n * \n * public inline fun <K,
M : MutableMap<in K, in Short>> ShortArray.associateByTo(destination: M, keySelector: (Short) -> K): M {\n  for
(element in this) {\n    destination.put(keySelector(element), element)\n  }\n  return destination\n}\n\n/**\n *
Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the
[keySelector] function applied to each element of the given array\n * and value is the element itself.\n * \n * If any
two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n *
@sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo\n * \n * public inline fun <K,
M : MutableMap<in K, in Int>> IntArray.associateByTo(destination: M, keySelector: (Int) -> K): M {\n  for
(element in this) {\n    destination.put(keySelector(element), element)\n  }\n  return destination\n}\n\n/**\n *
Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the
[keySelector] function applied to each element of the given array\n * and value is the element itself.\n * \n * If any
two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n *
@sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo\n * \n * public inline fun <K,
M : MutableMap<in K, in Long>> LongArray.associateByTo(destination: M, keySelector: (Long) -> K): M {\n  for
(element in this) {\n    destination.put(keySelector(element), element)\n  }\n  return destination\n}\n\n/**\n *
Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the
[keySelector] function applied to each element of the given array\n * and value is the element itself.\n * \n * If any
two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n *
@sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo\n * \n * public inline fun <K,
M : MutableMap<in K, in Float>> FloatArray.associateByTo(destination: M, keySelector: (Float) -> K): M {\n  for
(element in this) {\n    destination.put(keySelector(element), element)\n  }\n  return destination\n}\n\n/**\n *
Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the
[keySelector] function applied to each element of the given array\n * and value is the element itself.\n * \n * If any
two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n *
@sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo\n * \n * public inline fun <K,
M : MutableMap<in K, in Double>> DoubleArray.associateByTo(destination: M, keySelector: (Double) -> K): M
{\n  for (element in this) {\n    destination.put(keySelector(element), element)\n  }\n  return
destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is

```

provided by the [keySelector] function applied to each element of the given array\n * and value is the element itself.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo\n */\npublic inline fun <K, M : MutableMap<in K, in Boolean>> BooleanArray.associateByTo(destination: M, keySelector: (Boolean) -> K): M {\n for (element in this) {\n destination.put(keySelector(element), element)\n }\n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function applied to each element of the given array\n * and value is the element itself.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo\n */\npublic inline fun <K, M : MutableMap<in K, in Char>> CharArray.associateByTo(destination: M, keySelector: (Char) -> K): M {\n for (element in this) {\n destination.put(keySelector(element), element)\n }\n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function and\n * and value is provided by the [valueTransform] function applied to elements of the given array.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform\n */\npublic inline fun <T, K, V, M : MutableMap<in K, in V>> Array<out T>.associateByTo(destination: M, keySelector: (T) -> K, valueTransform: (T) -> V): M {\n for (element in this) {\n destination.put(keySelector(element), valueTransform(element))\n }\n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function and\n * and value is provided by the [valueTransform] function applied to elements of the given array.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform\n */\npublic inline fun <K, V, M : MutableMap<in K, in V>> ByteArray.associateByTo(destination: M, keySelector: (Byte) -> K, valueTransform: (Byte) -> V): M {\n for (element in this) {\n destination.put(keySelector(element), valueTransform(element))\n }\n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function and\n * and value is provided by the [valueTransform] function applied to elements of the given array.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform\n */\npublic inline fun <K, V, M : MutableMap<in K, in V>> ShortArray.associateByTo(destination: M, keySelector: (Short) -> K, valueTransform: (Short) -> V): M {\n for (element in this) {\n destination.put(keySelector(element), valueTransform(element))\n }\n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function and\n * and value is provided by the [valueTransform] function applied to elements of the given array.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform\n */\npublic inline fun <K, V, M : MutableMap<in K, in V>> IntArray.associateByTo(destination: M, keySelector: (Int) -> K, valueTransform: (Int) -> V): M {\n for (element in this) {\n destination.put(keySelector(element), valueTransform(element))\n }\n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function and\n * and value is provided by the [valueTransform] function applied to elements of the given array.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform\n */\npublic inline fun <K, V, M : MutableMap<in K, in V>> LongArray.associateByTo(destination: M, keySelector: (Long) -> K, valueTransform: (Long) -> V): M {\n for (element in this) {\n destination.put(keySelector(element), valueTransform(element))\n }\n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function and\n * and value is provided by

the [valueTransform] function applied to elements of the given array.

* If any two elements would have the same key returned by [keySelector] the last one gets added to the map.

@sample

```

samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform
*/npublic
inline fun <K, V, M : MutableMap<in K, in V>> FloatArray.associateByTo(destination: M, keySelector: (Float) ->
K, valueTransform: (Float) -> V): M {
    for (element in this) {
        destination.put(keySelector(element),
valueTransform(element))
    }
    return destination
}
*/n**
* Populates and returns the [destination] mutable
map with key-value pairs,
* where key is provided by the [keySelector] function and
* and value is provided by
the [valueTransform] function applied to elements of the given array.
* If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
* @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform
*/npublic
inline fun <K, V, M : MutableMap<in K, in V>> DoubleArray.associateByTo(destination: M, keySelector:
(Double) -> K, valueTransform: (Double) -> V): M {
    for (element in this) {
        destination.put(keySelector(element), valueTransform(element))
    }
    return destination
}
*/n**
* Populates
and returns the [destination] mutable map with key-value pairs,
* where key is provided by the [keySelector]
function and
* and value is provided by the [valueTransform] function applied to elements of the given array.
* If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
* @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform
*/npublic
inline fun <K, V, M : MutableMap<in K, in V>> BooleanArray.associateByTo(destination: M,
keySelector: (Boolean) -> K, valueTransform: (Boolean) -> V): M {
    for (element in this) {
        destination.put(keySelector(element), valueTransform(element))
    }
    return destination
}
*/n**
* Populates
and returns the [destination] mutable map with key-value pairs,
* where key is provided by the [keySelector]
function and
* and value is provided by the [valueTransform] function applied to elements of the given array.
* If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
* @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform
*/npublic
inline fun <K, V, M : MutableMap<in K, in V>> CharArray.associateByTo(destination: M, keySelector:
(Char) -> K, valueTransform: (Char) -> V): M {
    for (element in this) {
        destination.put(keySelector(element), valueTransform(element))
    }
    return destination
}
*/n**
* Populates
and returns the [destination] mutable map with key-value pairs
* provided by [transform] function applied to each
element of the given array.
* If any of two pairs would have the same key the last one gets added to the
map.
* @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo
*/npublic
inline fun <T, K, V, M : MutableMap<in K, in V>> Array<out T>.associateTo(destination: M, transform: (T) ->
Pair<K, V>): M {
    for (element in this) {
        destination += transform(element)
    }
    return
destination
}
*/n**
* Populates and returns the [destination] mutable map with key-value pairs
* provided by
[transform] function applied to each element of the given array.
* If any of two pairs would have the same key
the last one gets added to the map.
* @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo
*/npublic
inline fun <K, V, M :
MutableMap<in K, in V>> ByteArray.associateTo(destination: M, transform: (Byte) -> Pair<K, V>): M {
    for
(element in this) {
        destination += transform(element)
    }
    return destination
}
*/n**
* Populates and
returns the [destination] mutable map with key-value pairs
* provided by [transform] function applied to each
element of the given array.
* If any of two pairs would have the same key the last one gets added to the
map.
* @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo
*/npublic
inline fun <K, V, M : MutableMap<in K, in V>> ShortArray.associateTo(destination: M, transform: (Short) ->
Pair<K, V>): M {
    for (element in this) {
        destination += transform(element)
    }
    return
destination
}
*/n**
* Populates and returns the [destination] mutable map with key-value pairs
* provided by
[transform] function applied to each element of the given array.
* If any of two pairs would have the same key
the last one gets added to the map.
* @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo
*/npublic
inline fun <K, V, M :
MutableMap<in K, in V>> IntArray.associateTo(destination: M, transform: (Int) -> Pair<K, V>): M {
    for

```

```

(element in this) { \n      destination += transform(element)\n    } \n      return destination\n}\n\n/**\n * Populates and
returns the [destination] mutable map with key-value pairs\n * provided by [transform] function applied to each
element of the given array.\n * \n * If any of two pairs would have the same key the last one gets added to the
map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo\n */\npublic
inline fun <K, V, M : MutableMap<in K, in V>> LongArray.associateTo(destination: M, transform: (Long) ->
Pair<K, V>): M { \n      for (element in this) { \n      destination += transform(element)\n    } \n      return
destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs\n * provided by
[transform] function applied to each element of the given array.\n * \n * If any of two pairs would have the same key
the last one gets added to the map.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo\n */\npublic inline fun <K, V, M :
MutableMap<in K, in V>> FloatArray.associateTo(destination: M, transform: (Float) -> Pair<K, V>): M { \n      for
(element in this) { \n      destination += transform(element)\n    } \n      return destination\n}\n\n/**\n * Populates and
returns the [destination] mutable map with key-value pairs\n * provided by [transform] function applied to each
element of the given array.\n * \n * If any of two pairs would have the same key the last one gets added to the
map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo\n */\npublic
inline fun <K, V, M : MutableMap<in K, in V>> DoubleArray.associateTo(destination: M, transform: (Double) ->
Pair<K, V>): M { \n      for (element in this) { \n      destination += transform(element)\n    } \n      return
destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs\n * provided by
[transform] function applied to each element of the given array.\n * \n * If any of two pairs would have the same key
the last one gets added to the map.\n * \n * @sample
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo\n */\npublic inline fun <K, V, M :
MutableMap<in K, in V>> BooleanArray.associateTo(destination: M, transform: (Boolean) -> Pair<K, V>): M { \n
for (element in this) { \n      destination += transform(element)\n    } \n      return destination\n}\n\n/**\n * Populates
and returns the [destination] mutable map with key-value pairs\n * provided by [transform] function applied to each
element of the given array.\n * \n * If any of two pairs would have the same key the last one gets added to the
map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo\n */\npublic
inline fun <K, V, M : MutableMap<in K, in V>> CharArray.associateTo(destination: M, transform: (Char) ->
Pair<K, V>): M { \n      for (element in this) { \n      destination += transform(element)\n    } \n      return
destination\n}\n\n/**\n * Returns a [Map] where keys are elements from the given array and values are\n * produced
by the [valueSelector] function applied to each element.\n * \n * If any two elements are equal, the last one gets
added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n *
@sample samples.collections.Collections.Transformations.associateWith\n */\n@SinceKotlin("1.4")\npublic inline
fun <K, V> Array<out K>.associateWith(valueSelector: (K) -> V): Map<K, V> { \n      val result =
LinkedHashMap<K, V>(mapCapacity(size).coerceAtLeast(16))\n      return associateWithTo(result,
valueSelector)\n}\n\n/**\n * Returns a [Map] where keys are elements from the given array and values are\n *
produced by the [valueSelector] function applied to each element.\n * \n * If any two elements are equal, the last one
gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n *
@sample samples.collections.Collections.Transformations.associateWith\n
*/\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <V>
ByteArray.associateWith(valueSelector: (Byte) -> V): Map<Byte, V> { \n      val result = LinkedHashMap<Byte,
V>(mapCapacity(size).coerceAtLeast(16))\n      return associateWithTo(result, valueSelector)\n}\n\n/**\n * Returns a
[Map] where keys are elements from the given array and values are\n * produced by the [valueSelector] function
applied to each element.\n * \n * If any two elements are equal, the last one gets added to the map.\n * \n * The
returned map preserves the entry iteration order of the original array.\n * \n * @sample
samples.collections.Collections.Transformations.associateWith\n
*/\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <V>
ShortArray.associateWith(valueSelector: (Short) -> V): Map<Short, V> { \n      val result = LinkedHashMap<Short,
V>(mapCapacity(size).coerceAtLeast(16))\n      return associateWithTo(result, valueSelector)\n}\n\n/**\n * Returns a

```

[Map] where keys are elements from the given array and values are produced by the [valueSelector] function applied to each element. If any two elements are equal, the last one gets added to the map. The returned map preserves the entry iteration order of the original array. @sample

```

samples.collections.Collections.Transformations.associateWith
*/n@SinceKotlin("1.4")n@kotlin.internal.InlineOnlynpublic inline fun <V>
IntArray.associateWith(valueSelector: (Int) -> V): Map<Int, V> {n    val result = LinkedHashMap<Int,
V>(mapCapacity(size).coerceAtLeast(16))n    return associateWithTo(result, valueSelector)n}n/n/**n * Returns a
[Map] where keys are elements from the given array and values are produced by the [valueSelector] function
applied to each element.n * n * If any two elements are equal, the last one gets added to the map.n * n * The
returned map preserves the entry iteration order of the original array.n * n * @sample
samples.collections.Collections.Transformations.associateWith
*/n@SinceKotlin("1.4")n@kotlin.internal.InlineOnlynpublic inline fun <V>
LongArray.associateWith(valueSelector: (Long) -> V): Map<Long, V> {n    val result = LinkedHashMap<Long,
V>(mapCapacity(size).coerceAtLeast(16))n    return associateWithTo(result, valueSelector)n}n/n/**n * Returns a
[Map] where keys are elements from the given array and values are produced by the [valueSelector] function
applied to each element.n * n * If any two elements are equal, the last one gets added to the map.n * n * The
returned map preserves the entry iteration order of the original array.n * n * @sample
samples.collections.Collections.Transformations.associateWith
*/n@SinceKotlin("1.4")n@kotlin.internal.InlineOnlynpublic inline fun <V>
FloatArray.associateWith(valueSelector: (Float) -> V): Map<Float, V> {n    val result = LinkedHashMap<Float,
V>(mapCapacity(size).coerceAtLeast(16))n    return associateWithTo(result, valueSelector)n}n/n/**n * Returns a
[Map] where keys are elements from the given array and values are produced by the [valueSelector] function
applied to each element.n * n * If any two elements are equal, the last one gets added to the map.n * n * The
returned map preserves the entry iteration order of the original array.n * n * @sample
samples.collections.Collections.Transformations.associateWith
*/n@SinceKotlin("1.4")n@kotlin.internal.InlineOnlynpublic inline fun <V>
DoubleArray.associateWith(valueSelector: (Double) -> V): Map<Double, V> {n    val result =
LinkedHashMap<Double, V>(mapCapacity(size).coerceAtLeast(16))n    return associateWithTo(result,
valueSelector)n}n/n/**n * Returns a [Map] where keys are elements from the given array and values are
produced by the [valueSelector] function applied to each element.n * n * If any two elements are equal, the last one
gets added to the map.n * n * The returned map preserves the entry iteration order of the original array.n * n *
@sample samples.collections.Collections.Transformations.associateWith
*/n@SinceKotlin("1.4")n@kotlin.internal.InlineOnlynpublic inline fun <V>
BooleanArray.associateWith(valueSelector: (Boolean) -> V): Map<Boolean, V> {n    val result =
LinkedHashMap<Boolean, V>(mapCapacity(size).coerceAtLeast(16))n    return associateWithTo(result,
valueSelector)n}n/n/**n * Returns a [Map] where keys are elements from the given array and values are
produced by the [valueSelector] function applied to each element.n * n * If any two elements are equal, the last one
gets added to the map.n * n * The returned map preserves the entry iteration order of the original array.n * n *
@sample samples.collections.Collections.Transformations.associateWith
*/n@SinceKotlin("1.4")n@kotlin.internal.InlineOnlynpublic inline fun <V>
CharArray.associateWith(valueSelector: (Char) -> V): Map<Char, V> {n    val result = LinkedHashMap<Char,
V>(mapCapacity(size).coerceAtMost(128)).coerceAtLeast(16))n    return associateWithTo(result,
valueSelector)n}n/n/**n * Populates and returns the [destination] mutable map with key-value pairs for each
element of the given array,n * where key is the element itself and value is provided by the [valueSelector] function
applied to that key.n * n * If any two elements are equal, the last one overwrites the former value in the map.n * n
* @sample samples.collections.Collections.Transformations.associateWithTo
*/n@SinceKotlin("1.4")n@kotlin.internal.InlineOnlynpublic
inline fun <K, V, M : MutableMap<in K, in V>> Array<out K>.associateWithTo(destination: M, valueSelector: (K)
-> V): M {n    for (element in this) {n        destination.put(element, valueSelector(element))n    }n    return

```

```

destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs for each element
of the given array,\n * where key is the element itself and value is provided by the [valueSelector] function applied
to that key.\n * \n * If any two elements are equal, the last one overwrites the former value in the map.\n * \n *
@sample samples.collections.Collections.Transformations.associateWithTo\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <V, M : MutableMap<in Byte, in V>>
ByteArray.associateWithTo(destination: M, valueSelector: (Byte) -> V): M {\n  for (element in this) {\n
destination.put(element, valueSelector(element))\n  }\n  return destination\n}\n\n/**\n * Populates and returns the
[destination] mutable map with key-value pairs for each element of the given array,\n * where key is the element
itself and value is provided by the [valueSelector] function applied to that key.\n * \n * If any two elements are
equal, the last one overwrites the former value in the map.\n * \n * @sample
samples.collections.Collections.Transformations.associateWithTo\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <V, M : MutableMap<in Short, in V>>
ShortArray.associateWithTo(destination: M, valueSelector: (Short) -> V): M {\n  for (element in this) {\n
destination.put(element, valueSelector(element))\n  }\n  return destination\n}\n\n/**\n * Populates and returns the
[destination] mutable map with key-value pairs for each element of the given array,\n * where key is the element
itself and value is provided by the [valueSelector] function applied to that key.\n * \n * If any two elements are
equal, the last one overwrites the former value in the map.\n * \n * @sample
samples.collections.Collections.Transformations.associateWithTo\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <V, M : MutableMap<in Int, in V>>
IntArray.associateWithTo(destination: M, valueSelector: (Int) -> V): M {\n  for (element in this) {\n
destination.put(element, valueSelector(element))\n  }\n  return destination\n}\n\n/**\n * Populates and returns the
[destination] mutable map with key-value pairs for each element of the given array,\n * where key is the element
itself and value is provided by the [valueSelector] function applied to that key.\n * \n * If any two elements are
equal, the last one overwrites the former value in the map.\n * \n * @sample
samples.collections.Collections.Transformations.associateWithTo\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <V, M : MutableMap<in Long, in V>>
LongArray.associateWithTo(destination: M, valueSelector: (Long) -> V): M {\n  for (element in this) {\n
destination.put(element, valueSelector(element))\n  }\n  return destination\n}\n\n/**\n * Populates and returns the
[destination] mutable map with key-value pairs for each element of the given array,\n * where key is the element
itself and value is provided by the [valueSelector] function applied to that key.\n * \n * If any two elements are
equal, the last one overwrites the former value in the map.\n * \n * @sample
samples.collections.Collections.Transformations.associateWithTo\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <V, M : MutableMap<in Float, in V>>
FloatArray.associateWithTo(destination: M, valueSelector: (Float) -> V): M {\n  for (element in this) {\n
destination.put(element, valueSelector(element))\n  }\n  return destination\n}\n\n/**\n * Populates and returns the
[destination] mutable map with key-value pairs for each element of the given array,\n * where key is the element
itself and value is provided by the [valueSelector] function applied to that key.\n * \n * If any two elements are
equal, the last one overwrites the former value in the map.\n * \n * @sample
samples.collections.Collections.Transformations.associateWithTo\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <V, M : MutableMap<in Double, in V>>
DoubleArray.associateWithTo(destination: M, valueSelector: (Double) -> V): M {\n  for (element in this) {\n
destination.put(element, valueSelector(element))\n  }\n  return destination\n}\n\n/**\n * Populates and returns the
[destination] mutable map with key-value pairs for each element of the given array,\n * where key is the element
itself and value is provided by the [valueSelector] function applied to that key.\n * \n * If any two elements are
equal, the last one overwrites the former value in the map.\n * \n * @sample
samples.collections.Collections.Transformations.associateWithTo\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <V, M : MutableMap<in Boolean, in
V>> BooleanArray.associateWithTo(destination: M, valueSelector: (Boolean) -> V): M {\n  for (element in this)

```

```

    { destination.put(element, valueSelector(element)) } return destination }
}

* Populates and returns the [destination] mutable map with key-value pairs for each element of the given array, where key is the element itself and value is provided by the [valueSelector] function applied to that key. If any two elements are equal, the last one overwrites the former value in the map.

@sample
samples.collections.Collections.Transformations.associateWithTo

* Since Kotlin 1.4
@kotlin.internal.InlineOnly
public inline fun <V, M : MutableMap<in Char, in V>>
CharArray.associateWithTo(destination: M, valueSelector: (Char) -> V): M {
    for (element in this) {
        destination.put(element, valueSelector(element))
    }
    return destination
}

* Appends all elements to the given [destination] collection.
public fun <T, C : MutableCollection<in T>> Array<out T>.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

* Appends all elements to the given [destination] collection.
public fun <C : MutableCollection<in Byte>> ByteArray.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

* Appends all elements to the given [destination] collection.
public fun <C : MutableCollection<in Short>> ShortArray.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

* Appends all elements to the given [destination] collection.
public fun <C : MutableCollection<in Int>> IntArray.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

* Appends all elements to the given [destination] collection.
public fun <C : MutableCollection<in Long>> LongArray.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

* Appends all elements to the given [destination] collection.
public fun <C : MutableCollection<in Float>> FloatArray.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

* Appends all elements to the given [destination] collection.
public fun <C : MutableCollection<in Double>> DoubleArray.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

* Appends all elements to the given [destination] collection.
public fun <C : MutableCollection<in Boolean>> BooleanArray.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

* Appends all elements to the given [destination] collection.
public fun <C : MutableCollection<in Char>> CharArray.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

* Returns a new [HashSet] of all elements.
public fun <T> Array<out T>.toHashSet(): HashSet<T> {
    return toCollection(HashSet<T>(mapCapacity(size)))
}

* Returns a new [HashSet] of all elements.
public fun ByteArray.toHashSet(): HashSet<Byte> {
    return toCollection(HashSet<Byte>(mapCapacity(size)))
}

* Returns a new [HashSet] of all elements.
public fun ShortArray.toHashSet(): HashSet<Short> {
    return toCollection(HashSet<Short>(mapCapacity(size)))
}

* Returns a new [HashSet] of all elements.
public fun IntArray.toHashSet(): HashSet<Int> {
    return toCollection(HashSet<Int>(mapCapacity(size)))
}

* Returns a new [HashSet] of all elements.
public fun LongArray.toHashSet(): HashSet<Long> {
    return toCollection(HashSet<Long>(mapCapacity(size)))
}

* Returns a new [HashSet] of all elements.
public fun FloatArray.toHashSet(): HashSet<Float> {
    return toCollection(HashSet<Float>(mapCapacity(size)))
}

* Returns a new [HashSet] of all elements.
public fun DoubleArray.toHashSet(): HashSet<Double> {
    return toCollection(HashSet<Double>(mapCapacity(size)))
}

* Returns a new [HashSet] of all elements.
public fun BooleanArray.toHashSet(): HashSet<Boolean> {
    return toCollection(HashSet<Boolean>(mapCapacity(size)))
}

* Returns a new [HashSet] of all elements.
public fun CharArray.toHashSet(): HashSet<Char> {
    return toCollection(HashSet<Char>(mapCapacity(size.coerceAtMost(128))))
}

* Returns a [List] containing all elements.
public fun <T> Array<out T>.toList(): List<T> {
    return when (size) {
        0 -> emptyList()
    }
}

```



```

The returned set preserves the element iteration order of the original array.\n */\npublic fun LongArray.toSet():
Set<Long> {\n    return when (size) {\n        0 -> emptySet()\n        1 -> setOf(this[0])\n        else ->
toCollection(LinkedHashSet<Long>(mapCapacity(size)))\n    }\n}\n\n/**\n * Returns a [Set] of all elements.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic fun FloatArray.toSet():
Set<Float> {\n    return when (size) {\n        0 -> emptySet()\n        1 -> setOf(this[0])\n        else ->
toCollection(LinkedHashSet<Float>(mapCapacity(size)))\n    }\n}\n\n/**\n * Returns a [Set] of all elements.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic fun DoubleArray.toSet():
Set<Double> {\n    return when (size) {\n        0 -> emptySet()\n        1 -> setOf(this[0])\n        else ->
toCollection(LinkedHashSet<Double>(mapCapacity(size)))\n    }\n}\n\n/**\n * Returns a [Set] of all elements.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic fun
BooleanArray.toSet(): Set<Boolean> {\n    return when (size) {\n        0 -> emptySet()\n        1 -> setOf(this[0])\n        else -> toCollection(LinkedHashSet<Boolean>(mapCapacity(size)))\n    }\n}\n\n/**\n * Returns a [Set] of all
elements.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic fun
CharArray.toSet(): Set<Char> {\n    return when (size) {\n        0 -> emptySet()\n        1 -> setOf(this[0])\n        else -
> toCollection(LinkedHashSet<Char>(mapCapacity(size.coerceAtMost(128))))\n    }\n}\n\n/**\n * Returns a single
list of all elements yielded from results of [transform] function being invoked on each element of original array.\n *
\n * @sample samples.collections.Collections.Transformations.flatMap\n */\npublic inline fun <T, R> Array<out
T>.flatMap(transform: (T) -> Iterable<R>): List<R> {\n    return flatMapTo(ArrayList<R>(), transform)\n}\n\n/**\n *
Returns a single list of all elements yielded from results of [transform] function being invoked on each element of
original array.\n * \n * @sample samples.collections.Collections.Transformations.flatMap\n */\npublic inline fun
<R> ByteArray.flatMap(transform: (Byte) -> Iterable<R>): List<R> {\n    return flatMapTo(ArrayList<R>(),
transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being
invoked on each element of original array.\n * \n * @sample
samples.collections.Collections.Transformations.flatMap\n */\npublic inline fun <R> ShortArray.flatMap(transform:
(Short) -> Iterable<R>): List<R> {\n    return flatMapTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single
list of all elements yielded from results of [transform] function being invoked on each element of original array.\n *
\n * @sample samples.collections.Collections.Transformations.flatMap\n */\npublic inline fun <R>
IntArray.flatMap(transform: (Int) -> Iterable<R>): List<R> {\n    return flatMapTo(ArrayList<R>(),
transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being
invoked on each element of original array.\n * \n * @sample
samples.collections.Collections.Transformations.flatMap\n */\npublic inline fun <R> LongArray.flatMap(transform:
(Long) -> Iterable<R>): List<R> {\n    return flatMapTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single
list of all elements yielded from results of [transform] function being invoked on each element of original array.\n *
\n * @sample samples.collections.Collections.Transformations.flatMap\n */\npublic inline fun <R>
FloatArray.flatMap(transform: (Float) -> Iterable<R>): List<R> {\n    return flatMapTo(ArrayList<R>(),
transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being
invoked on each element of original array.\n * \n * @sample
samples.collections.Collections.Transformations.flatMap\n */\npublic inline fun <R>
DoubleArray.flatMap(transform: (Double) -> Iterable<R>): List<R> {\n    return flatMapTo(ArrayList<R>(),
transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being
invoked on each element of original array.\n * \n * @sample
samples.collections.Collections.Transformations.flatMap\n */\npublic inline fun <R>
BooleanArray.flatMap(transform: (Boolean) -> Iterable<R>): List<R> {\n    return flatMapTo(ArrayList<R>(),
transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being
invoked on each element of original array.\n * \n * @sample
samples.collections.Collections.Transformations.flatMap\n */\npublic inline fun <R> CharArray.flatMap(transform:
(Char) -> Iterable<R>): List<R> {\n    return flatMapTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single
list of all elements yielded from results of [transform] function being invoked on each element of original array.\n *

```

```

\n * @sample samples.collections.Collections.Transformations.flatMap\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapSequence")\npublic inline fun <T, R> Array<out
T>.flatMap(transform: (T) -> Sequence<R>): List<R> {\n    return flatMapTo(ArrayList<R>(),
transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being
invoked on each element\n * and its index in the original array.\n * \n * @sample
samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterable")\n@kotlin.internal.InlineOnly\npublic
inline fun <T, R> Array<out T>.flatMapIndexed(transform: (index: Int, T) -> Iterable<R>): List<R> {\n    return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n *
@sample samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterable")\n@kotlin.internal.InlineOnly\npublic
inline fun <R> ByteArray.flatMapIndexed(transform: (index: Int, Byte) -> Iterable<R>): List<R> {\n    return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n *
@sample samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterable")\n@kotlin.internal.InlineOnly\npublic
inline fun <R> ShortArray.flatMapIndexed(transform: (index: Int, Short) -> Iterable<R>): List<R> {\n    return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n *
@sample samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterable")\n@kotlin.internal.InlineOnly\npublic
inline fun <R> IntArray.flatMapIndexed(transform: (index: Int, Int) -> Iterable<R>): List<R> {\n    return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n *
@sample samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterable")\n@kotlin.internal.InlineOnly\npublic
inline fun <R> LongArray.flatMapIndexed(transform: (index: Int, Long) -> Iterable<R>): List<R> {\n    return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n *
@sample samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterable")\n@kotlin.internal.InlineOnly\npublic
inline fun <R> FloatArray.flatMapIndexed(transform: (index: Int, Float) -> Iterable<R>): List<R> {\n    return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n *
@sample samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterable")\n@kotlin.internal.InlineOnly\npublic
inline fun <R> DoubleArray.flatMapIndexed(transform: (index: Int, Double) -> Iterable<R>): List<R> {\n    return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n *

```

```

@sample samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterable")\n@kotlin.internal.InlineOnly\npublic
inline fun <R> BooleanArray.flatMapIndexed(transform: (index: Int, Boolean) -> Iterable<R>): List<R> {\n
return flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded
from results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n *
@sample samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterable")\n@kotlin.internal.InlineOnly\npublic
inline fun <R> CharArray.flatMapIndexed(transform: (index: Int, Char) -> Iterable<R>): List<R> {\n
return flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n *
@sample samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedSequence")\n@kotlin.internal.InlineOnly\npubli
c inline fun <T, R> Array<out T>.flatMapIndexed(transform: (index: Int, T) -> Sequence<R>): List<R> {\n
return flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Appends all elements yielded from results of
[transform] function being invoked on each element\n * and its index in the original array, to the given
[destination].\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterableTo")\n@kotlin.internal.InlineOnly\npubli
c inline fun <T, R, C : MutableCollection<in R>> Array<out T>.flatMapIndexedTo(destination: C, transform:
(index: Int, T) -> Iterable<R>): C {\n
var index = 0\n
for (element in this) {\n
val list = transform(index++,
element)\n
destination.addAll(list)\n
}\n
return destination\n}\n\n/**\n * Appends all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array, to the given
[destination].\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterableTo")\n@kotlin.internal.InlineOnly\npubli
c inline fun <R, C : MutableCollection<in R>> ByteArray.flatMapIndexedTo(destination: C, transform: (index: Int,
Byte) -> Iterable<R>): C {\n
var index = 0\n
for (element in this) {\n
val list = transform(index++,
element)\n
destination.addAll(list)\n
}\n
return destination\n}\n\n/**\n * Appends all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array, to the given
[destination].\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterableTo")\n@kotlin.internal.InlineOnly\npubli
c inline fun <R, C : MutableCollection<in R>> ShortArray.flatMapIndexedTo(destination: C, transform: (index: Int,
Short) -> Iterable<R>): C {\n
var index = 0\n
for (element in this) {\n
val list = transform(index++,
element)\n
destination.addAll(list)\n
}\n
return destination\n}\n\n/**\n * Appends all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array, to the given
[destination].\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterableTo")\n@kotlin.internal.InlineOnly\npubli
c inline fun <R, C : MutableCollection<in R>> IntArray.flatMapIndexedTo(destination: C, transform: (index: Int,
Int) -> Iterable<R>): C {\n
var index = 0\n
for (element in this) {\n
val list = transform(index++, element)\n
destination.addAll(list)\n
}\n
return destination\n}\n\n/**\n * Appends all elements yielded from results of
[transform] function being invoked on each element\n * and its index in the original array, to the given
[destination].\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution

```

```
ByLambdaReturnType\n@kotlin.jvm.JvmName("\\flatMapIndexedIterableTo\\")\n@kotlin.internal.InlineOnly\npublic inline fun <R, C : MutableCollection<in R>> LongArray.flatMapIndexedTo(destination: C, transform: (index: Int, Long) -> Iterable<R>): C {\n    var index = 0\n    for (element in this) {\n        val list = transform(index++, element)\n        destination.addAll(list)\n    }\n    return destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each element\n * and its index in the original array, to the given [destination].\n */
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("\\flatMapIndexedIterableTo\\")\n@kotlin.internal.InlineOnly\npublic inline fun <R, C : MutableCollection<in R>> FloatArray.flatMapIndexedTo(destination: C, transform: (index: Int, Float) -> Iterable<R>): C {\n    var index = 0\n    for (element in this) {\n        val list = transform(index++, element)\n        destination.addAll(list)\n    }\n    return destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each element\n * and its index in the original array, to the given [destination].\n */
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("\\flatMapIndexedIterableTo\\")\n@kotlin.internal.InlineOnly\npublic inline fun <R, C : MutableCollection<in R>> DoubleArray.flatMapIndexedTo(destination: C, transform: (index: Int, Double) -> Iterable<R>): C {\n    var index = 0\n    for (element in this) {\n        val list = transform(index++, element)\n        destination.addAll(list)\n    }\n    return destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each element\n * and its index in the original array, to the given [destination].\n */
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("\\flatMapIndexedIterableTo\\")\n@kotlin.internal.InlineOnly\npublic inline fun <R, C : MutableCollection<in R>> BooleanArray.flatMapIndexedTo(destination: C, transform: (index: Int, Boolean) -> Iterable<R>): C {\n    var index = 0\n    for (element in this) {\n        val list = transform(index++, element)\n        destination.addAll(list)\n    }\n    return destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each element\n * and its index in the original array, to the given [destination].\n */
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("\\flatMapIndexedIterableTo\\")\n@kotlin.internal.InlineOnly\npublic inline fun <R, C : MutableCollection<in R>> CharArray.flatMapIndexedTo(destination: C, transform: (index: Int, Char) -> Iterable<R>): C {\n    var index = 0\n    for (element in this) {\n        val list = transform(index++, element)\n        destination.addAll(list)\n    }\n    return destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each element\n * and its index in the original array, to the given [destination].\n */
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("\\flatMapIndexedSequenceTo\\")\n@kotlin.internal.InlineOnly\npublic inline fun <T, R, C : MutableCollection<in R>> Array<out T>.flatMapIndexedTo(destination: C, transform: (index: Int, T) -> Sequence<R>): C {\n    var index = 0\n    for (element in this) {\n        val list = transform(index++, element)\n        destination.addAll(list)\n    }\n    return destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each element of original array, to the given [destination].\n */\n\npublic inline fun <T, R, C : MutableCollection<in R>> Array<out T>.flatMapTo(destination: C, transform: (T) -> Iterable<R>): C {\n    for (element in this) {\n        val list = transform(element)\n        destination.addAll(list)\n    }\n    return destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each element of original array, to the given [destination].\n */\n\npublic inline fun <R, C : MutableCollection<in R>> ByteArray.flatMapTo(destination: C, transform: (Byte) -> Iterable<R>): C {\n    for (element in this) {\n        val list = transform(element)\n        destination.addAll(list)\n    }\n    return destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each element of original array, to the given [destination].\n */\n\npublic inline fun <R, C : MutableCollection<in R>>
```

```

ShortArray.flatMapTo(destination: C, transform: (Short) -> Iterable<R>): C {
    for (element in this) {
        val list = transform(element)
        destination.addAll(list)
    }
    return destination
}
/**
 * Appends all elements yielded from results of [transform] function being invoked on each element of original array, to the given [destination].
 */
public inline fun <R, C : MutableCollection<in R>> IntArray.flatMapTo(destination: C, transform: (Int) -> Iterable<R>): C {
    for (element in this) {
        val list = transform(element)
        destination.addAll(list)
    }
    return destination
}
/**
 * Appends all elements yielded from results of [transform] function being invoked on each element of original array, to the given [destination].
 */
public inline fun <R, C : MutableCollection<in R>> LongArray.flatMapTo(destination: C, transform: (Long) -> Iterable<R>): C {
    for (element in this) {
        val list = transform(element)
        destination.addAll(list)
    }
    return destination
}
/**
 * Appends all elements yielded from results of [transform] function being invoked on each element of original array, to the given [destination].
 */
public inline fun <R, C : MutableCollection<in R>> FloatArray.flatMapTo(destination: C, transform: (Float) -> Iterable<R>): C {
    for (element in this) {
        val list = transform(element)
        destination.addAll(list)
    }
    return destination
}
/**
 * Appends all elements yielded from results of [transform] function being invoked on each element of original array, to the given [destination].
 */
public inline fun <R, C : MutableCollection<in R>> DoubleArray.flatMapTo(destination: C, transform: (Double) -> Iterable<R>): C {
    for (element in this) {
        val list = transform(element)
        destination.addAll(list)
    }
    return destination
}
/**
 * Appends all elements yielded from results of [transform] function being invoked on each element of original array, to the given [destination].
 */
public inline fun <R, C : MutableCollection<in R>> BooleanArray.flatMapTo(destination: C, transform: (Boolean) -> Iterable<R>): C {
    for (element in this) {
        val list = transform(element)
        destination.addAll(list)
    }
    return destination
}
/**
 * Appends all elements yielded from results of [transform] function being invoked on each element of original array, to the given [destination].
 */
public inline fun <R, C : MutableCollection<in R>> CharArray.flatMapTo(destination: C, transform: (Char) -> Iterable<R>): C {
    for (element in this) {
        val list = transform(element)
        destination.addAll(list)
    }
    return destination
}
/**
 * Appends all elements yielded from results of [transform] function being invoked on each element of original array, to the given [destination].
 */
}

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@kotlin.jvm.JvmName("flatMapSequenceTo")
public inline fun <T, R, C : MutableCollection<in R>> Array<out T>.flatMapTo(destination: C, transform: (T) -> Sequence<R>): C {
    for (element in this) {
        val list = transform(element)
        destination.addAll(list)
    }
    return destination
}
/**
 * Groups elements of the original array by the key returned by the given [keySelector] function
 * applied to each element and returns a map where each group key is associated with a list of corresponding elements.
 * The returned map preserves the entry iteration order of the keys produced from the original array.
 */
@sample samples.collections.Collections.Transformations.groupBy
public inline fun <T, K> Array<out T>.groupBy(keySelector: (T) -> K): Map<K, List<T>> {
    return groupByTo(LinkedHashMap<K, MutableList<T>>(), keySelector)
}
/**
 * Groups elements of the original array by the key returned by the given [keySelector] function
 * applied to each element and returns a map where each group key is associated with a list of corresponding elements.
 * The returned map preserves the entry iteration order of the keys produced from the original array.
 */
@sample samples.collections.Collections.Transformations.groupBy
public inline fun <K>
ByteArray.groupBy(keySelector: (Byte) -> K): Map<K, List<Byte>> {
    return groupByTo(LinkedHashMap<K, MutableList<Byte>>(), keySelector)
}
/**
 * Groups elements of the original array by the key returned by the given [keySelector] function
 * applied to each element and returns a map where each group key is associated with a list of corresponding elements.
 * The returned map preserves the entry iteration order of the keys produced from the original array.
 */
@sample samples.collections.Collections.Transformations.groupBy
public inline fun <K>
ShortArray.groupBy(keySelector: (Short) -> K): Map<K, List<Short>> {
    return groupByTo(LinkedHashMap<K, MutableList<Short>>(), keySelector)
}
/**
 * Groups elements of the original array by the key returned by the given [keySelector] function
 * applied to each element and returns a map

```

where each group key is associated with a list of corresponding elements.\n * \n * The returned map preserves the entry iteration order of the keys produced from the original array.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <K> IntArray.groupBy(keySelector: (Int) -> K): Map<K, List<Int>> {\n return groupByTo(LinkedHashMap<K, MutableList<Int>>(), keySelector)\n}\n\n/**\n * Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each element and returns a map where each group key is associated with a list of corresponding elements.\n * \n * The returned map preserves the entry iteration order of the keys produced from the original array.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <K> LongArray.groupBy(keySelector: (Long) -> K): Map<K, List<Long>> {\n return groupByTo(LinkedHashMap<K, MutableList<Long>>(), keySelector)\n}\n\n/**\n * Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each element and returns a map where each group key is associated with a list of corresponding elements.\n * \n * The returned map preserves the entry iteration order of the keys produced from the original array.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <K> FloatArray.groupBy(keySelector: (Float) -> K): Map<K, List<Float>> {\n return groupByTo(LinkedHashMap<K, MutableList<Float>>(), keySelector)\n}\n\n/**\n * Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each element and returns a map where each group key is associated with a list of corresponding elements.\n * \n * The returned map preserves the entry iteration order of the keys produced from the original array.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <K> DoubleArray.groupBy(keySelector: (Double) -> K): Map<K, List<Double>> {\n return groupByTo(LinkedHashMap<K, MutableList<Double>>(), keySelector)\n}\n\n/**\n * Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each element and returns a map where each group key is associated with a list of corresponding elements.\n * \n * The returned map preserves the entry iteration order of the keys produced from the original array.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <K> BooleanArray.groupBy(keySelector: (Boolean) -> K): Map<K, List<Boolean>> {\n return groupByTo(LinkedHashMap<K, MutableList<Boolean>>(), keySelector)\n}\n\n/**\n * Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each element and returns a map where each group key is associated with a list of corresponding elements.\n * \n * The returned map preserves the entry iteration order of the keys produced from the original array.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <K> CharArray.groupBy(keySelector: (Char) -> K): Map<K, List<Char>> {\n return groupByTo(LinkedHashMap<K, MutableList<Char>>(), keySelector)\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the original array\n * by the key returned by the given [keySelector] function applied to the element\n * and returns a map where each group key is associated with a list of corresponding values.\n * \n * The returned map preserves the entry iteration order of the keys produced from the original array.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeysAndValues\n */\npublic inline fun <T, K, V> Array<out T>.groupBy(keySelector: (T) -> K, valueTransform: (T) -> V): Map<K, List<V>> {\n return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the original array\n * by the key returned by the given [keySelector] function applied to the element\n * and returns a map where each group key is associated with a list of corresponding values.\n * \n * The returned map preserves the entry iteration order of the keys produced from the original array.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeysAndValues\n */\npublic inline fun <K, V> ByteArray.groupBy(keySelector: (Byte) -> K, valueTransform: (Byte) -> V): Map<K, List<V>> {\n return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the original array\n * by the key returned by the given [keySelector] function applied to the element\n * and returns a map where each group key is associated

with a list of corresponding values.\n * \n * The returned map preserves the entry iteration order of the keys produced from the original array.\n * \n * @sample

```

samples.collections.Collections.Transformations.groupByKeyAndValues\n * \npublic inline fun <K, V>
ShortArray.groupBy(keySelector: (Short) -> K, valueTransform: (Short) -> V): Map<K, List<V>> {\n return
groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n * Groups values
returned by the [valueTransform] function applied to each element of the original array\n * by the key returned by
the given [keySelector] function applied to the element\n * and returns a map where each group key is associated
with a list of corresponding values.\n * \n * The returned map preserves the entry iteration order of the keys
produced from the original array.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeyAndValues\n * \npublic inline fun <K, V>
IntArray.groupBy(keySelector: (Int) -> K, valueTransform: (Int) -> V): Map<K, List<V>> {\n return
groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n * Groups values
returned by the [valueTransform] function applied to each element of the original array\n * by the key returned by
the given [keySelector] function applied to the element\n * and returns a map where each group key is associated
with a list of corresponding values.\n * \n * The returned map preserves the entry iteration order of the keys
produced from the original array.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeyAndValues\n * \npublic inline fun <K, V>
LongArray.groupBy(keySelector: (Long) -> K, valueTransform: (Long) -> V): Map<K, List<V>> {\n return
groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n * Groups values
returned by the [valueTransform] function applied to each element of the original array\n * by the key returned by
the given [keySelector] function applied to the element\n * and returns a map where each group key is associated
with a list of corresponding values.\n * \n * The returned map preserves the entry iteration order of the keys
produced from the original array.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeyAndValues\n * \npublic inline fun <K, V>
FloatArray.groupBy(keySelector: (Float) -> K, valueTransform: (Float) -> V): Map<K, List<V>> {\n return
groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n * Groups values
returned by the [valueTransform] function applied to each element of the original array\n * by the key returned by
the given [keySelector] function applied to the element\n * and returns a map where each group key is associated
with a list of corresponding values.\n * \n * The returned map preserves the entry iteration order of the keys
produced from the original array.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeyAndValues\n * \npublic inline fun <K, V>
DoubleArray.groupBy(keySelector: (Double) -> K, valueTransform: (Double) -> V): Map<K, List<V>> {\n
return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n * Groups
values returned by the [valueTransform] function applied to each element of the original array\n * by the key
returned by the given [keySelector] function applied to the element\n * and returns a map where each group key is
associated with a list of corresponding values.\n * \n * The returned map preserves the entry iteration order of
the keys produced from the original array.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeyAndValues\n * \npublic inline fun <K, V>
BooleanArray.groupBy(keySelector: (Boolean) -> K, valueTransform: (Boolean) -> V): Map<K, List<V>> {\n
return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n * Groups
values returned by the [valueTransform] function applied to each element of the original array\n * by the key
returned by the given [keySelector] function applied to the element\n * and returns a map where each group key is
associated with a list of corresponding values.\n * \n * The returned map preserves the entry iteration order of
the keys produced from the original array.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeyAndValues\n * \npublic inline fun <K, V>
CharArray.groupBy(keySelector: (Char) -> K, valueTransform: (Char) -> V): Map<K, List<V>> {\n return
groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n * Groups elements
of the original array by the key returned by the given [keySelector] function\n * applied to each element and puts to

```

```

the [destination] map each group key associated with a list of corresponding elements.\n * \n * @return The
[destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n * \n * public inline
fun <T, K, M : MutableMap<in K, MutableList<T>>> Array<out T>.groupByTo(destination: M, keySelector: (T) -
-> K): M {\n for (element in this) {\n val key = keySelector(element)\n val list = destination.getOrPut(key)
{ ArrayList<T>() }\n list.add(element)\n }\n return destination}\n * \n * Groups elements of the
original array by the key returned by the given [keySelector] function\n * applied to each element and puts to the
[destination] map each group key associated with a list of corresponding elements.\n * \n * @return The
[destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n * \n * public inline
fun <K, M : MutableMap<in K, MutableList<Byte>>> ByteArray.groupByTo(destination: M, keySelector: (Byte) -
-> K): M {\n for (element in this) {\n val key = keySelector(element)\n val list = destination.getOrPut(key)
{ ArrayList<Byte>() }\n list.add(element)\n }\n return destination}\n * \n * Groups elements of the
original array by the key returned by the given [keySelector] function\n * applied to each element and puts to the
[destination] map each group key associated with a list of corresponding elements.\n * \n * @return The
[destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n * \n * public inline
fun <K, M : MutableMap<in K, MutableList<Short>>> ShortArray.groupByTo(destination: M, keySelector: (Short)
-> K): M {\n for (element in this) {\n val key = keySelector(element)\n val list =
destination.getOrPut(key) { ArrayList<Short>() }\n list.add(element)\n }\n return destination}\n * \n *
Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each
element and puts to the [destination] map each group key associated with a list of corresponding elements.\n * \n *
@return The [destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n
* \n * public inline fun <K, M : MutableMap<in K, MutableList<Int>>> IntArray.groupByTo(destination: M,
keySelector: (Int) -> K): M {\n for (element in this) {\n val key = keySelector(element)\n val list =
destination.getOrPut(key) { ArrayList<Int>() }\n list.add(element)\n }\n return destination}\n * \n *
Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each
element and puts to the [destination] map each group key associated with a list of corresponding elements.\n * \n *
@return The [destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n
* \n * public inline fun <K, M : MutableMap<in K, MutableList<Long>>> LongArray.groupByTo(destination: M,
keySelector: (Long) -> K): M {\n for (element in this) {\n val key = keySelector(element)\n val list =
destination.getOrPut(key) { ArrayList<Long>() }\n list.add(element)\n }\n return destination}\n * \n *
Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each
element and puts to the [destination] map each group key associated with a list of corresponding elements.\n * \n *
@return The [destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n
* \n * public inline fun <K, M : MutableMap<in K, MutableList<Float>>> FloatArray.groupByTo(destination: M,
keySelector: (Float) -> K): M {\n for (element in this) {\n val key = keySelector(element)\n val list =
destination.getOrPut(key) { ArrayList<Float>() }\n list.add(element)\n }\n return destination}\n * \n *
Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each
element and puts to the [destination] map each group key associated with a list of corresponding elements.\n * \n *
@return The [destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n
* \n * public inline fun <K, M : MutableMap<in K, MutableList<Double>>> DoubleArray.groupByTo(destination: M,
keySelector: (Double) -> K): M {\n for (element in this) {\n val key = keySelector(element)\n val list =
destination.getOrPut(key) { ArrayList<Double>() }\n list.add(element)\n }\n return destination}\n * \n *
Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each
element and puts to the [destination] map each group key associated with a list of corresponding elements.\n * \n *
@return The [destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n
* \n * public inline fun <K, M : MutableMap<in K, MutableList<Boolean>>> BooleanArray.groupByTo(destination:
M, keySelector: (Boolean) -> K): M {\n for (element in this) {\n val key = keySelector(element)\n val list
= destination.getOrPut(key) { ArrayList<Boolean>() }\n list.add(element)\n }\n return
destination}\n * \n * Groups elements of the original array by the key returned by the given [keySelector]

```



```

destination\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the
original array\n * by the key returned by the given [keySelector] function applied to the element\n * and puts to the
[destination] map each group key associated with a list of corresponding values.\n * \n * @return The [destination]
map.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeyAndValues\n */\npublic
inline fun <K, V, M : MutableMap<in K, MutableList<V>>> DoubleArray.groupByTo(destination: M, keySelector:
(Double) -> K, valueTransform: (Double) -> V): M {\n    for (element in this) {\n        val key =
keySelector(element)\n        val list = destination.getOrPut(key) { ArrayList<V>() }\n
list.add(valueTransform(element))\n    }\n    return destination\n}\n\n/**\n * Groups values returned by the
[valueTransform] function applied to each element of the original array\n * by the key returned by the given
[keySelector] function applied to the element\n * and puts to the [destination] map each group key associated with a
list of corresponding values.\n * \n * @return The [destination] map.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeyAndValues\n */\npublic inline fun <K, V, M :
MutableMap<in K, MutableList<V>>> BooleanArray.groupByTo(destination: M, keySelector: (Boolean) -> K,
valueTransform: (Boolean) -> V): M {\n    for (element in this) {\n        val key = keySelector(element)\n        val list
= destination.getOrPut(key) { ArrayList<V>() }\n        list.add(valueTransform(element))\n    }\n    return
destination\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the
original array\n * by the key returned by the given [keySelector] function applied to the element\n * and puts to the
[destination] map each group key associated with a list of corresponding values.\n * \n * @return The [destination]
map.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeyAndValues\n */\npublic
inline fun <K, V, M : MutableMap<in K, MutableList<V>>> CharArray.groupByTo(destination: M, keySelector:
(Char) -> K, valueTransform: (Char) -> V): M {\n    for (element in this) {\n        val key = keySelector(element)\n
        val list = destination.getOrPut(key) { ArrayList<V>() }\n        list.add(valueTransform(element))\n    }\n    return
destination\n}\n\n/**\n * Creates a [Grouping] source from an array to be used later with one of group-and-fold
operations\n * using the specified [keySelector] function to extract a key from each element.\n * \n * @sample
samples.collections.Grouping.groupingByEachCount\n */\n@SinceKotlin("1.1")\npublic inline fun <T, K>
Array<out T>.groupingBy(crossinline keySelector: (T) -> K): Grouping<T, K> {\n    return object : Grouping<T,
K> {\n        override fun sourceIterator(): Iterator<T> = this@groupingBy.iterator()\n        override fun
keyOf(element: T): K = keySelector(element)\n    }\n}\n\n/**\n * Returns a list containing the results of applying
the given [transform] function\n * to each element in the original array.\n * \n * @sample
samples.collections.Collections.Transformations.map\n */\npublic inline fun <T, R> Array<out T>.map(transform:
(T) -> R): List<R> {\n    return mapTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the
results of applying the given [transform] function\n * to each element in the original array.\n * \n * @sample
samples.collections.Collections.Transformations.map\n */\npublic inline fun <R> ByteArray.map(transform: (Byte)
-> R): List<R> {\n    return mapTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the
results of applying the given [transform] function\n * to each element in the original array.\n * \n * @sample
samples.collections.Collections.Transformations.map\n */\npublic inline fun <R> ShortArray.map(transform:
(Short) -> R): List<R> {\n    return mapTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing
the results of applying the given [transform] function\n * to each element in the original array.\n * \n * @sample
samples.collections.Collections.Transformations.map\n */\npublic inline fun <R> IntArray.map(transform: (Int) ->
R): List<R> {\n    return mapTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results
of applying the given [transform] function\n * to each element in the original array.\n * \n * @sample
samples.collections.Collections.Transformations.map\n */\npublic inline fun <R> LongArray.map(transform:
(Long) -> R): List<R> {\n    return mapTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing
the results of applying the given [transform] function\n * to each element in the original array.\n * \n * @sample
samples.collections.Collections.Transformations.map\n */\npublic inline fun <R> FloatArray.map(transform: (Float)
-> R): List<R> {\n    return mapTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the
results of applying the given [transform] function\n * to each element in the original array.\n * \n * @sample
samples.collections.Collections.Transformations.map\n */\npublic inline fun <R> DoubleArray.map(transform:

```

(Double) -> R): List<R> {\n return mapTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each element in the original array.\n * \n * @sample samples.collections.Collections.Transformations.map\n * \npublic inline fun <R> BooleanArray.map(transform: (Boolean) -> R): List<R> {\n return mapTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each element in the original array.\n * \n * \n * @sample samples.collections.Collections.Transformations.map\n * \npublic inline fun <R> CharArray.map(transform: (Char) -> R): List<R> {\n return mapTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each element and its index in the original array.\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n * \npublic inline fun <T, R> Array<out T>.mapIndexed(transform: (index: Int, T) -> R): List<R> {\n return mapIndexedTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each element and its index in the original array.\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n * \npublic inline fun <R> ShortArray.mapIndexed(transform: (index: Int, Short) -> R): List<R> {\n return mapIndexedTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each element and its index in the original array.\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n * \npublic inline fun <R> IntArray.mapIndexed(transform: (index: Int, Int) -> R): List<R> {\n return mapIndexedTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each element and its index in the original array.\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n * \npublic inline fun <R> LongArray.mapIndexed(transform: (index: Int, Long) -> R): List<R> {\n return mapIndexedTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each element and its index in the original array.\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n * \npublic inline fun <R> FloatArray.mapIndexed(transform: (index: Int, Float) -> R): List<R> {\n return mapIndexedTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each element and its index in the original array.\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n * \npublic inline fun <R> DoubleArray.mapIndexed(transform: (index: Int, Double) -> R): List<R> {\n return mapIndexedTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each element and its index in the original array.\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n * \npublic inline fun <R> BooleanArray.mapIndexed(transform: (index: Int, Boolean) -> R): List<R> {\n return mapIndexedTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each element and its index in the original array.\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n * \npublic inline fun <R> CharArray.mapIndexed(transform: (index: Int, Char) -> R): List<R> {\n return mapIndexedTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing only the non-null results of applying the given [transform] function\n * to each element and its index in the original array.\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n * \npublic inline fun <T, R : Any> Array<out

```

T>.mapIndexedNotNull(transform: (index: Int, T) -> R?): List<R> {\n  return
mapIndexedNotNullTo(ArrayList<R>(), transform)\n}\n\n/**\n * Applies the given [transform] function to each
element and its index in the original array\n * and appends only the non-null results to the given [destination].\n *
@param [transform] function that takes the index of an element and the element itself\n * and returns the result of
the transform applied to the element.\n */\npublic inline fun <T, R : Any, C : MutableCollection<in R>> Array<out
T>.mapIndexedNotNullTo(destination: C, transform: (index: Int, T) -> R?): C {\n  forEachIndexed { index,
element -> transform(index, element)?.let { destination.add(it) } }\n  return destination\n}\n\n/**\n * Applies the
given [transform] function to each element and its index in the original array\n * and appends the results to the given
[destination].\n * @param [transform] function that takes the index of an element and the element itself\n * and
returns the result of the transform applied to the element.\n */\npublic inline fun <T, R, C : MutableCollection<in
R>> Array<out T>.mapIndexedTo(destination: C, transform: (index: Int, T) -> R): C {\n  var index = 0\n  for
(item in this)\n    destination.add(transform(index++, item))\n  return destination\n}\n\n/**\n * Applies the given
[transform] function to each element and its index in the original array\n * and appends the results to the given
[destination].\n * @param [transform] function that takes the index of an element and the element itself\n * and
returns the result of the transform applied to the element.\n */\npublic inline fun <R, C : MutableCollection<in
R>> ByteArray.mapIndexedTo(destination: C, transform: (index: Int, Byte) -> R): C {\n  var index = 0\n  for (item in
this)\n    destination.add(transform(index++, item))\n  return destination\n}\n\n/**\n * Applies the given
[transform] function to each element and its index in the original array\n * and appends the results to the given
[destination].\n * @param [transform] function that takes the index of an element and the element itself\n * and
returns the result of the transform applied to the element.\n */\npublic inline fun <R, C : MutableCollection<in
R>> ShortArray.mapIndexedTo(destination: C, transform: (index: Int, Short) -> R): C {\n  var index = 0\n  for (item in
this)\n    destination.add(transform(index++, item))\n  return destination\n}\n\n/**\n * Applies the given
[transform] function to each element and its index in the original array\n * and appends the results to the given
[destination].\n * @param [transform] function that takes the index of an element and the element itself\n * and
returns the result of the transform applied to the element.\n */\npublic inline fun <R, C : MutableCollection<in
R>> IntArray.mapIndexedTo(destination: C, transform: (index: Int, Int) -> R): C {\n  var index = 0\n  for (item in
this)\n    destination.add(transform(index++, item))\n  return destination\n}\n\n/**\n * Applies the given
[transform] function to each element and its index in the original array\n * and appends the results to the given
[destination].\n * @param [transform] function that takes the index of an element and the element itself\n * and
returns the result of the transform applied to the element.\n */\npublic inline fun <R, C : MutableCollection<in
R>> LongArray.mapIndexedTo(destination: C, transform: (index: Int, Long) -> R): C {\n  var index = 0\n  for (item in
this)\n    destination.add(transform(index++, item))\n  return destination\n}\n\n/**\n * Applies the given
[transform] function to each element and its index in the original array\n * and appends the results to the given
[destination].\n * @param [transform] function that takes the index of an element and the element itself\n * and
returns the result of the transform applied to the element.\n */\npublic inline fun <R, C : MutableCollection<in
R>> FloatArray.mapIndexedTo(destination: C, transform: (index: Int, Float) -> R): C {\n  var index = 0\n  for (item in
this)\n    destination.add(transform(index++, item))\n  return destination\n}\n\n/**\n * Applies the given
[transform] function to each element and its index in the original array\n * and appends the results to the given
[destination].\n * @param [transform] function that takes the index of an element and the element itself\n * and
returns the result of the transform applied to the element.\n */\npublic inline fun <R, C : MutableCollection<in
R>> DoubleArray.mapIndexedTo(destination: C, transform: (index: Int, Double) -> R): C {\n  var index = 0\n  for
(item in this)\n    destination.add(transform(index++, item))\n  return destination\n}\n\n/**\n * Applies the given
[transform] function to each element and its index in the original array\n * and appends the results to the given
[destination].\n * @param [transform] function that takes the index of an element and the element itself\n * and
returns the result of the transform applied to the element.\n */\npublic inline fun <R, C : MutableCollection<in
R>> BooleanArray.mapIndexedTo(destination: C, transform: (index: Int, Boolean) -> R): C {\n  var index = 0\n  for
(item in this)\n    destination.add(transform(index++, item))\n  return destination\n}\n\n/**\n * Applies the given
[transform] function to each element and its index in the original array\n * and appends the results to the given

```

```

[destination].\n * @param [transform] function that takes the index of an element and the element itself\n * and
returns the result of the transform applied to the element.\n *\npublic inline fun <R, C : MutableCollection<in R>>
CharArray.mapIndexedTo(destination: C, transform: (index: Int, Char) -> R): C {\n    var index = 0\n    for (item in
this)\n        destination.add(transform(index++, item))\n    return destination\n}\n\n/**\n * Returns a list containing
only the non-null results of applying the given [transform] function\n * to each element in the original array.\n * \n *
@sample samples.collections.Collections.Transformations.mapNotNull\n *\npublic inline fun <T, R : Any>
Array<out T>.mapNotNull(transform: (T) -> R?): List<R> {\n    return mapNotNullTo(ArrayList<R>(),
transform)\n}\n\n/**\n * Applies the given [transform] function to each element in the original array\n * and
appends only the non-null results to the given [destination].\n *\npublic inline fun <T, R : Any, C :
MutableCollection<in R>> Array<out T>.mapNotNullTo(destination: C, transform: (T) -> R?): C {\n    forEach {
element -> transform(element)?.let { destination.add(it) } }\n    return destination\n}\n\n/**\n * Applies the given
[transform] function to each element of the original array\n * and appends the results to the given [destination].\n
*\npublic inline fun <T, R, C : MutableCollection<in R>> Array<out T>.mapTo(destination: C, transform: (T) ->
R): C {\n    for (item in this)\n        destination.add(transform(item))\n    return destination\n}\n\n/**\n * Applies the
given [transform] function to each element of the original array\n * and appends the results to the given
[destination].\n *\npublic inline fun <R, C : MutableCollection<in R>> ByteArray.mapTo(destination: C,
transform: (Byte) -> R): C {\n    for (item in this)\n        destination.add(transform(item))\n    return
destination\n}\n\n/**\n * Applies the given [transform] function to each element of the original array\n * and
appends the results to the given [destination].\n *\npublic inline fun <R, C : MutableCollection<in R>>
ShortArray.mapTo(destination: C, transform: (Short) -> R): C {\n    for (item in this)\n
destination.add(transform(item))\n    return destination\n}\n\n/**\n * Applies the given [transform] function to each
element of the original array\n * and appends the results to the given [destination].\n *\npublic inline fun <R, C :
MutableCollection<in R>> IntArray.mapTo(destination: C, transform: (Int) -> R): C {\n    for (item in this)\n
destination.add(transform(item))\n    return destination\n}\n\n/**\n * Applies the given [transform] function to each
element of the original array\n * and appends the results to the given [destination].\n *\npublic inline fun <R, C :
MutableCollection<in R>> LongArray.mapTo(destination: C, transform: (Long) -> R): C {\n    for (item in this)\n
destination.add(transform(item))\n    return destination\n}\n\n/**\n * Applies the given [transform] function to
each element of the original array\n * and appends the results to the given [destination].\n *\npublic inline fun <R,
C : MutableCollection<in R>> FloatArray.mapTo(destination: C, transform: (Float) -> R): C {\n    for (item in
this)\n        destination.add(transform(item))\n    return destination\n}\n\n/**\n * Applies the given [transform]
function to each element of the original array\n * and appends the results to the given [destination].\n *\npublic
inline fun <R, C : MutableCollection<in R>> DoubleArray.mapTo(destination: C, transform: (Double) -> R): C {\n
    for (item in this)\n        destination.add(transform(item))\n    return destination\n}\n\n/**\n * Applies the given
[transform] function to each element of the original array\n * and appends the results to the given [destination].\n
*\npublic inline fun <R, C : MutableCollection<in R>> BooleanArray.mapTo(destination: C, transform: (Boolean)
-> R): C {\n    for (item in this)\n        destination.add(transform(item))\n    return destination\n}\n\n/**\n * Applies
the given [transform] function to each element of the original array\n * and appends the results to the given
[destination].\n *\npublic inline fun <R, C : MutableCollection<in R>> CharArray.mapTo(destination: C,
transform: (Char) -> R): C {\n    for (item in this)\n        destination.add(transform(item))\n    return
destination\n}\n\n/**\n * Returns a lazy [Iterable] that wraps each element of the original array\n * into an
[IndexValue] containing the index of that element and the element itself.\n *\npublic fun <T> Array<out
T>.withIndex(): Iterable<IndexedValue<T>> {\n    return IndexingIterable { iterator() }\n}\n\n/**\n * Returns a
lazy [Iterable] that wraps each element of the original array\n * into an [IndexedValue] containing the index of that
element and the element itself.\n *\npublic fun ByteArray.withIndex(): Iterable<IndexedValue<Byte>> {\n    return
IndexingIterable { iterator() }\n}\n\n/**\n * Returns a lazy [Iterable] that wraps each element of the original array\n
* into an [IndexedValue] containing the index of that element and the element itself.\n *\npublic fun
ShortArray.withIndex(): Iterable<IndexedValue<Short>> {\n    return IndexingIterable { iterator() }\n}\n\n/**\n *
Returns a lazy [Iterable] that wraps each element of the original array\n * into an [IndexedValue] containing the

```

index of that element and the element itself.

```

public fun IntArray.withIndex(): Iterable<IndexedValue<Int>>
{
    return IndexingIterable { iterator() }
}

```

* Returns a lazy [Iterable] that wraps each element of the original array into an [IndexedValue] containing the index of that element and the element itself.

```

public fun LongArray.withIndex(): Iterable<IndexedValue<Long>>
{
    return IndexingIterable { iterator() }
}

```

* Returns a lazy [Iterable] that wraps each element of the original array into an [IndexedValue] containing the index of that element and the element itself.

```

public fun FloatArray.withIndex():
Iterable<IndexedValue<Float>>
{
    return IndexingIterable { iterator() }
}

```

* Returns a lazy [Iterable] that wraps each element of the original array into an [IndexedValue] containing the index of that element and the element itself.

```

public fun DoubleArray.withIndex(): Iterable<IndexedValue<Double>>
{
    return
IndexingIterable { iterator() }
}

```

* Returns a lazy [Iterable] that wraps each element of the original array into an [IndexedValue] containing the index of that element and the element itself.

```

public fun BooleanArray.withIndex(): Iterable<IndexedValue<Boolean>>
{
    return IndexingIterable { iterator()
}
}

```

* Returns a lazy [Iterable] that wraps each element of the original array into an [IndexedValue] containing the index of that element and the element itself.

```

public fun CharArray.withIndex():
Iterable<IndexedValue<Char>>
{
    return IndexingIterable { iterator() }
}

```

* Returns a list containing only distinct elements from the given array.

* Among equal elements of the given array, only the first one will be present in the resulting list.

* The elements in the resulting list are in the same order as they were in the source array.

```

@sample samples.collections.Collections.Transformations.distinctAndDistinctBy
<T> Array<out T>.distinct(): List<T>
{
    return this.toMutableSet().toList()
}

```

* Returns a list containing only distinct elements from the given array.

* The elements in the resulting list are in the same order as they were in the source array.

```

@sample
samples.collections.Collections.Transformations.distinctAndDistinctBy
public fun ByteArray.distinct():
List<Byte>
{
    return this.toMutableSet().toList()
}

```

* Returns a list containing only distinct elements from the given array.

* The elements in the resulting list are in the same order as they were in the source array.

```

@sample samples.collections.Collections.Transformations.distinctAndDistinctBy
public fun ShortArray.distinct(): List<Short>
{
    return this.toMutableSet().toList()
}

```

* Returns a list containing only distinct elements from the given array.

* The elements in the resulting list are in the same order as they were in the source array.

```

@sample
samples.collections.Collections.Transformations.distinctAndDistinctBy
public fun IntArray.distinct():
List<Int>
{
    return this.toMutableSet().toList()
}

```

* Returns a list containing only distinct elements from the given array.

* The elements in the resulting list are in the same order as they were in the source array.

```

@sample samples.collections.Collections.Transformations.distinctAndDistinctBy
public fun LongArray.distinct(): List<Long>
{
    return this.toMutableSet().toList()
}

```

* Returns a list containing only distinct elements from the given array.

* The elements in the resulting list are in the same order as they were in the source array.

```

@sample
samples.collections.Collections.Transformations.distinctAndDistinctBy
public fun FloatArray.distinct():
List<Float>
{
    return this.toMutableSet().toList()
}

```

* Returns a list containing only distinct elements from the given array.

* The elements in the resulting list are in the same order as they were in the source array.

```

@sample samples.collections.Collections.Transformations.distinctAndDistinctBy
public fun DoubleArray.distinct(): List<Double>
{
    return this.toMutableSet().toList()
}

```

* Returns a list containing only distinct elements from the given array.

* The elements in the resulting list are in the same order as they were in the source array.

```

@sample
samples.collections.Collections.Transformations.distinctAndDistinctBy
public fun BooleanArray.distinct():
List<Boolean>
{
    return this.toMutableSet().toList()
}

```

* Returns a list containing only distinct elements from the given array.

* The elements in the resulting list are in the same order as they were in the source array.

```

@sample samples.collections.Collections.Transformations.distinctAndDistinctBy
public fun CharArray.distinct(): List<Char>
{
    return this.toMutableSet().toList()
}

```

* Returns a list containing only elements from the given array having distinct keys returned by the given [selector] function.

Among elements of the given array with equal keys, only the first one will be present in the resulting list. The elements in the resulting list are in the same order as they were in the source array.

```

@sample
samples.collections.Collections.Transformations.distinctAndDistinctBy
public inline fun <T, K> Array<out T>.distinctBy(selector: (T) -> K): List<T> {
    val set = HashSet<K>()
    val list = ArrayList<T>()
    for (e in this) {
        val key = selector(e)
        if (set.add(key)) list.add(e)
    }
    return list
}

```

Returns a list containing only elements from the given array having distinct keys returned by the given [selector] function. The elements in the resulting list are in the same order as they were in the source array.

```

@sample
samples.collections.Collections.Transformations.distinctAndDistinctBy
public inline fun <K> ByteArray.distinctBy(selector: (Byte) -> K): List<Byte> {
    val set = HashSet<K>()
    val list = ArrayList<Byte>()
    for (e in this) {
        val key = selector(e)
        if (set.add(key)) list.add(e)
    }
    return list
}

```

Returns a list containing only elements from the given array having distinct keys returned by the given [selector] function. The elements in the resulting list are in the same order as they were in the source array.

```

@sample
samples.collections.Collections.Transformations.distinctAndDistinctBy
public inline fun <K> ShortArray.distinctBy(selector: (Short) -> K): List<Short> {
    val set = HashSet<K>()
    val list = ArrayList<Short>()
    for (e in this) {
        val key = selector(e)
        if (set.add(key)) list.add(e)
    }
    return list
}

```

Returns a list containing only elements from the given array having distinct keys returned by the given [selector] function. The elements in the resulting list are in the same order as they were in the source array.

```

@sample
samples.collections.Collections.Transformations.distinctAndDistinctBy
public inline fun <K> IntArray.distinctBy(selector: (Int) -> K): List<Int> {
    val set = HashSet<K>()
    val list = ArrayList<Int>()
    for (e in this) {
        val key = selector(e)
        if (set.add(key)) list.add(e)
    }
    return list
}

```

Returns a list containing only elements from the given array having distinct keys returned by the given [selector] function. The elements in the resulting list are in the same order as they were in the source array.

```

@sample
samples.collections.Collections.Transformations.distinctAndDistinctBy
public inline fun <K> LongArray.distinctBy(selector: (Long) -> K): List<Long> {
    val set = HashSet<K>()
    val list = ArrayList<Long>()
    for (e in this) {
        val key = selector(e)
        if (set.add(key)) list.add(e)
    }
    return list
}

```

Returns a list containing only elements from the given array having distinct keys returned by the given [selector] function. The elements in the resulting list are in the same order as they were in the source array.

```

@sample
samples.collections.Collections.Transformations.distinctAndDistinctBy
public inline fun <K> FloatArray.distinctBy(selector: (Float) -> K): List<Float> {
    val set = HashSet<K>()
    val list = ArrayList<Float>()
    for (e in this) {
        val key = selector(e)
        if (set.add(key)) list.add(e)
    }
    return list
}

```

Returns a list containing only elements from the given array having distinct keys returned by the given [selector] function. The elements in the resulting list are in the same order as they were in the source array.

```

@sample
samples.collections.Collections.Transformations.distinctAndDistinctBy
public inline fun <K> DoubleArray.distinctBy(selector: (Double) -> K): List<Double> {
    val set = HashSet<K>()
    val list = ArrayList<Double>()
    for (e in this) {
        val key = selector(e)
        if (set.add(key)) list.add(e)
    }
    return list
}

```

Returns a list containing only elements from the given array having distinct keys returned by the given [selector] function. The elements in the resulting list are in the same order as they were in the source array.

```

@sample
samples.collections.Collections.Transformations.distinctAndDistinctBy
public inline fun <K> BooleanArray.distinctBy(selector: (Boolean) -> K): List<Boolean> {
    val set = HashSet<K>()
    val list = ArrayList<Boolean>()
    for (e in this) {
        val key = selector(e)
        if (set.add(key)) list.add(e)
    }
    return list
}

```

Returns a list containing only elements from the given array having distinct keys returned by the given [selector] function. The elements in the resulting list are in the same order as they were in the source array.

```

samples.collections.Collections.Transformations.distinctAndDistinctBy\n *^\npublic inline fun <K>
CharArray.distinctBy(selector: (Char) -> K): List<Char> {\n    val set = HashSet<K>()\n    val list =
ArrayList<Char>()\n    for (e in this) {\n        val key = selector(e)\n        if (set.add(key))\n            list.add(e)\n    }\n    return list\n}\n\n/**\n * Returns a set containing all elements that are contained by both this array and the specified
collection.\n * \n * The returned set preserves the element iteration order of the original array.\n * \n * To get a set
containing all elements that are contained at least in one of these collections use [union].\n *^\npublic infix fun <T>
Array<out T>.intersect(other: Iterable<T>): Set<T> {\n    val set = this.toMutableSet()\n    set.retainAll(other)\n
return set\n}\n\n/**\n * Returns a set containing all elements that are contained by both this array and the specified
collection.\n * \n * The returned set preserves the element iteration order of the original array.\n * \n * To get a set
containing all elements that are contained at least in one of these collections use [union].\n *^\npublic infix fun
ByteArray.intersect(other: Iterable<Byte>): Set<Byte> {\n    val set = this.toMutableSet()\n    set.retainAll(other)\n
return set\n}\n\n/**\n * Returns a set containing all elements that are contained by both this array and the specified
collection.\n * \n * The returned set preserves the element iteration order of the original array.\n * \n * To get a set
containing all elements that are contained at least in one of these collections use [union].\n *^\npublic infix fun
ShortArray.intersect(other: Iterable<Short>): Set<Short> {\n    val set = this.toMutableSet()\n
set.retainAll(other)\n    return set\n}\n\n/**\n * Returns a set containing all elements that are contained by both this
array and the specified collection.\n * \n * The returned set preserves the element iteration order of the original
array.\n * \n * To get a set containing all elements that are contained at least in one of these collections use
[union].\n *^\npublic infix fun IntArray.intersect(other: Iterable<Int>): Set<Int> {\n    val set = this.toMutableSet()\n
set.retainAll(other)\n    return set\n}\n\n/**\n * Returns a set containing all elements that are contained by both this
array and the specified collection.\n * \n * The returned set preserves the element iteration order of the original
array.\n * \n * To get a set containing all elements that are contained at least in one of these collections use
[union].\n *^\npublic infix fun LongArray.intersect(other: Iterable<Long>): Set<Long> {\n    val set =
this.toMutableSet()\n    set.retainAll(other)\n    return set\n}\n\n/**\n * Returns a set containing all elements that are
contained by both this array and the specified collection.\n * \n * The returned set preserves the element iteration
order of the original array.\n * \n * To get a set containing all elements that are contained at least in one of these
collections use [union].\n *^\npublic infix fun FloatArray.intersect(other: Iterable<Float>): Set<Float> {\n    val set
= this.toMutableSet()\n    set.retainAll(other)\n    return set\n}\n\n/**\n * Returns a set containing all elements that
are contained by both this array and the specified collection.\n * \n * The returned set preserves the element iteration
order of the original array.\n * \n * To get a set containing all elements that are contained at least in one of these
collections use [union].\n *^\npublic infix fun DoubleArray.intersect(other: Iterable<Double>): Set<Double> {\n
val set = this.toMutableSet()\n    set.retainAll(other)\n    return set\n}\n\n/**\n * Returns a set containing all
elements that are contained by both this array and the specified collection.\n * \n * The returned set preserves the
element iteration order of the original array.\n * \n * To get a set containing all elements that are contained at least
in one of these collections use [union].\n *^\npublic infix fun BooleanArray.intersect(other: Iterable<Boolean>):
Set<Boolean> {\n    val set = this.toMutableSet()\n    set.retainAll(other)\n    return set\n}\n\n/**\n * Returns a set
containing all elements that are contained by both this array and the specified collection.\n * \n * The returned set
preserves the element iteration order of the original array.\n * \n * To get a set containing all elements that are
contained at least in one of these collections use [union].\n *^\npublic infix fun CharArray.intersect(other:
Iterable<Char>): Set<Char> {\n    val set = this.toMutableSet()\n    set.retainAll(other)\n    return set\n}\n\n/**\n
* Returns a set containing all elements that are contained by this array and not contained by the specified collection.\n
* \n * The returned set preserves the element iteration order of the original array.\n *^\npublic infix fun <T>
Array<out T>.subtract(other: Iterable<T>): Set<T> {\n    val set = this.toMutableSet()\n    set.removeAll(other)\n
return set\n}\n\n/**\n * Returns a set containing all elements that are contained by this array and not contained by
the specified collection.\n * \n * The returned set preserves the element iteration order of the original array.\n
*^\npublic infix fun ByteArray.subtract(other: Iterable<Byte>): Set<Byte> {\n    val set = this.toMutableSet()\n
set.removeAll(other)\n    return set\n}\n\n/**\n * Returns a set containing all elements that are contained by this
array and not contained by the specified collection.\n * \n * The returned set preserves the element iteration order of

```

the original array.

```

public infix fun ShortArray.subtract(other: Iterable<Short>): Set<Short> {
    val set = this.toMutableSet()
    set.removeAll(other)
    return set
}

```

* Returns a set containing all elements that are contained by this array and not contained by the specified collection.

* The returned set preserves the element iteration order of the original array.

```

public infix fun IntArray.subtract(other: Iterable<Int>): Set<Int> {
    val set = this.toMutableSet()
    set.removeAll(other)
    return set
}

```

* Returns a set containing all elements that are contained by this array and not contained by the specified collection.

* The returned set preserves the element iteration order of the original array.

```

public infix fun LongArray.subtract(other: Iterable<Long>): Set<Long> {
    val set = this.toMutableSet()
    set.removeAll(other)
    return set
}

```

* Returns a set containing all elements that are contained by this array and not contained by the specified collection.

* The returned set preserves the element iteration order of the original array.

```

public infix fun FloatArray.subtract(other: Iterable<Float>): Set<Float> {
    val set = this.toMutableSet()
    set.removeAll(other)
    return set
}

```

* Returns a set containing all elements that are contained by this array and not contained by the specified collection.

* The returned set preserves the element iteration order of the original array.

```

public infix fun DoubleArray.subtract(other: Iterable<Double>): Set<Double> {
    val set = this.toMutableSet()
    set.removeAll(other)
    return set
}

```

* Returns a set containing all elements that are contained by this array and not contained by the specified collection.

* The returned set preserves the element iteration order of the original array.

```

public infix fun BooleanArray.subtract(other: Iterable<Boolean>): Set<Boolean> {
    val set = this.toMutableSet()
    set.removeAll(other)
    return set
}

```

* Returns a set containing all elements that are contained by this array and not contained by the specified collection.

* The returned set preserves the element iteration order of the original array.

```

public infix fun CharArray.subtract(other: Iterable<Char>): Set<Char> {
    val set = this.toMutableSet()
    set.removeAll(other)
    return set
}

```

* Returns a new [MutableSet] containing all distinct elements from the given array.

* The returned set preserves the element iteration order of the original array.

```

public fun <T> Array<out T>.toMutableSet(): MutableSet<T> {
    return toCollection(LinkedHashSet<T>(mapCapacity(size)))
}

```

* Returns a new [MutableSet] containing all distinct elements from the given array.

* The returned set preserves the element iteration order of the original array.

```

public fun ByteArray.toMutableSet(): MutableSet<Byte> {
    return toCollection(LinkedHashSet<Byte>(mapCapacity(size)))
}

```

* Returns a new [MutableSet] containing all distinct elements from the given array.

* The returned set preserves the element iteration order of the original array.

```

public fun ShortArray.toMutableSet(): MutableSet<Short> {
    return toCollection(LinkedHashSet<Short>(mapCapacity(size)))
}

```

* Returns a new [MutableSet] containing all distinct elements from the given array.

* The returned set preserves the element iteration order of the original array.

```

public fun IntArray.toMutableSet(): MutableSet<Int> {
    return toCollection(LinkedHashSet<Int>(mapCapacity(size)))
}

```

* Returns a new [MutableSet] containing all distinct elements from the given array.

* The returned set preserves the element iteration order of the original array.

```

public fun LongArray.toMutableSet(): MutableSet<Long> {
    return toCollection(LinkedHashSet<Long>(mapCapacity(size)))
}

```

* Returns a new [MutableSet] containing all distinct elements from the given array.

* The returned set preserves the element iteration order of the original array.

```

public fun FloatArray.toMutableSet(): MutableSet<Float> {
    return toCollection(LinkedHashSet<Float>(mapCapacity(size)))
}

```

* Returns a new [MutableSet] containing all distinct elements from the given array.

* The returned set preserves the element iteration order of the original array.

```

public fun DoubleArray.toMutableSet(): MutableSet<Double> {
    return toCollection(LinkedHashSet<Double>(mapCapacity(size)))
}

```

* Returns a new [MutableSet] containing all distinct elements from the given array.

* The returned set preserves the element iteration order of the original array.

```

public fun BooleanArray.toMutableSet(): MutableSet<Boolean> {
    return toCollection(LinkedHashSet<Boolean>(mapCapacity(size)))
}

```

* Returns a new [MutableSet] containing all distinct elements from the given array.

* The returned set preserves the element iteration order of the original array.

```

public fun CharArray.toMutableSet(): MutableSet<Char> {
    return toCollection(LinkedHashSet<Char>(mapCapacity(size.coerceAtMost(128))))
}

```

* Returns a set containing

all distinct elements from both collections.

`\n * \n * The returned set preserves the element iteration order of the original array.\n * Those elements of the [other] collection that are unique are iterated in the end\n * in the order of the [other] collection.\n * \n * To get a set containing all elements that are contained in both collections use [intersect].\n */\npublic infix fun <T> Array<out T>.union(other: Iterable<T>): Set<T> {\n val set = this.toMutableSet()\n set.addAll(other)\n return set\n}\n\n/**\n * Returns a set containing all distinct elements from both collections.\n * \n * The returned set preserves the element iteration order of the original array.\n * Those elements of the [other] collection that are unique are iterated in the end\n * in the order of the [other] collection.\n * \n * To get a set containing all elements that are contained in both collections use [intersect].\n */\npublic infix fun ByteArray.union(other: Iterable<Byte>): Set<Byte> {\n val set = this.toMutableSet()\n set.addAll(other)\n return set\n}\n\n/**\n * Returns a set containing all distinct elements from both collections.\n * \n * The returned set preserves the element iteration order of the original array.\n * Those elements of the [other] collection that are unique are iterated in the end\n * in the order of the [other] collection.\n * \n * To get a set containing all elements that are contained in both collections use [intersect].\n */\npublic infix fun ShortArray.union(other: Iterable<Short>): Set<Short> {\n val set = this.toMutableSet()\n set.addAll(other)\n return set\n}\n\n/**\n * Returns a set containing all distinct elements from both collections.\n * \n * The returned set preserves the element iteration order of the original array.\n * Those elements of the [other] collection that are unique are iterated in the end\n * in the order of the [other] collection.\n * \n * To get a set containing all elements that are contained in both collections use [intersect].\n */\npublic infix fun IntArray.union(other: Iterable<Int>): Set<Int> {\n val set = this.toMutableSet()\n set.addAll(other)\n return set\n}\n\n/**\n * Returns a set containing all distinct elements from both collections.\n * \n * The returned set preserves the element iteration order of the original array.\n * Those elements of the [other] collection that are unique are iterated in the end\n * in the order of the [other] collection.\n * \n * To get a set containing all elements that are contained in both collections use [intersect].\n */\npublic infix fun LongArray.union(other: Iterable<Long>): Set<Long> {\n val set = this.toMutableSet()\n set.addAll(other)\n return set\n}\n\n/**\n * Returns a set containing all distinct elements from both collections.\n * \n * The returned set preserves the element iteration order of the original array.\n * Those elements of the [other] collection that are unique are iterated in the end\n * in the order of the [other] collection.\n * \n * To get a set containing all elements that are contained in both collections use [intersect].\n */\npublic infix fun FloatArray.union(other: Iterable<Float>): Set<Float> {\n val set = this.toMutableSet()\n set.addAll(other)\n return set\n}\n\n/**\n * Returns a set containing all distinct elements from both collections.\n * \n * The returned set preserves the element iteration order of the original array.\n * Those elements of the [other] collection that are unique are iterated in the end\n * in the order of the [other] collection.\n * \n * To get a set containing all elements that are contained in both collections use [intersect].\n */\npublic infix fun DoubleArray.union(other: Iterable<Double>): Set<Double> {\n val set = this.toMutableSet()\n set.addAll(other)\n return set\n}\n\n/**\n * Returns a set containing all distinct elements from both collections.\n * \n * The returned set preserves the element iteration order of the original array.\n * Those elements of the [other] collection that are unique are iterated in the end\n * in the order of the [other] collection.\n * \n * To get a set containing all elements that are contained in both collections use [intersect].\n */\npublic infix fun BooleanArray.union(other: Iterable<Boolean>): Set<Boolean> {\n val set = this.toMutableSet()\n set.addAll(other)\n return set\n}\n\n/**\n * Returns a set containing all distinct elements from both collections.\n * \n * The returned set preserves the element iteration order of the original array.\n * Those elements of the [other] collection that are unique are iterated in the end\n * in the order of the [other] collection.\n * \n * To get a set containing all elements that are contained in both collections use [intersect].\n */\npublic infix fun CharArray.union(other: Iterable<Char>): Set<Char> {\n val set = this.toMutableSet()\n set.addAll(other)\n return set\n}\n\n/**\n * Returns `true` if all elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.all\n */\npublic inline fun <T> Array<out T>.all(predicate: (T) -> Boolean): Boolean {\n for (element in this) if (!predicate(element)) return false\n return true\n}\n\n/**\n * Returns `true` if all elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.all\n */\npublic inline fun ByteArray.all(predicate: (Byte) -> Boolean): Boolean {\n for (element in this) if (!predicate(element)) return false\n return true\n}\n\n/**\n * Returns `true` if`

```

all elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.all\n
*\npublic inline fun ShortArray.all(predicate: (Short) -> Boolean): Boolean {\n  for (element in this) if
(!predicate(element)) return false\n  return true\n}\n\n**\n * Returns `true` if all elements match the given
[predicate].\n * \n * @sample samples.collections.Collections.Aggregates.all\n
*\npublic inline fun
IntArray.all(predicate: (Int) -> Boolean): Boolean {\n  for (element in this) if (!predicate(element)) return false\n
return true\n}\n\n**\n * Returns `true` if all elements match the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.all\n
*\npublic inline fun LongArray.all(predicate: (Long) -> Boolean):
Boolean {\n  for (element in this) if (!predicate(element)) return false\n  return true\n}\n\n**\n * Returns `true` if
all elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.all\n
*\npublic inline fun FloatArray.all(predicate: (Float) -> Boolean): Boolean {\n  for (element in this) if
(!predicate(element)) return false\n  return true\n}\n\n**\n * Returns `true` if all elements match the given
[predicate].\n * \n * @sample samples.collections.Collections.Aggregates.all\n
*\npublic inline fun
DoubleArray.all(predicate: (Double) -> Boolean): Boolean {\n  for (element in this) if (!predicate(element)) return
false\n  return true\n}\n\n**\n * Returns `true` if all elements match the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.all\n
*\npublic inline fun BooleanArray.all(predicate: (Boolean) ->
Boolean): Boolean {\n  for (element in this) if (!predicate(element)) return false\n  return true\n}\n\n**\n *
Returns `true` if all elements match the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.all\n
*\npublic inline fun CharArray.all(predicate: (Char) -> Boolean):
Boolean {\n  for (element in this) if (!predicate(element)) return false\n  return true\n}\n\n**\n * Returns `true` if
array has at least one element.\n * \n * @sample samples.collections.Collections.Aggregates.any\n
*\npublic fun
<T> Array<out T>.any(): Boolean {\n  return !isEmpty()\n}\n\n**\n * Returns `true` if array has at least one
element.\n * \n * @sample samples.collections.Collections.Aggregates.any\n
*\npublic fun ByteArray.any():
Boolean {\n  return !isEmpty()\n}\n\n**\n * Returns `true` if array has at least one element.\n * \n * @sample
samples.collections.Collections.Aggregates.any\n
*\npublic fun ShortArray.any(): Boolean {\n  return
!isEmpty()\n}\n\n**\n * Returns `true` if array has at least one element.\n * \n * @sample
samples.collections.Collections.Aggregates.any\n
*\npublic fun IntArray.any(): Boolean {\n  return
!isEmpty()\n}\n\n**\n * Returns `true` if array has at least one element.\n * \n * @sample
samples.collections.Collections.Aggregates.any\n
*\npublic fun LongArray.any(): Boolean {\n  return
!isEmpty()\n}\n\n**\n * Returns `true` if array has at least one element.\n * \n * @sample
samples.collections.Collections.Aggregates.any\n
*\npublic fun FloatArray.any(): Boolean {\n  return
!isEmpty()\n}\n\n**\n * Returns `true` if array has at least one element.\n * \n * @sample
samples.collections.Collections.Aggregates.any\n
*\npublic fun DoubleArray.any(): Boolean {\n  return
!isEmpty()\n}\n\n**\n * Returns `true` if array has at least one element.\n * \n * @sample
samples.collections.Collections.Aggregates.any\n
*\npublic fun BooleanArray.any(): Boolean {\n  return
!isEmpty()\n}\n\n**\n * Returns `true` if array has at least one element.\n * \n * @sample
samples.collections.Collections.Aggregates.any\n
*\npublic fun CharArray.any(): Boolean {\n  return
!isEmpty()\n}\n\n**\n * Returns `true` if at least one element matches the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.anyWithPredicate\n
*\npublic inline fun <T> Array<out
T>.any(predicate: (T) -> Boolean): Boolean {\n  for (element in this) if (predicate(element)) return true\n  return
false\n}\n\n**\n * Returns `true` if at least one element matches the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.anyWithPredicate\n
*\npublic inline fun ByteArray.any(predicate:
(Byte) -> Boolean): Boolean {\n  for (element in this) if (predicate(element)) return true\n  return
false\n}\n\n**\n * Returns `true` if at least one element matches the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.anyWithPredicate\n
*\npublic inline fun ShortArray.any(predicate:
(Short) -> Boolean): Boolean {\n  for (element in this) if (predicate(element)) return true\n  return
false\n}\n\n**\n * Returns `true` if at least one element matches the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.anyWithPredicate\n
*\npublic inline fun IntArray.any(predicate: (Int) -
> Boolean): Boolean {\n  for (element in this) if (predicate(element)) return true\n  return false\n}\n\n**\n *

```

Returns `true` if at least one element matches the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.anyWithPredicate\n * \n public inline fun LongArray.any(predicate:
(Long) -> Boolean): Boolean {\n for (element in this) if (predicate(element)) return true\n return
false\n}\n\n/**\n * Returns `true` if at least one element matches the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.anyWithPredicate\n * \n public inline fun FloatArray.any(predicate:
(Float) -> Boolean): Boolean {\n for (element in this) if (predicate(element)) return true\n return
false\n}\n\n/**\n * Returns `true` if at least one element matches the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.anyWithPredicate\n * \n public inline fun DoubleArray.any(predicate:
(Double) -> Boolean): Boolean {\n for (element in this) if (predicate(element)) return true\n return
false\n}\n\n/**\n * Returns `true` if at least one element matches the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.anyWithPredicate\n * \n public inline fun BooleanArray.any(predicate:
(Boolean) -> Boolean): Boolean {\n for (element in this) if (predicate(element)) return true\n return
false\n}\n\n/**\n * Returns `true` if at least one element matches the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.anyWithPredicate\n * \n public inline fun CharArray.any(predicate:
(Char) -> Boolean): Boolean {\n for (element in this) if (predicate(element)) return true\n return
false\n}\n\n/**\n * Returns the number of elements in this array.\n * \n @kotlin.internal.InlineOnly\n public inline
fun <T> Array<out T>.count(): Int {\n return size\n}\n\n/**\n * Returns the number of elements in this array.\n * \n @kotlin.internal.InlineOnly\n public inline fun ByteArray.count(): Int {\n return size\n}\n\n/**\n * Returns the
number of elements in this array.\n * \n @kotlin.internal.InlineOnly\n public inline fun ShortArray.count(): Int {\n
return size\n}\n\n/**\n * Returns the number of elements in this array.\n * \n @kotlin.internal.InlineOnly\n public
inline fun IntArray.count(): Int {\n return size\n}\n\n/**\n * Returns the number of elements in this array.\n * \n @kotlin.internal.InlineOnly\n public inline fun LongArray.count(): Int {\n return size\n}\n\n/**\n * Returns the
number of elements in this array.\n * \n @kotlin.internal.InlineOnly\n public inline fun FloatArray.count(): Int {\n
return size\n}\n\n/**\n * Returns the number of elements in this array.\n * \n @kotlin.internal.InlineOnly\n public
inline fun DoubleArray.count(): Int {\n return size\n}\n\n/**\n * Returns the number of elements in this array.\n * \n @kotlin.internal.InlineOnly\n public inline fun BooleanArray.count(): Int {\n return size\n}\n\n/**\n * Returns
the number of elements in this array.\n * \n @kotlin.internal.InlineOnly\n public inline fun CharArray.count(): Int {\n
return size\n}\n\n/**\n * Returns the number of elements matching the given [predicate].\n * \n public inline fun
<T> Array<out T>.count(predicate: (T) -> Boolean): Int {\n var count = 0\n for (element in this) if
(predicate(element)) ++count\n return count\n}\n\n/**\n * Returns the number of elements matching the given
[predicate].\n * \n public inline fun ByteArray.count(predicate: (Byte) -> Boolean): Int {\n var count = 0\n for
(element in this) if (predicate(element)) ++count\n return count\n}\n\n/**\n * Returns the number of elements
matching the given [predicate].\n * \n public inline fun ShortArray.count(predicate: (Short) -> Boolean): Int {\n var
count = 0\n for (element in this) if (predicate(element)) ++count\n return count\n}\n\n/**\n * Returns the
number of elements matching the given [predicate].\n * \n public inline fun IntArray.count(predicate: (Int) ->
Boolean): Int {\n var count = 0\n for (element in this) if (predicate(element)) ++count\n return
count\n}\n\n/**\n * Returns the number of elements matching the given [predicate].\n * \n public inline fun
LongArray.count(predicate: (Long) -> Boolean): Int {\n var count = 0\n for (element in this) if
(predicate(element)) ++count\n return count\n}\n\n/**\n * Returns the number of elements matching the given
[predicate].\n * \n public inline fun FloatArray.count(predicate: (Float) -> Boolean): Int {\n var count = 0\n for
(element in this) if (predicate(element)) ++count\n return count\n}\n\n/**\n * Returns the number of elements
matching the given [predicate].\n * \n public inline fun DoubleArray.count(predicate: (Double) -> Boolean): Int {\n
var count = 0\n for (element in this) if (predicate(element)) ++count\n return count\n}\n\n/**\n * Returns the
number of elements matching the given [predicate].\n * \n public inline fun BooleanArray.count(predicate: (Boolean)
-> Boolean): Int {\n var count = 0\n for (element in this) if (predicate(element)) ++count\n return
count\n}\n\n/**\n * Returns the number of elements matching the given [predicate].\n * \n public inline fun
CharArray.count(predicate: (Char) -> Boolean): Int {\n var count = 0\n for (element in this) if
(predicate(element)) ++count\n return count\n}\n\n/**\n * Accumulates value starting with [initial] value and

applying [operation] from left to right
 * to current accumulator value and each element.
 * Returns the specified [initial] value if the array is empty.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 */
 public inline fun <T, R> Array<out T>.fold(initial: R, operation: (acc: R, T) -> R): R {
 \n var accumulator = initial
 for (element in this)
 accumulator = operation(accumulator, element)
 return accumulator
 }
 */
 * Accumulates value starting with [initial] value and applying [operation] from left to right
 * to current accumulator value and each element.
 * Returns the specified [initial] value if the array is empty.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 */
 public inline fun <R> ByteArray.fold(initial: R, operation: (acc: R, Byte) -> R): R {
 \n var accumulator = initial
 for (element in this)
 accumulator = operation(accumulator, element)
 return accumulator
 }
 */
 * Accumulates value starting with [initial] value and applying [operation] from left to right
 * to current accumulator value and each element.
 * Returns the specified [initial] value if the array is empty.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 */
 public inline fun <R> ShortArray.fold(initial: R, operation: (acc: R, Short) -> R): R {
 \n var accumulator = initial
 for (element in this)
 accumulator = operation(accumulator, element)
 return accumulator
 }
 */
 * Accumulates value starting with [initial] value and applying [operation] from left to right
 * to current accumulator value and each element.
 * Returns the specified [initial] value if the array is empty.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 */
 public inline fun <R> IntArray.fold(initial: R, operation: (acc: R, Int) -> R): R {
 \n var accumulator = initial
 for (element in this)
 accumulator = operation(accumulator, element)
 return accumulator
 }
 */
 * Accumulates value starting with [initial] value and applying [operation] from left to right
 * to current accumulator value and each element.
 * Returns the specified [initial] value if the array is empty.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 */
 public inline fun <R> LongArray.fold(initial: R, operation: (acc: R, Long) -> R): R {
 \n var accumulator = initial
 for (element in this)
 accumulator = operation(accumulator, element)
 return accumulator
 }
 */
 * Accumulates value starting with [initial] value and applying [operation] from left to right
 * to current accumulator value and each element.
 * Returns the specified [initial] value if the array is empty.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 */
 public inline fun <R> FloatArray.fold(initial: R, operation: (acc: R, Float) -> R): R {
 \n var accumulator = initial
 for (element in this)
 accumulator = operation(accumulator, element)
 return accumulator
 }
 */
 * Accumulates value starting with [initial] value and applying [operation] from left to right
 * to current accumulator value and each element.
 * Returns the specified [initial] value if the array is empty.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 */
 public inline fun <R> DoubleArray.fold(initial: R, operation: (acc: R, Double) -> R): R {
 \n var accumulator = initial
 for (element in this)
 accumulator = operation(accumulator, element)
 return accumulator
 }
 */
 * Accumulates value starting with [initial] value and applying [operation] from left to right
 * to current accumulator value and each element.
 * Returns the specified [initial] value if the array is empty.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 */
 public inline fun <R> BooleanArray.fold(initial: R, operation: (acc: R, Boolean) -> R): R {
 \n var accumulator = initial
 for (element in this)
 accumulator = operation(accumulator, element)
 return accumulator
 }
 */
 * Accumulates value starting with [initial] value and applying [operation] from left to right
 * to current accumulator value and each element.
 * Returns the specified [initial] value if the array is empty.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 */
 public inline fun <R> CharArray.fold(initial: R, operation: (acc: R, Char) -> R): R {
 \n var accumulator = initial
 for (element in this)
 accumulator = operation(accumulator, element)
 return accumulator
 }
 */
 * Accumulates value starting with [initial] value and applying [operation] from left to right
 * to current accumulator value and each element with its index in the original array.
 * Returns the specified [initial] value if the array is empty.
 * @param [operation] function that takes the index of an element, current accumulator value
 * and

```

the element itself, and calculates the next accumulator value.
public inline fun <T, R> Array<out
T>.foldIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): R {
    var index = 0
    var accumulator =
initial
    for (element in this) accumulator = operation(index++, accumulator, element)
    return
accumulator
}
 Accumulates value starting with [initial] value and applying [operation] from left to
right
 to current accumulator value and each element with its index in the original array.
 Returns the
specified [initial] value if the array is empty.
 @param [operation] function that takes the index of an
element, current accumulator value
 and the element itself, and calculates the next accumulator value.
public inline fun <R> ByteArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Byte) -> R): R {
    var
index = 0
    var accumulator = initial
    for (element in this) accumulator = operation(index++, accumulator,
element)
    return accumulator
}
 Accumulates value starting with [initial] value and applying
[operation] from left to right
 to current accumulator value and each element with its index in the original array.
 Returns the
specified [initial] value if the array is empty.
 @param [operation] function that takes the
index of an element, current accumulator value
 and the element itself, and calculates the next accumulator
value.
public inline fun <R> ShortArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Short) -> R): R {
    var
index = 0
    var accumulator = initial
    for (element in this) accumulator = operation(index++,
accumulator, element)
    return accumulator
}
 Accumulates value starting with [initial] value and
applying [operation] from left to right
 to current accumulator value and each element with its index in the
original array.
 Returns the
specified [initial] value if the array is empty.
 @param [operation]
function that takes the index of an element, current accumulator value
 and the element itself, and calculates the
next accumulator value.
public inline fun <R> IntArray.foldIndexed(initial: R, operation: (index: Int, acc: R,
Int) -> R): R {
    var index = 0
    var accumulator = initial
    for (element in this) accumulator =
operation(index++, accumulator, element)
    return accumulator
}
 Accumulates value starting with
[initial] value and applying [operation] from left to right
 to current accumulator value and each element with its
index in the original array.
 Returns the
specified [initial] value if the array is empty.
 @param
[operation] function that takes the index of an element, current accumulator value
 and the element itself, and
calculates the next accumulator value.
public inline fun <R> LongArray.foldIndexed(initial: R, operation:
(index: Int, acc: R, Long) -> R): R {
    var index = 0
    var accumulator = initial
    for (element in this)
accumulator = operation(index++, accumulator, element)
    return accumulator
}
 Accumulates value
starting with [initial] value and applying [operation] from left to right
 to current accumulator value and each
element with its index in the original array.
 Returns the
specified [initial] value if the array is empty.
 @param
[operation] function that takes the index of an element, current accumulator value
 and the element
itself, and calculates the next accumulator value.
public inline fun <R> FloatArray.foldIndexed(initial: R,
operation: (index: Int, acc: R, Float) -> R): R {
    var index = 0
    var accumulator = initial
    for (element in
this) accumulator = operation(index++, accumulator, element)
    return accumulator
}
 Accumulates
value starting with [initial] value and applying [operation] from left to right
 to current accumulator value and
each element with its index in the original array.
 Returns the
specified [initial] value if the array is empty.
 @param
[operation] function that takes the index of an element, current accumulator value
 and the
element itself, and calculates the next accumulator value.
public inline fun <R>
DoubleArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Double) -> R): R {
    var index = 0
    var
accumulator = initial
    for (element in this) accumulator = operation(index++, accumulator, element)
    return
accumulator
}
 Accumulates value starting with [initial] value and applying [operation] from left to
right
 to current accumulator value and each element with its index in the original array.
 Returns the
specified [initial] value if the array is empty.
 @param [operation] function that takes the index of an
element, current accumulator value
 and the element itself, and calculates the next accumulator value.
public inline fun <R> BooleanArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Boolean) -> R): R {
    var
index = 0
    var accumulator = initial
    for (element in this) accumulator = operation(index++, accumulator,
element)
    return accumulator
}
 Accumulates value starting with [initial] value and applying
[operation] from left to right
 to current accumulator value and each element with its index in the original array.

```

* \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n */\npublic inline fun <R> CharArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Char) -> R): R {\n var index = 0\n var accumulator = initial\n for (element in this) accumulator = operation(index++, accumulator, element)\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value.\n */\npublic inline fun <T, R> Array<out T>.foldRight(initial: R, operation: (T, acc: R) -> R): R {\n var index = lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator = operation(get(index--), accumulator)\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value.\n */\npublic inline fun <R> ByteArray.foldRight(initial: R, operation: (Byte, acc: R) -> R): R {\n var index = lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator = operation(get(index--), accumulator)\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value.\n */\npublic inline fun <R> ShortArray.foldRight(initial: R, operation: (Short, acc: R) -> R): R {\n var index = lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator = operation(get(index--), accumulator)\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value.\n */\npublic inline fun <R> IntArray.foldRight(initial: R, operation: (Int, acc: R) -> R): R {\n var index = lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator = operation(get(index--), accumulator)\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value.\n */\npublic inline fun <R> LongArray.foldRight(initial: R, operation: (Long, acc: R) -> R): R {\n var index = lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator = operation(get(index--), accumulator)\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value.\n */\npublic inline fun <R> FloatArray.foldRight(initial: R, operation: (Float, acc: R) -> R): R {\n var index = lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator = operation(get(index--), accumulator)\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value.\n */\npublic inline fun <R> DoubleArray.foldRight(initial: R, operation: (Double, acc: R) -> R): R {\n var index = lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator = operation(get(index--), accumulator)\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value.\n */\npublic inline fun <R> BooleanArray.foldRight(initial: R, operation: (Boolean, acc: R) -> R): R {\n var index = lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator =

```

operation(get(index--), accumulator)\n } \n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value.\n */\npublic inline fun <R>
CharArray.foldRight(initial: R, operation: (Char, acc: R) -> R): R {\n var index = lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator = operation(get(index--), accumulator)\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself\n * and current accumulator value, and calculates the next accumulator value.\n */\npublic inline fun <T, R> Array<out T>.foldRightIndexed(initial: R, operation: (index: Int, T, acc: R) -> R): R {\n var index = lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator = operation(index, get(index), accumulator)\n --index\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself\n * and current accumulator value, and calculates the next accumulator value.\n */\npublic inline fun <R> ByteArray.foldRightIndexed(initial: R, operation: (index: Int, Byte, acc: R) -> R): R {\n var index = lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator = operation(index, get(index), accumulator)\n --index\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself\n * and current accumulator value, and calculates the next accumulator value.\n */\npublic inline fun <R> ShortArray.foldRightIndexed(initial: R, operation: (index: Int, Short, acc: R) -> R): R {\n var index = lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator = operation(index, get(index), accumulator)\n --index\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself\n * and current accumulator value, and calculates the next accumulator value.\n */\npublic inline fun <R> IntArray.foldRightIndexed(initial: R, operation: (index: Int, Int, acc: R) -> R): R {\n var index = lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator = operation(index, get(index), accumulator)\n --index\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself\n * and current accumulator value, and calculates the next accumulator value.\n */\npublic inline fun <R> LongArray.foldRightIndexed(initial: R, operation: (index: Int, Long, acc: R) -> R): R {\n var index = lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator = operation(index, get(index), accumulator)\n --index\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself\n * and current accumulator value, and calculates the next accumulator value.\n */\npublic inline fun <R> FloatArray.foldRightIndexed(initial: R, operation: (index: Int, Float, acc: R) -> R): R {\n var index = lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator = operation(index, get(index), accumulator)\n --index\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself\n * and current accumulator

```

value, and calculates the next accumulator value.

```

public inline fun <R> DoubleArray.foldRightIndexed(initial:
R, operation: (index: Int, Double, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

```

Accumulates value starting with [initial] value and applying [operation] from right to left to each element with its index in the original array and current accumulator value. Returns the specified [initial] value if the array is empty. @param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.

```

public inline fun <R> BooleanArray.foldRightIndexed(initial: R, operation: (index: Int, Boolean, acc: R) -> R):
R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator =
operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

```

Accumulates value starting with [initial] value and applying [operation] from right to left to each element with its index in the original array and current accumulator value. Returns the specified [initial] value if the array is empty. @param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.

```

public inline fun <R> CharArray.foldRightIndexed(initial: R,
operation: (index: Int, Char, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

```

Performs the given [action] on each element.

```

public inline fun <T> Array<out T>.forEach(action: (T) -> Unit): Unit {
    for (element in this) action(element)
}

```

Performs the given [action] on each element.

```

public inline fun ByteArray.forEach(action: (Byte) -> Unit): Unit {
    for (element in this) action(element)
}

```

Performs the given [action] on each element.

```

public inline fun ShortArray.forEach(action: (Short) -> Unit): Unit {
    for (element in this) action(element)
}

```

Performs the given [action] on each element.

```

public inline fun IntArray.forEach(action: (Int) -> Unit): Unit {
    for (element in this) action(element)
}

```

Performs the given [action] on each element.

```

public inline fun LongArray.forEach(action: (Long) -> Unit): Unit {
    for (element in this) action(element)
}

```

Performs the given [action] on each element.

```

public inline fun FloatArray.forEach(action: (Float) -> Unit): Unit {
    for (element in this) action(element)
}

```

Performs the given [action] on each element.

```

public inline fun DoubleArray.forEach(action: (Double) -> Unit): Unit {
    for (element in this) action(element)
}

```

Performs the given [action] on each element.

```

public inline fun BooleanArray.forEach(action: (Boolean) ->
Unit): Unit {
    for (element in this) action(element)
}

```

Performs the given [action] on each element.

```

public inline fun CharArray.forEach(action: (Char) -> Unit): Unit {
    for (element in this)
action(element)
}

```

Performs the given [action] on each element, providing sequential index with the element. @param [action] function that takes the index of an element and the element itself and performs the action on the element.

```

public inline fun <T> Array<out T>.forEachIndexed(action: (index: Int, T) -> Unit):
Unit {
    var index = 0
    for (item in this) action(index++, item)
}

```

Performs the given [action] on each element, providing sequential index with the element. @param [action] function that takes the index of an element and the element itself and performs the action on the element.

```

public inline fun ByteArray.forEachIndexed(action: (index: Int, Byte) -> Unit): Unit {
    var index = 0
    for (item in this)
action(index++, item)
}

```

Performs the given [action] on each element, providing sequential index with the element. @param [action] function that takes the index of an element and the element itself and performs the action on the element.

```

public inline fun ShortArray.forEachIndexed(action: (index: Int, Short) -> Unit):
Unit {
    var index = 0
    for (item in this) action(index++, item)
}

```

Performs the given [action] on each element, providing sequential index with the element. @param [action] function that takes the index of an element and the element itself and performs the action on the element.

```

public inline fun IntArray.forEachIndexed(action: (index: Int, Int) -> Unit): Unit {
    var index = 0
    for (item in this)
action(index++, item)
}

```

Performs the given [action] on each element, providing sequential index with the element. @param [action] function that takes the index of an element and the element itself and performs the action on the element.

```

public inline fun LongArray.forEachIndexed(action: (index: Int, Long) -> Unit):
Unit {
    var index = 0
    for (item in this) action(index++, item)
}

```

Performs the given [action] on

each element, providing sequential index with the element.\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the element.\n */\npublic inline fun

FloatArray.forEachIndexed(action: (index: Int, Float) -> Unit): Unit {\n var index = 0\n for (item in this) action(index++, item)\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element.\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the element.\n */\npublic inline fun DoubleArray.forEachIndexed(action: (index: Int, Double) -> Unit): Unit {\n var index = 0\n for (item in this) action(index++, item)\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element.\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the element.\n */\npublic inline fun BooleanArray.forEachIndexed(action: (index: Int, Boolean) -> Unit): Unit {\n var index = 0\n for (item in this) action(index++, item)\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element.\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the element.\n */\npublic inline fun CharArray.forEachIndexed(action: (index: Int, Char) -> Unit): Unit {\n var index = 0\n for (item in this) action(index++, item)\n}\n\n@Deprecated("Use maxOrNull instead.", ReplaceWith("this.maxOrNull()"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5", hiddenSince = "1.6")\n@SinceKotlin("1.1")\npublic fun Array<out Double>.max(): Double? {\n return maxOrNull()\n}\n\n@Deprecated("Use maxOrNull instead.", ReplaceWith("this.maxOrNull()"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5", hiddenSince = "1.6")\n@SinceKotlin("1.1")\npublic fun Array<out Float>.max(): Float? {\n return maxOrNull()\n}\n\n@Deprecated("Use maxOrNull instead.", ReplaceWith("this.maxOrNull()"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5", hiddenSince = "1.6")\npublic fun <T : Comparable<T>> Array<out T>.max(): T? {\n return maxOrNull()\n}\n\n@Deprecated("Use maxOrNull instead.", ReplaceWith("this.maxOrNull()"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5", hiddenSince = "1.6")\npublic fun ByteArray.max(): Byte? {\n return maxOrNull()\n}\n\n@Deprecated("Use maxOrNull instead.", ReplaceWith("this.maxOrNull()"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5", hiddenSince = "1.6")\npublic fun ShortArray.max(): Short? {\n return maxOrNull()\n}\n\n@Deprecated("Use maxOrNull instead.", ReplaceWith("this.maxOrNull()"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5", hiddenSince = "1.6")\npublic fun IntArray.max(): Int? {\n return maxOrNull()\n}\n\n@Deprecated("Use maxOrNull instead.", ReplaceWith("this.maxOrNull()"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5", hiddenSince = "1.6")\npublic fun LongArray.max(): Long? {\n return maxOrNull()\n}\n\n@Deprecated("Use maxOrNull instead.", ReplaceWith("this.maxOrNull()"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5", hiddenSince = "1.6")\npublic fun FloatArray.max(): Float? {\n return maxOrNull()\n}\n\n@Deprecated("Use maxOrNull instead.", ReplaceWith("this.maxOrNull()"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5", hiddenSince = "1.6")\npublic fun DoubleArray.max(): Double? {\n return maxOrNull()\n}\n\n@Deprecated("Use maxOrNull instead.", ReplaceWith("this.maxOrNull()"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5", hiddenSince = "1.6")\npublic fun CharArray.max(): Char? {\n return maxOrNull()\n}\n\n@Deprecated("Use maxByOrNull instead.", ReplaceWith("this.maxByOrNull(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5", hiddenSince = "1.6")\npublic inline fun <T, R : Comparable<R>> Array<out T>.maxBy(selector: (T) -> R): T? {\n return maxByOrNull(selector)\n}\n\n@Deprecated("Use maxByOrNull instead.", ReplaceWith("this.maxByOrNull(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5", hiddenSince = "1.6")\npublic inline fun <R : Comparable<R>> ByteArray.maxBy(selector: (Byte) -> R): Byte? {\n return maxByOrNull(selector)\n}\n\n@Deprecated("Use maxByOrNull instead.", ReplaceWith("this.maxByOrNull(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5", hiddenSince = "1.6")\npublic inline fun <R : Comparable<R>> ShortArray.maxBy(selector: (Short) -> R):

```

Short? {\n  return maxByOrNull(selector)\n}\n\n@Deprecated("Use maxByOrNull instead.\",
ReplaceWith("this.maxByOrNull(selector)")\n)\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince =
"1.5", hiddenSince = "1.6")\npublic inline fun <R : Comparable<R>> IntArray.maxBy(selector: (Int) -> R): Int?
{\n  return maxByOrNull(selector)\n}\n\n@Deprecated("Use maxByOrNull instead.\",
ReplaceWith("this.maxByOrNull(selector)")\n)\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince =
"1.5", hiddenSince = "1.6")\npublic inline fun <R : Comparable<R>> LongArray.maxBy(selector: (Long) -> R):
Long? {\n  return maxByOrNull(selector)\n}\n\n@Deprecated("Use maxByOrNull instead.\",
ReplaceWith("this.maxByOrNull(selector)")\n)\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince =
"1.5", hiddenSince = "1.6")\npublic inline fun <R : Comparable<R>> FloatArray.maxBy(selector: (Float) -> R):
Float? {\n  return maxByOrNull(selector)\n}\n\n@Deprecated("Use maxByOrNull instead.\",
ReplaceWith("this.maxByOrNull(selector)")\n)\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince =
"1.5", hiddenSince = "1.6")\npublic inline fun <R : Comparable<R>> DoubleArray.maxBy(selector: (Double) ->
R): Double? {\n  return maxByOrNull(selector)\n}\n\n@Deprecated("Use maxByOrNull instead.\",
ReplaceWith("this.maxByOrNull(selector)")\n)\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince =
"1.5", hiddenSince = "1.6")\npublic inline fun <R : Comparable<R>> BooleanArray.maxBy(selector: (Boolean) -
> R): Boolean? {\n  return maxByOrNull(selector)\n}\n\n@Deprecated("Use maxByOrNull instead.\",
ReplaceWith("this.maxByOrNull(selector)")\n)\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince =
"1.5", hiddenSince = "1.6")\npublic inline fun <R : Comparable<R>> CharArray.maxBy(selector: (Char) -> R):
Char? {\n  return maxByOrNull(selector)\n}\n\n/**\n * Returns the first element yielding the largest value of the
given function or `null` if there are no elements.\n * \n * @sample
samples.collections.Collections.Aggregates.maxByOrNull\n * \n * @SinceKotlin("1.4")\n * \n * public inline fun <T, R :
Comparable<R>> Array<out T>.maxByOrNull(selector: (T) -> R): T? {\n  if (isEmpty()) return null\n  var
maxElem = this[0]\n  val lastIndex = this.lastIndex\n  if (lastIndex == 0) return maxElem\n  var maxValue =
selector(maxElem)\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    val v = selector(e)\n    if (maxValue < v)
{\n      maxElem = e\n      maxValue = v\n    }\n  }\n  return maxElem\n}\n\n/**\n * Returns the first
element yielding the largest value of the given function or `null` if there are no elements.\n * \n * @sample
samples.collections.Collections.Aggregates.maxByOrNull\n * \n * @SinceKotlin("1.4")\n * \n * public inline fun <R :
Comparable<R>> ByteArray.maxByOrNull(selector: (Byte) -> R): Byte? {\n  if (isEmpty()) return null\n  var
maxElem = this[0]\n  val lastIndex = this.lastIndex\n  if (lastIndex == 0) return maxElem\n  var maxValue =
selector(maxElem)\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    val v = selector(e)\n    if (maxValue < v)
{\n      maxElem = e\n      maxValue = v\n    }\n  }\n  return maxElem\n}\n\n/**\n * Returns the first
element yielding the largest value of the given function or `null` if there are no elements.\n * \n * @sample
samples.collections.Collections.Aggregates.maxByOrNull\n * \n * @SinceKotlin("1.4")\n * \n * public inline fun <R :
Comparable<R>> ShortArray.maxByOrNull(selector: (Short) -> R): Short? {\n  if (isEmpty()) return null\n  var
maxElem = this[0]\n  val lastIndex = this.lastIndex\n  if (lastIndex == 0) return maxElem\n  var maxValue =
selector(maxElem)\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    val v = selector(e)\n    if (maxValue < v)
{\n      maxElem = e\n      maxValue = v\n    }\n  }\n  return maxElem\n}\n\n/**\n * Returns the first
element yielding the largest value of the given function or `null` if there are no elements.\n * \n * @sample
samples.collections.Collections.Aggregates.maxByOrNull\n * \n * @SinceKotlin("1.4")\n * \n * public inline fun <R :
Comparable<R>> IntArray.maxByOrNull(selector: (Int) -> R): Int? {\n  if (isEmpty()) return null\n  var maxElem
= this[0]\n  val lastIndex = this.lastIndex\n  if (lastIndex == 0) return maxElem\n  var maxValue =
selector(maxElem)\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    val v = selector(e)\n    if (maxValue < v)
{\n      maxElem = e\n      maxValue = v\n    }\n  }\n  return maxElem\n}\n\n/**\n * Returns the first
element yielding the largest value of the given function or `null` if there are no elements.\n * \n * @sample
samples.collections.Collections.Aggregates.maxByOrNull\n * \n * @SinceKotlin("1.4")\n * \n * public inline fun <R :
Comparable<R>> LongArray.maxByOrNull(selector: (Long) -> R): Long? {\n  if (isEmpty()) return null\n  var
maxElem = this[0]\n  val lastIndex = this.lastIndex\n  if (lastIndex == 0) return maxElem\n  var maxValue =
selector(maxElem)\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    val v = selector(e)\n    if (maxValue < v)

```

```

{\n      maxElem = e\n      maxValue = v\n    }\n }\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function or `null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.maxByOrNull\n */\n@SinceKotlin("1.4")\npublic inline fun <R : Comparable<R>> FloatArray.maxByOrNull(selector: (Float) -> R): Float? {\n    if (isEmpty()) return null\n    var maxElem = this[0]\n    val lastIndex = this.lastIndex\n    if (lastIndex == 0) return maxElem\n    var maxValue = selector(maxElem)\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        val v = selector(e)\n        if (maxValue < v) {\n            maxElem = e\n            maxValue = v\n        }\n    }\n }\n }\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function or `null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.maxByOrNull\n */\n@SinceKotlin("1.4")\npublic inline fun <R : Comparable<R>> DoubleArray.maxByOrNull(selector: (Double) -> R): Double? {\n    if (isEmpty()) return null\n    var maxElem = this[0]\n    val lastIndex = this.lastIndex\n    if (lastIndex == 0) return maxElem\n    var maxValue = selector(maxElem)\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        val v = selector(e)\n        if (maxValue < v) {\n            maxElem = e\n            maxValue = v\n        }\n    }\n }\n }\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function or `null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.maxByOrNull\n */\n@SinceKotlin("1.4")\npublic inline fun <R : Comparable<R>> BooleanArray.maxByOrNull(selector: (Boolean) -> R): Boolean? {\n    if (isEmpty()) return null\n    var maxElem = this[0]\n    val lastIndex = this.lastIndex\n    if (lastIndex == 0) return maxElem\n    var maxValue = selector(maxElem)\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        val v = selector(e)\n        if (maxValue < v) {\n            maxElem = e\n            maxValue = v\n        }\n    }\n }\n }\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function or `null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.maxByOrNull\n */\n@SinceKotlin("1.4")\npublic inline fun <R : Comparable<R>> CharArray.maxByOrNull(selector: (Char) -> R): Char? {\n    if (isEmpty()) return null\n    var maxElem = this[0]\n    val lastIndex = this.lastIndex\n    if (lastIndex == 0) return maxElem\n    var maxValue = selector(maxElem)\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        val v = selector(e)\n        if (maxValue < v) {\n            maxElem = e\n            maxValue = v\n        }\n    }\n }\n }\n return maxElem\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.maxOf(selector: (T) -> Double): Double {\n    if (isEmpty()) throw NoSuchElementException()\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.maxOf(selector: (Byte) -> Double): Double {\n    if (isEmpty()) throw NoSuchElementException()\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.maxOf(selector: (Short) -> Double): Double {\n    if (isEmpty()) throw NoSuchElementException()\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is

```

```

`NaN`.n * .n * @throws NoSuchElementException if the array is empty.n
*/n@SinceKotlin("1.4")n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)n@OverloadResolution
ByLambdaReturnTypen@kotlin.internal.InlineOnlynpublic inline fun IntArray.maxOf(selector: (Int) -> Double):
Double {n if (isEmpty()) throw NoSuchElementException()n var maxValue = selector(this[0])n for (i in
1..lastIndex) {n val v = selector(this[i])n maxValue = maxOf(maxValue, v)n }n return
maxValue\n}\n/n/**n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the array.n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.n * \n * @throws NoSuchElementException if the array is empty.n
*/n@SinceKotlin("1.4")n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)n@OverloadResolution
ByLambdaReturnTypen@kotlin.internal.InlineOnlynpublic inline fun LongArray.maxOf(selector: (Long) ->
Double): Double {n if (isEmpty()) throw NoSuchElementException()n var maxValue = selector(this[0])n for
(i in 1..lastIndex) {n val v = selector(this[i])n maxValue = maxOf(maxValue, v)n }n return
maxValue\n}\n/n/**n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the array.n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.n * \n * @throws NoSuchElementException if the array is empty.n
*/n@SinceKotlin("1.4")n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)n@OverloadResolution
ByLambdaReturnTypen@kotlin.internal.InlineOnlynpublic inline fun FloatArray.maxOf(selector: (Float) ->
Double): Double {n if (isEmpty()) throw NoSuchElementException()n var maxValue = selector(this[0])n for
(i in 1..lastIndex) {n val v = selector(this[i])n maxValue = maxOf(maxValue, v)n }n return
maxValue\n}\n/n/**n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the array.n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.n * \n * @throws NoSuchElementException if the array is empty.n
*/n@SinceKotlin("1.4")n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)n@OverloadResolution
ByLambdaReturnTypen@kotlin.internal.InlineOnlynpublic inline fun DoubleArray.maxOf(selector: (Double) ->
Double): Double {n if (isEmpty()) throw NoSuchElementException()n var maxValue = selector(this[0])n for
(i in 1..lastIndex) {n val v = selector(this[i])n maxValue = maxOf(maxValue, v)n }n return
maxValue\n}\n/n/**n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the array.n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.n * \n * @throws NoSuchElementException if the array is empty.n
*/n@SinceKotlin("1.4")n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)n@OverloadResolution
ByLambdaReturnTypen@kotlin.internal.InlineOnlynpublic inline fun BooleanArray.maxOf(selector: (Boolean) ->
Double): Double {n if (isEmpty()) throw NoSuchElementException()n var maxValue = selector(this[0])n for
(i in 1..lastIndex) {n val v = selector(this[i])n maxValue = maxOf(maxValue, v)n }n return
maxValue\n}\n/n/**n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the array.n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.n * \n * @throws NoSuchElementException if the array is empty.n
*/n@SinceKotlin("1.4")n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)n@OverloadResolution
ByLambdaReturnTypen@kotlin.internal.InlineOnlynpublic inline fun CharArray.maxOf(selector: (Char) ->
Double): Double {n if (isEmpty()) throw NoSuchElementException()n var maxValue = selector(this[0])n for
(i in 1..lastIndex) {n val v = selector(this[i])n maxValue = maxOf(maxValue, v)n }n return
maxValue\n}\n/n/**n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the array.n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.n * \n * @throws NoSuchElementException if the array is empty.n
*/n@SinceKotlin("1.4")n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)n@OverloadResolution
ByLambdaReturnTypen@kotlin.internal.InlineOnlynpublic inline fun <T> Array<out T>.maxOf(selector: (T) ->
Float): Float {n if (isEmpty()) throw NoSuchElementException()n var maxValue = selector(this[0])n for (i
in 1..lastIndex) {n val v = selector(this[i])n maxValue = maxOf(maxValue, v)n }n return
maxValue\n}\n/n/**n * Returns the largest value among all values produced by [selector] function\n * applied to

```

each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.maxOf(selector: (Byte) -> Float):
Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var maxValue = selector(this[0])\n    for (i in
1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return
maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.maxOf(selector: (Short) ->
Float): Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var maxValue = selector(this[0])\n    for (i
in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return
maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.maxOf(selector: (Int) -> Float):
Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var maxValue = selector(this[0])\n    for (i in
1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return
maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.maxOf(selector: (Long) ->
Float): Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var maxValue = selector(this[0])\n    for (i
in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return
maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.maxOf(selector: (Float) ->
Float): Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var maxValue = selector(this[0])\n    for (i
in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return
maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.maxOf(selector: (Double) ->
Float): Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var maxValue = selector(this[0])\n    for (i
in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return
maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.maxOf(selector: (Boolean) ->
Float): Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var maxValue = selector(this[0])\n    for (i
in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return
maxValue\n}

```



```

    maxValue = v\n    }\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values
produced by [selector] function\n * applied to each element in the array.\n * \n * @throws
NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
DoubleArray.maxOf(selector: (Double) -> R): R {\n    if (isEmpty()) throw NoSuchElementException()\n    var
maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (maxValue < v) {\n
            maxValue = v\n        }\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values
produced by [selector] function\n * applied to each element in the array.\n * \n * @throws
NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
BooleanArray.maxOf(selector: (Boolean) -> R): R {\n    if (isEmpty()) throw NoSuchElementException()\n    var
maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (maxValue < v) {\n
            maxValue = v\n        }\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values
produced by [selector] function\n * applied to each element in the array.\n * \n * @throws
NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
CharArray.maxOf(selector: (Char) -> R): R {\n    if (isEmpty()) throw NoSuchElementException()\n    var
maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (maxValue < v) {\n
            maxValue = v\n        }\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values
produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If
any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.maxOfOrNull(selector:
(T) -> Double): Double? {\n    if (isEmpty()) return null\n    var maxValue = selector(this[0])\n    for (i in
1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return
maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function
is `NaN`, the returned result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.maxOfOrNull(selector: (Byte) ->
Double): Double? {\n    if (isEmpty()) return null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n
        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns
the largest value among all values produced by [selector] function\n * applied to each element in the array or `null`
if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.maxOfOrNull(selector: (Short) -
> Double): Double? {\n    if (isEmpty()) return null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex)
{\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns
the largest value among all values produced by [selector] function\n * applied to each element in the array
or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned
result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.maxOfOrNull(selector: (Int) ->
Double): Double? {\n    if (isEmpty()) return null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n

```

val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n } return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.maxOrNull(selector: (Long) -> Double): Double? {\n    if (isEmpty()) return null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.maxOrNull(selector: (Float) -> Double): Double? {\n    if (isEmpty()) return null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.maxOrNull(selector: (Double) -> Double): Double? {\n    if (isEmpty()) return null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.maxOrNull(selector: (Boolean) -> Double): Double? {\n    if (isEmpty()) return null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.maxOrNull(selector: (Char) -> Double): Double? {\n    if (isEmpty()) return null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.maxOrNull(selector: (T) -> Float): Float? {\n    if (isEmpty()) return null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.maxOrNull(selector: (Byte) ->
```

Float): Float? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.maxOfOrNull(selector: (Short) -> Float): Float? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.maxOfOrNull(selector: (Int) -> Float): Float? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.maxOfOrNull(selector: (Long) -> Float): Float? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.maxOfOrNull(selector: (Float) -> Float): Float? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.maxOfOrNull(selector: (Double) -> Float): Float? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.maxOfOrNull(selector: (Boolean) -> Float): Float? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
```

ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.maxOrNull(selector: (Char) -> Float): Float? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Comparable<R>> Array<out T>.maxOrNull(selector: (T) -> R): R? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (maxValue < v) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
ByteArray.maxOrNull(selector: (Byte) -> R): R? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (maxValue < v) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
ShortArray.maxOrNull(selector: (Short) -> R): R? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (maxValue < v) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
IntArray.maxOrNull(selector: (Int) -> R): R? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (maxValue < v) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
LongArray.maxOrNull(selector: (Long) -> R): R? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (maxValue < v) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
FloatArray.maxOrNull(selector: (Float) -> R): R? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (maxValue < v) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
DoubleArray.maxOrNull(selector: (Double) -> R): R? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (maxValue < v) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution

```

ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
BooleanArray.maxOfOrNull(selector: (Boolean) -> R): R? {\n  if (isEmpty()) return null\n  var maxValue =
selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if (maxValue < v) {\n
maxValue = v\n    }\n  }\n  return maxValue\n}\n\n/**\n * Returns the largest value among all values produced
by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
*\n *\n @SinceKotlin("1.4")\n @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n @OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
CharArray.maxOfOrNull(selector: (Char) -> R): R? {\n  if (isEmpty()) return null\n  var maxValue =
selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if (maxValue < v) {\n
maxValue = v\n    }\n  }\n  return maxValue\n}\n\n/**\n * Returns the largest value according to the provided
[comparator]\n * among all values produced by [selector] function applied to each element in the array.\n * \n *
@throws NoSuchElementException if the array is empty.\n
*\n *\n @SinceKotlin("1.4")\n @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n @OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R> Array<out
T>.maxOfWith(comparator: Comparator<in R>, selector: (T) -> R): R {\n  if (isEmpty()) throw
NoSuchElementException()\n  var maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v =
selector(this[i])\n    if (comparator.compare(maxValue, v) < 0) {\n      maxValue = v\n    }\n  }\n  return
maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values
produced by [selector] function applied to each element in the array.\n * \n * @throws NoSuchElementException if
the array is empty.\n
*\n *\n @SinceKotlin("1.4")\n @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n @OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> ByteArray.maxOfWith(comparator:
Comparator<in R>, selector: (Byte) -> R): R {\n  if (isEmpty()) throw NoSuchElementException()\n  var
maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if
(comparator.compare(maxValue, v) < 0) {\n      maxValue = v\n    }\n  }\n  return maxValue\n}\n\n/**\n *
Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n *\n @SinceKotlin("1.4")\n @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n @OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> ShortArray.maxOfWith(comparator:
Comparator<in R>, selector: (Short) -> R): R {\n  if (isEmpty()) throw NoSuchElementException()\n  var
maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if
(comparator.compare(maxValue, v) < 0) {\n      maxValue = v\n    }\n  }\n  return maxValue\n}\n\n/**\n *
Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n *\n @SinceKotlin("1.4")\n @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n @OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> IntArray.maxOfWith(comparator:
Comparator<in R>, selector: (Int) -> R): R {\n  if (isEmpty()) throw NoSuchElementException()\n  var maxValue
= selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if
(comparator.compare(maxValue, v) < 0) {\n      maxValue = v\n    }\n  }\n  return maxValue\n}\n\n/**\n *
Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n *\n @SinceKotlin("1.4")\n @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n @OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> LongArray.maxOfWith(comparator:
Comparator<in R>, selector: (Long) -> R): R {\n  if (isEmpty()) throw NoSuchElementException()\n  var
maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if
(comparator.compare(maxValue, v) < 0) {\n      maxValue = v\n    }\n  }\n  return maxValue\n}\n\n/**\n *
Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n

```

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> FloatArray.maxOfWith(comparator:
Comparator<in R>, selector: (Float) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var
maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n *
Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> DoubleArray.maxOfWith(comparator:
Comparator<in R>, selector: (Double) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var
maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n *
Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> BooleanArray.maxOfWith(comparator:
Comparator<in R>, selector: (Boolean) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var
maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n *
Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> CharArray.maxOfWith(comparator:
Comparator<in R>, selector: (Char) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var
maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n *
Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R> Array<out
T>.maxOfWithOrNull(comparator: Comparator<in R>, selector: (T) -> R): R? {\n if (isEmpty()) return null\n
var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n *
Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>
ByteArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Byte) -> R): R? {\n if (isEmpty()) return
null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n *
Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>
ShortArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Short) -> R): R? {\n if (isEmpty())
return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n *
Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]

```

```

function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>
IntArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Int) -> R): R? {\n if (isEmpty()) return
null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n *
Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>
LongArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Long) -> R): R? {\n if (isEmpty())
return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n *
Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>
FloatArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Float) -> R): R? {\n if (isEmpty())
return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n *
Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>
DoubleArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Double) -> R): R? {\n if (isEmpty())
return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n *
Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>
BooleanArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Boolean) -> R): R? {\n if (isEmpty())
return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n *
Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>
CharArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Char) -> R): R? {\n if (isEmpty()) return
null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n *
Returns the largest element or `null` if there are no elements.\n * \n * If any of elements is `NaN` returns `NaN`.\n
*\n@SinceKotlin("1.4")\npublic fun Array<out Double>.maxOrNull(): Double? {\n if (isEmpty()) return null\n
var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n max = maxOf(max, e)\n }\n return
max\n}\n\n/**\n * Returns the largest element or `null` if there are no elements.\n * \n * If any of elements is `NaN`
returns `NaN`.\n
*\n@SinceKotlin("1.4")\npublic fun Array<out Float>.maxOrNull(): Float? {\n if (isEmpty())
return null\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n max = maxOf(max, e)\n }\n
return max\n}\n\n/**\n * Returns the largest element or `null` if there are no elements.\n

```

```

*^@SinceKotlin("1.4")\npublic fun <T : Comparable<T>> Array<out T>.maxOrNull(): T? {\n  if (isEmpty())\n  return null\n  var max = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if (max < e) max = e\n  }\n  return max\n}\n\n**\n * Returns the largest element or `null` if there are no elements.\n
*^@SinceKotlin("1.4")\npublic fun ByteArray.maxOrNull(): Byte? {\n  if (isEmpty()) return null\n  var max =\n  this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if (max < e) max = e\n  }\n  return max\n}\n\n**\n * Returns the largest element or `null` if there are no elements.\n
*^@SinceKotlin("1.4")\npublic fun\nShortArray.maxOrNull(): Short? {\n  if (isEmpty()) return null\n  var max = this[0]\n  for (i in 1..lastIndex) {\n\n    val e = this[i]\n    if (max < e) max = e\n  }\n  return max\n}\n\n**\n * Returns the largest element or `null` if\n  there are no elements.\n
*^@SinceKotlin("1.4")\npublic fun IntArray.maxOrNull(): Int? {\n  if (isEmpty())\n  return null\n  var max = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if (max < e) max = e\n  }\n  return max\n}\n\n**\n * Returns the largest element or `null` if there are no elements.\n
*^@SinceKotlin("1.4")\npublic fun LongArray.maxOrNull(): Long? {\n  if (isEmpty()) return null\n  var max\n  = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if (max < e) max = e\n  }\n  return max\n}\n\n**\n * Returns the largest element or `null` if there are no elements.\n * \n * If any of elements is `NaN` returns `NaN`.\n
*^@SinceKotlin("1.4")\npublic fun FloatArray.maxOrNull(): Float? {\n  if (isEmpty()) return null\n  var max\n  = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    max = maxOf(max, e)\n  }\n  return\n  max\n}\n\n**\n * Returns the largest element or `null` if there are no elements.\n * \n * If any of elements is `NaN`\n  returns `NaN`.\n
*^@SinceKotlin("1.4")\npublic fun DoubleArray.maxOrNull(): Double? {\n  if (isEmpty())\n  return null\n  var max = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    max = maxOf(max, e)\n  }\n  return\n  max\n}\n\n**\n * Returns the largest element or `null` if there are no elements.\n
*^@SinceKotlin("1.4")\npublic fun CharArray.maxOrNull(): Char? {\n  if (isEmpty()) return null\n  var max =\n  this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if (max < e) max = e\n  }\n  return\n  max\n}\n\n@Deprecated("Use maxWithOrNull instead.")\n  ReplaceWith("this.maxWithOrNull(comparator)")\n}@DeprecatedSinceKotlin(warningSince = "1.4", errorSince\n  = "1.5", hiddenSince = "1.6")\npublic fun <T> Array<out T>.maxWith(comparator: Comparator<in T>): T? {\n  return\n  maxWithOrNull(comparator)\n}\n\n@Deprecated("Use maxWithOrNull instead.")\n  ReplaceWith("this.maxWithOrNull(comparator)")\n}@DeprecatedSinceKotlin(warningSince = "1.4", errorSince\n  = "1.5", hiddenSince = "1.6")\npublic fun ByteArray.maxWith(comparator: Comparator<in Byte>): Byte? {\n  return\n  maxWithOrNull(comparator)\n}\n\n@Deprecated("Use maxWithOrNull instead.")\n  ReplaceWith("this.maxWithOrNull(comparator)")\n}@DeprecatedSinceKotlin(warningSince = "1.4", errorSince\n  = "1.5", hiddenSince = "1.6")\npublic fun ShortArray.maxWith(comparator: Comparator<in Short>): Short? {\n  return\n  maxWithOrNull(comparator)\n}\n\n@Deprecated("Use maxWithOrNull instead.")\n  ReplaceWith("this.maxWithOrNull(comparator)")\n}@DeprecatedSinceKotlin(warningSince = "1.4", errorSince\n  = "1.5", hiddenSince = "1.6")\npublic fun IntArray.maxWith(comparator: Comparator<in Int>): Int? {\n  return\n  maxWithOrNull(comparator)\n}\n\n@Deprecated("Use maxWithOrNull instead.")\n  ReplaceWith("this.maxWithOrNull(comparator)")\n}@DeprecatedSinceKotlin(warningSince = "1.4", errorSince\n  = "1.5", hiddenSince = "1.6")\npublic fun LongArray.maxWith(comparator: Comparator<in Long>): Long? {\n  return\n  maxWithOrNull(comparator)\n}\n\n@Deprecated("Use maxWithOrNull instead.")\n  ReplaceWith("this.maxWithOrNull(comparator)")\n}@DeprecatedSinceKotlin(warningSince = "1.4", errorSince\n  = "1.5", hiddenSince = "1.6")\npublic fun FloatArray.maxWith(comparator: Comparator<in Float>): Float? {\n  return\n  maxWithOrNull(comparator)\n}\n\n@Deprecated("Use maxWithOrNull instead.")\n  ReplaceWith("this.maxWithOrNull(comparator)")\n}@DeprecatedSinceKotlin(warningSince = "1.4", errorSince\n  = "1.5", hiddenSince = "1.6")\npublic fun DoubleArray.maxWith(comparator: Comparator<in Double>): Double?\n  {\n  return\n  maxWithOrNull(comparator)\n}\n\n@Deprecated("Use maxWithOrNull instead.")\n  ReplaceWith("this.maxWithOrNull(comparator)")\n}@DeprecatedSinceKotlin(warningSince = "1.4", errorSince\n  = "1.5", hiddenSince = "1.6")\npublic fun BooleanArray.maxWith(comparator: Comparator<in Boolean>):\n  Boolean? {\n  return\n  maxWithOrNull(comparator)\n}\n\n@Deprecated("Use maxWithOrNull instead.")\n  ReplaceWith("this.maxWithOrNull(comparator)")\n}@DeprecatedSinceKotlin(warningSince = "1.4", errorSince

```

```

= \"1.5\", hiddenSince = \"1.6\")\npublic fun CharArray.maxWith(comparator: Comparator<in Char>): Char? {\n
return maxWithOrNull(comparator)\n}\n\n/**\n * Returns the first element having the largest value according to the
provided [comparator] or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\npublic fun <T> Array<out
T>.maxWithOrNull(comparator: Comparator<in T>): T? {\n    if (isEmpty()) return null\n    var max = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (comparator.compare(max, e) < 0) max = e\n    }\n    return
max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator] or `null`
if there are no elements.\n */\n@SinceKotlin(\"1.4\")\npublic fun ByteArray.maxWithOrNull(comparator:
Comparator<in Byte>): Byte? {\n    if (isEmpty()) return null\n    var max = this[0]\n    for (i in 1..lastIndex) {\n
        val e = this[i]\n        if (comparator.compare(max, e) < 0) max = e\n    }\n    return max\n}\n\n/**\n * Returns the
first element having the largest value according to the provided [comparator] or `null` if there are no elements.\n
*/\n@SinceKotlin(\"1.4\")\npublic fun ShortArray.maxWithOrNull(comparator: Comparator<in Short>): Short? {\n
    if (isEmpty()) return null\n    var max = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if
(comparator.compare(max, e) < 0) max = e\n    }\n    return max\n}\n\n/**\n * Returns the first element having the
largest value according to the provided [comparator] or `null` if there are no elements.\n
*/\n@SinceKotlin(\"1.4\")\npublic fun IntArray.maxWithOrNull(comparator: Comparator<in Int>): Int? {\n    if
(isEmpty()) return null\n    var max = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if
(comparator.compare(max, e) < 0) max = e\n    }\n    return max\n}\n\n/**\n * Returns the first element having the
largest value according to the provided [comparator] or `null` if there are no elements.\n
*/\n@SinceKotlin(\"1.4\")\npublic fun LongArray.maxWithOrNull(comparator: Comparator<in Long>): Long? {\n
    if (isEmpty()) return null\n    var max = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if
(comparator.compare(max, e) < 0) max = e\n    }\n    return max\n}\n\n/**\n * Returns the first element having the
largest value according to the provided [comparator] or `null` if there are no elements.\n
*/\n@SinceKotlin(\"1.4\")\npublic fun FloatArray.maxWithOrNull(comparator: Comparator<in Float>): Float? {\n
    if (isEmpty()) return null\n    var max = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if
(comparator.compare(max, e) < 0) max = e\n    }\n    return max\n}\n\n/**\n * Returns the first element having the
largest value according to the provided [comparator] or `null` if there are no elements.\n
*/\n@SinceKotlin(\"1.4\")\npublic fun DoubleArray.maxWithOrNull(comparator: Comparator<in Double>):
Double? {\n    if (isEmpty()) return null\n    var max = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n
        if (comparator.compare(max, e) < 0) max = e\n    }\n    return max\n}\n\n/**\n * Returns the first element having the
largest value according to the provided [comparator] or `null` if there are no elements.\n
*/\n@SinceKotlin(\"1.4\")\npublic fun BooleanArray.maxWithOrNull(comparator: Comparator<in Boolean>):
Boolean? {\n    if (isEmpty()) return null\n    var max = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n
        if (comparator.compare(max, e) < 0) max = e\n    }\n    return max\n}\n\n/**\n * Returns the first element having the
largest value according to the provided [comparator] or `null` if there are no elements.\n
*/\n@SinceKotlin(\"1.4\")\npublic fun CharArray.maxWithOrNull(comparator: Comparator<in Char>): Char? {\n
    if (isEmpty()) return null\n    var max = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if
(comparator.compare(max, e) < 0) max = e\n    }\n    return max\n}\n\n@Deprecated(\"Use minOrNull instead.\",
ReplaceWith(\"this.minOrNull()\"))\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince = \"1.5\",
hiddenSince = \"1.6\")\n@SinceKotlin(\"1.1\")\npublic fun Array<out Double>.min(): Double? {\n    return
minOrNull()\n}\n\n@Deprecated(\"Use minOrNull instead.\",
ReplaceWith(\"this.minOrNull()\"))\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince = \"1.5\",
hiddenSince = \"1.6\")\n@SinceKotlin(\"1.1\")\npublic fun Array<out Float>.min(): Float? {\n    return
minOrNull()\n}\n\n@Deprecated(\"Use minOrNull instead.\",
ReplaceWith(\"this.minOrNull()\"))\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince = \"1.5\",
hiddenSince = \"1.6\")\npublic fun <T : Comparable<T>> Array<out T>.min(): T? {\n    return
minOrNull()\n}\n\n@Deprecated(\"Use minOrNull instead.\",
ReplaceWith(\"this.minOrNull()\"))\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince = \"1.5\",
hiddenSince = \"1.6\")\npublic fun ByteArray.min(): Byte? {\n    return minOrNull()\n}\n\n@Deprecated(\"Use

```

```

minOrNull instead.\", ReplaceWith(\"this.minOrNull()\")\n\n@DeprecatedSinceKotlin(warningSince = \"1.4\",
errorSince = \"1.5\", hiddenSince = \"1.6\")\npublic fun ShortArray.min(): Short? {\n    return
minOrNull()\n}\n\n@Deprecated(\"Use minOrNull instead.\",
ReplaceWith(\"this.minOrNull()\")\n\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince = \"1.5\",
hiddenSince = \"1.6\")\npublic fun IntArray.min(): Int? {\n    return minOrNull()\n}\n\n@Deprecated(\"Use
minOrNull instead.\", ReplaceWith(\"this.minOrNull()\")\n\n@DeprecatedSinceKotlin(warningSince = \"1.4\",
errorSince = \"1.5\", hiddenSince = \"1.6\")\npublic fun LongArray.min(): Long? {\n    return
minOrNull()\n}\n\n@Deprecated(\"Use minOrNull instead.\",
ReplaceWith(\"this.minOrNull()\")\n\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince = \"1.5\",
hiddenSince = \"1.6\")\npublic fun FloatArray.min(): Float? {\n    return minOrNull()\n}\n\n@Deprecated(\"Use
minOrNull instead.\", ReplaceWith(\"this.minOrNull()\")\n\n@DeprecatedSinceKotlin(warningSince = \"1.4\",
errorSince = \"1.5\", hiddenSince = \"1.6\")\npublic fun DoubleArray.min(): Double? {\n    return
minOrNull()\n}\n\n@Deprecated(\"Use minOrNull instead.\",
ReplaceWith(\"this.minOrNull()\")\n\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince = \"1.5\",
hiddenSince = \"1.6\")\npublic fun CharArray.min(): Char? {\n    return minOrNull()\n}\n\n@Deprecated(\"Use
minByOrNull instead.\", ReplaceWith(\"this.minByOrNull(selector)\")\n\n@DeprecatedSinceKotlin(warningSince =
\"1.4\", errorSince = \"1.5\", hiddenSince = \"1.6\")\npublic inline fun <T, R : Comparable<R>> Array<out
T>.minBy(selector: (T) -> R): T? {\n    return minByOrNull(selector)\n}\n\n@Deprecated(\"Use minByOrNull
instead.\", ReplaceWith(\"this.minByOrNull(selector)\")\n\n@DeprecatedSinceKotlin(warningSince = \"1.4\",
errorSince = \"1.5\", hiddenSince = \"1.6\")\npublic inline fun <R : Comparable<R>> ByteArray.minBy(selector:
(Byte) -> R): Byte? {\n    return minByOrNull(selector)\n}\n\n@Deprecated(\"Use minByOrNull instead.\",
ReplaceWith(\"this.minByOrNull(selector)\")\n\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince =
\"1.5\", hiddenSince = \"1.6\")\npublic inline fun <R : Comparable<R>> ShortArray.minBy(selector: (Short) -> R):
Short? {\n    return minByOrNull(selector)\n}\n\n@Deprecated(\"Use minByOrNull instead.\",
ReplaceWith(\"this.minByOrNull(selector)\")\n\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince =
\"1.5\", hiddenSince = \"1.6\")\npublic inline fun <R : Comparable<R>> IntArray.minBy(selector: (Int) -> R): Int?
{\n    return minByOrNull(selector)\n}\n\n@Deprecated(\"Use minByOrNull instead.\",
ReplaceWith(\"this.minByOrNull(selector)\")\n\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince =
\"1.5\", hiddenSince = \"1.6\")\npublic inline fun <R : Comparable<R>> LongArray.minBy(selector: (Long) -> R):
Long? {\n    return minByOrNull(selector)\n}\n\n@Deprecated(\"Use minByOrNull instead.\",
ReplaceWith(\"this.minByOrNull(selector)\")\n\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince =
\"1.5\", hiddenSince = \"1.6\")\npublic inline fun <R : Comparable<R>> FloatArray.minBy(selector: (Float) -> R):
Float? {\n    return minByOrNull(selector)\n}\n\n@Deprecated(\"Use minByOrNull instead.\",
ReplaceWith(\"this.minByOrNull(selector)\")\n\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince =
\"1.5\", hiddenSince = \"1.6\")\npublic inline fun <R : Comparable<R>> DoubleArray.minBy(selector: (Double) ->
R): Double? {\n    return minByOrNull(selector)\n}\n\n@Deprecated(\"Use minByOrNull instead.\",
ReplaceWith(\"this.minByOrNull(selector)\")\n\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince =
\"1.5\", hiddenSince = \"1.6\")\npublic inline fun <R : Comparable<R>> BooleanArray.minBy(selector: (Boolean) -
> R): Boolean? {\n    return minByOrNull(selector)\n}\n\n@Deprecated(\"Use minByOrNull instead.\",
ReplaceWith(\"this.minByOrNull(selector)\")\n\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince =
\"1.5\", hiddenSince = \"1.6\")\npublic inline fun <R : Comparable<R>> CharArray.minBy(selector: (Char) -> R):
Char? {\n    return minByOrNull(selector)\n}\n\n/**\n * Returns the first element yielding the smallest value of the
given function or `null` if there are no elements.\n * \n * @sample
samples.collections.Collections.Aggregates.minByOrNull\n * \n * @since Kotlin(\"1.4\")\n */\npublic inline fun <T, R :
Comparable<R>> Array<out T>.minByOrNull(selector: (T) -> R): T? {\n    if (isEmpty()) return null\n    var
minElem = this[0]\n    val lastIndex = this.lastIndex\n    if (lastIndex == 0) return minElem\n    var minValue =
selector(minElem)\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        val v = selector(e)\n        if (minValue > v)
{\n            minElem = e\n            minValue = v\n        }\n    }\n    return minElem\n}\n\n/**\n * Returns the first

```

```

element yielding the smallest value of the given function or `null` if there are no elements.
 * @sample
samples.collections.Collections.Aggregates.minByOrNull
 *^n@SinceKotlin("1.4")
public inline fun <R :
Comparable<R>> ByteArray.minByOrNull(selector: (Byte) -> R): Byte? {
    if (isEmpty()) return null
    var minElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return minElem
    var minValue =
selector(minElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (minValue > v)
        {
            minElem = e
            minValue = v
        }
    }
    return minElem
}
 * Returns the first
element yielding the smallest value of the given function or `null` if there are no elements.
 * @sample
samples.collections.Collections.Aggregates.minByOrNull
 *^n@SinceKotlin("1.4")
public inline fun <R :
Comparable<R>> ShortArray.minByOrNull(selector: (Short) -> R): Short? {
    if (isEmpty()) return null
    var minElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return minElem
    var minValue =
selector(minElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (minValue > v)
        {
            minElem = e
            minValue = v
        }
    }
    return minElem
}
 * Returns the first
element yielding the smallest value of the given function or `null` if there are no elements.
 * @sample
samples.collections.Collections.Aggregates.minByOrNull
 *^n@SinceKotlin("1.4")
public inline fun <R :
Comparable<R>> IntArray.minByOrNull(selector: (Int) -> R): Int? {
    if (isEmpty()) return null
    var minElem =
this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return minElem
    var minValue =
selector(minElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (minValue > v)
        {
            minElem = e
            minValue = v
        }
    }
    return minElem
}
 * Returns the first
element yielding the smallest value of the given function or `null` if there are no elements.
 * @sample
samples.collections.Collections.Aggregates.minByOrNull
 *^n@SinceKotlin("1.4")
public inline fun <R :
Comparable<R>> LongArray.minByOrNull(selector: (Long) -> R): Long? {
    if (isEmpty()) return null
    var minElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return minElem
    var minValue =
selector(minElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (minValue > v)
        {
            minElem = e
            minValue = v
        }
    }
    return minElem
}
 * Returns the first
element yielding the smallest value of the given function or `null` if there are no elements.
 * @sample
samples.collections.Collections.Aggregates.minByOrNull
 *^n@SinceKotlin("1.4")
public inline fun <R :
Comparable<R>> FloatArray.minByOrNull(selector: (Float) -> R): Float? {
    if (isEmpty()) return null
    var minElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return minElem
    var minValue =
selector(minElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (minValue > v)
        {
            minElem = e
            minValue = v
        }
    }
    return minElem
}
 * Returns the first
element yielding the smallest value of the given function or `null` if there are no elements.
 * @sample
samples.collections.Collections.Aggregates.minByOrNull
 *^n@SinceKotlin("1.4")
public inline fun <R :
Comparable<R>> DoubleArray.minByOrNull(selector: (Double) -> R): Double? {
    if (isEmpty()) return null
    var minElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return minElem
    var minValue =
selector(minElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (minValue > v)
        {
            minElem = e
            minValue = v
        }
    }
    return minElem
}
 * Returns the first
element yielding the smallest value of the given function or `null` if there are no elements.
 * @sample
samples.collections.Collections.Aggregates.minByOrNull
 *^n@SinceKotlin("1.4")
public inline fun <R :
Comparable<R>> BooleanArray.minByOrNull(selector: (Boolean) -> R): Boolean? {
    if (isEmpty()) return
null
    var minElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return minElem
    var
minValue = selector(minElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if
(minValue > v) {
            minElem = e
            minValue = v
        }
    }
    return minElem
}
 * Returns the first
element yielding the smallest value of the given function or `null` if there are no elements.
 * @sample
samples.collections.Collections.Aggregates.minByOrNull
 *^n@SinceKotlin("1.4")
public inline fun
<R : Comparable<R>> CharArray.minByOrNull(selector: (Char) -> R): Char? {
    if (isEmpty()) return null
    var minElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return minElem
    var minValue =
selector(minElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (minValue > v)
        {
            minElem = e
            minValue = v
        }
    }
    return minElem
}
 * Returns the smallest

```

value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws

NoSuchElementException if the array is empty.\n

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
```

```
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.minOf(selector: (T) -> Double): Double {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return
```

```
minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
```

```
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.minOf(selector: (Byte) -> Double): Double {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return
```

```
minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
```

```
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.minOf(selector: (Short) -> Double): Double {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return
```

```
minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
```

```
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.minOf(selector: (Int) -> Double): Double {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return
```

```
minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
```

```
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.minOf(selector: (Long) -> Double): Double {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return
```

```
minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
```

```
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.minOf(selector: (Float) -> Double): Double {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return
```

```
minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
```

```
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.minOf(selector: (Double) -> Double): Double {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for
```

```

(i in 1..lastIndex) {\n    val v = selector(this[i])\n    minValue = minOf(minValue, v)\n } \n return
minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.minOf(selector: (Boolean) ->
Double): Double {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for
(i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return
minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.minOf(selector: (Char) ->
Double): Double {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for
(i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return
minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.minOf(selector: (T) ->
Float): Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in
1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return
minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.minOf(selector: (Byte) -> Float):
Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in
1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return
minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.minOf(selector: (Short) ->
Float): Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in
1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return
minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.minOf(selector: (Int) -> Float):
Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in
1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return
minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to
each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.minOf(selector: (Long) ->

```

```

Float): Float {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array.
 * If any of values produced by [selector] function is `NaN`, the returned result is
 * `NaN`.
 * @throws NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun FloatArray.minOf(selector: (Float) -> Float):
Float {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array.
 * If any of values produced by [selector] function is `NaN`, the returned result is
 * `NaN`.
 * @throws NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun DoubleArray.minOf(selector: (Double) ->
Float): Float {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array.
 * If any of values produced by [selector] function is `NaN`, the returned result is
 * `NaN`.
 * @throws NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun BooleanArray.minOf(selector: (Boolean) ->
Float): Float {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array.
 * If any of values produced by [selector] function is `NaN`, the returned result is
 * `NaN`.
 * @throws NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun CharArray.minOf(selector: (Char) -> Float):
Float {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array.
 * @throws NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <T, R : Comparable<R>> Array<out T>.minOf(selector: (T) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (minValue > v) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array.
 * @throws NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>>
ByteArray.minOf(selector: (Byte) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (minValue > v) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array.
 * @throws NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>>

```

```

ShortArray.minOf(selector: (Short) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (minValue > v) {
            minValue = v
        }
    }
    return minValue
}
Returns the smallest value among all values produced by [selector] function applied to each element in the array.
@throws NoSuchElementException if the array is empty.

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>>
IntArray.minOf(selector: (Int) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (minValue > v) {
            minValue = v
        }
    }
    return minValue
}
Returns the smallest value among all values produced by [selector] function applied to each element in the array.
@throws NoSuchElementException if the array is empty.

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>>
LongArray.minOf(selector: (Long) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (minValue > v) {
            minValue = v
        }
    }
    return minValue
}
Returns the smallest value among all values produced by [selector] function applied to each element in the array.
@throws NoSuchElementException if the array is empty.

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>>
FloatArray.minOf(selector: (Float) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (minValue > v) {
            minValue = v
        }
    }
    return minValue
}
Returns the smallest value among all values produced by [selector] function applied to each element in the array.
@throws NoSuchElementException if the array is empty.

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>>
DoubleArray.minOf(selector: (Double) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (minValue > v) {
            minValue = v
        }
    }
    return minValue
}
Returns the smallest value among all values produced by [selector] function applied to each element in the array.
@throws NoSuchElementException if the array is empty.

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>>
BooleanArray.minOf(selector: (Boolean) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (minValue > v) {
            minValue = v
        }
    }
    return minValue
}
Returns the smallest value among all values produced by [selector] function applied to each element in the array.
@throws NoSuchElementException if the array is empty.

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>>
CharArray.minOf(selector: (Char) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (minValue > v) {
            minValue = v
        }
    }
    return minValue
}
Returns the smallest value among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.
If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution

```

```

ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.minOfOrNull(selector:
(T) -> Double): Double? {\n  if (isEmpty()) return null\n  var minValue = selector(this[0])\n  for (i in
1..lastIndex) {\n    val v = selector(this[i])\n    minValue = minOf(minValue, v)\n  }\n  return
minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to
each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function
is `NaN`, the returned result is `NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.minOfOrNull(selector: (Byte) ->
Double): Double? {\n  if (isEmpty()) return null\n  var minValue = selector(this[0])\n  for (i in 1..lastIndex) {\n
val v = selector(this[i])\n    minValue = minOf(minValue, v)\n  }\n  return minValue\n}\n\n/**\n * Returns
the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null`
if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.minOfOrNull(selector: (Short) -
> Double): Double? {\n  if (isEmpty()) return null\n  var minValue = selector(this[0])\n  for (i in 1..lastIndex) {\n
val v = selector(this[i])\n    minValue = minOf(minValue, v)\n  }\n  return minValue\n}\n\n/**\n * Returns
the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null`
if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.minOfOrNull(selector: (Int) ->
Double): Double? {\n  if (isEmpty()) return null\n  var minValue = selector(this[0])\n  for (i in 1..lastIndex) {\n
val v = selector(this[i])\n    minValue = minOf(minValue, v)\n  }\n  return minValue\n}\n\n/**\n * Returns
the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null`
if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.minOfOrNull(selector: (Long) -
> Double): Double? {\n  if (isEmpty()) return null\n  var minValue = selector(this[0])\n  for (i in 1..lastIndex) {\n
val v = selector(this[i])\n    minValue = minOf(minValue, v)\n  }\n  return minValue\n}\n\n/**\n * Returns
the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null`
if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.minOfOrNull(selector: (Float) ->
Double): Double? {\n  if (isEmpty()) return null\n  var minValue = selector(this[0])\n  for (i in 1..lastIndex) {\n
val v = selector(this[i])\n    minValue = minOf(minValue, v)\n  }\n  return minValue\n}\n\n/**\n * Returns
the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null`
if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.minOfOrNull(selector:
(Double) -> Double): Double? {\n  if (isEmpty()) return null\n  var minValue = selector(this[0])\n  for (i in
1..lastIndex) {\n    val v = selector(this[i])\n    minValue = minOf(minValue, v)\n  }\n  return
minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to
each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function
is `NaN`, the returned result is `NaN`.\n

```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.minOfOrNull(selector:\n(Boolean) -> Double): Double? {\n    if (isEmpty()) return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return\n    minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function\n is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.minOfOrNull(selector: (Char) ->\nDouble): Double? {\n    if (isEmpty()) return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns\n the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null`\n if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is\n `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.minOfOrNull(selector:\n(T) -> Float): Float? {\n    if (isEmpty()) return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex)\n    {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array\n or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned\n result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.minOfOrNull(selector: (Byte) ->\nFloat): Float? {\n    if (isEmpty()) return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns the\n smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if\n there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is\n `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.minOfOrNull(selector: (Short) -\n> Float): Float? {\n    if (isEmpty()) return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns the\n smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if\n there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is\n `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.minOfOrNull(selector: (Int) ->\nFloat): Float? {\n    if (isEmpty()) return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns the\n smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if\n there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is\n `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.minOfOrNull(selector: (Long) -\n> Float): Float? {\n    if (isEmpty()) return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns the\n smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if\n there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
```

`NaN`.\n

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.minOfOrNull(selector: (Float) ->\nFloat): Float? {\n    if (isEmpty()) return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns the\n smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if\n there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is\n `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.minOfOrNull(selector:\n(Double) -> Float): Float? {\n    if (isEmpty()) return null\n    var minValue = selector(this[0])\n    for (i in\n 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return\n  minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to\n each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function\n is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.minOfOrNull(selector:\n(Boolean) -> Float): Float? {\n    if (isEmpty()) return null\n    var minValue = selector(this[0])\n    for (i in\n 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return\n  minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to\n each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function\n is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.minOfOrNull(selector: (Char) ->\nFloat): Float? {\n    if (isEmpty()) return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns the\n smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if\n there are no elements.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Comparable<R>> Array<out\nT>.minOfOrNull(selector: (T) -> R): R? {\n    if (isEmpty()) return null\n    var minValue = selector(this[0])\n    for\n (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (minValue > v) {\n            minValue = v\n        }\n    }\n    return\n  minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * \n * applied to each element in the array or `null` if there are no elements.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>\nByteArray.minOfOrNull(selector: (Byte) -> R): R? {\n    if (isEmpty()) return null\n    var minValue =\n  selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (minValue > v) {\n            minValue = v\n        }\n    }\n    return\n  minValue\n}\n\n/**\n * Returns the smallest value among all values produced\n by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>\nShortArray.minOfOrNull(selector: (Short) -> R): R? {\n    if (isEmpty()) return null\n    var minValue =\n  selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (minValue > v) {\n            minValue = v\n        }\n    }\n    return\n  minValue\n}\n\n/**\n * Returns the smallest value among all values produced\n by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
```

```

IntArray.minOrNull(selector: (Int) -> R): R? {\n  if (isEmpty()) return null\n  var minValue =
selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if (minValue > v) {\n
minValue = v\n    }\n  }\n  return minValue\n}\n\n/**\n * Returns the smallest value among all values produced
by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
*\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
LongArray.minOrNull(selector: (Long) -> R): R? {\n  if (isEmpty()) return null\n  var minValue =
selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if (minValue > v) {\n
minValue = v\n    }\n  }\n  return minValue\n}\n\n/**\n * Returns the smallest value among all values produced
by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
*\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
FloatArray.minOrNull(selector: (Float) -> R): R? {\n  if (isEmpty()) return null\n  var minValue =
selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if (minValue > v) {\n
minValue = v\n    }\n  }\n  return minValue\n}\n\n/**\n * Returns the smallest value among all values produced
by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
*\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
DoubleArray.minOrNull(selector: (Double) -> R): R? {\n  if (isEmpty()) return null\n  var minValue =
selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if (minValue > v) {\n
minValue = v\n    }\n  }\n  return minValue\n}\n\n/**\n * Returns the smallest value among all values produced
by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
*\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
BooleanArray.minOrNull(selector: (Boolean) -> R): R? {\n  if (isEmpty()) return null\n  var minValue =
selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if (minValue > v) {\n
minValue = v\n    }\n  }\n  return minValue\n}\n\n/**\n * Returns the smallest value among all values produced
by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
*\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
CharArray.minOrNull(selector: (Char) -> R): R? {\n  if (isEmpty()) return null\n  var minValue =
selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if (minValue > v) {\n
minValue = v\n    }\n  }\n  return minValue\n}\n\n/**\n * Returns the smallest value according to the provided
[comparator]\n * among all values produced by [selector] function applied to each element in the array.\n * \n *
@throws NoSuchElementException if the array is empty.\n
*\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R> Array<out
T>.minOfWith(comparator: Comparator<in R>, selector: (T) -> R): R {\n  if (isEmpty()) throw
NoSuchElementException()\n  var minValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v =
selector(this[i])\n    if (comparator.compare(minValue, v) > 0) {\n      minValue = v\n    }\n  }\n  return
minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values
produced by [selector] function applied to each element in the array.\n * \n * @throws NoSuchElementException if
the array is empty.\n
*\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> ByteArray.minOfWith(comparator:
Comparator<in R>, selector: (Byte) -> R): R {\n  if (isEmpty()) throw NoSuchElementException()\n  var
minValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if
(comparator.compare(minValue, v) > 0) {\n      minValue = v\n    }\n  }\n  return minValue\n}\n\n/**\n

```

Returns the smallest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array. @throws NoSuchElementException if the array is empty.

```

*\/n@SinceKotlin("1.4")n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)n@OverloadResolution
ByLambdaReturnTypen@kotlin.internal.InlineOnlynpublic inline fun <R> ShortArray.minOfWith(comparator:
Comparator<in R>, selector: (Short) -> R): R {n if (isEmpty()) throw NoSuchElementException()n var
minValue = selector(this[0])n for (i in 1..lastIndex) {n val v = selector(this[i])n if
(comparator.compare(minValue, v) > 0) {n minValue = v\n }n }n return minValue\n}\n\n/**n
Returns the smallest value according to the provided [comparator] among all values produced by [selector]
function applied to each element in the array. @throws NoSuchElementException if the array is empty.
*\/n@SinceKotlin("1.4")n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)n@OverloadResolution
ByLambdaReturnTypen@kotlin.internal.InlineOnlynpublic inline fun <R> IntArray.minOfWith(comparator:
Comparator<in R>, selector: (Int) -> R): R {n if (isEmpty()) throw NoSuchElementException()n var minValue
= selector(this[0])n for (i in 1..lastIndex) {n val v = selector(this[i])n if (comparator.compare(minValue,
v) > 0) {n minValue = v\n }n }n return minValue\n}\n\n/**n
Returns the smallest value according to the provided [comparator] among all values produced by [selector] function applied to each
element in the array. @throws NoSuchElementException if the array is empty.
*\/n@SinceKotlin("1.4")n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)n@OverloadResolution
ByLambdaReturnTypen@kotlin.internal.InlineOnlynpublic inline fun <R> LongArray.minOfWith(comparator:
Comparator<in R>, selector: (Long) -> R): R {n if (isEmpty()) throw NoSuchElementException()n var
minValue = selector(this[0])n for (i in 1..lastIndex) {n val v = selector(this[i])n if
(comparator.compare(minValue, v) > 0) {n minValue = v\n }n }n return minValue\n}\n\n/**n
Returns the smallest value according to the provided [comparator] among all values produced by [selector]
function applied to each element in the array. @throws NoSuchElementException if the array is empty.
*\/n@SinceKotlin("1.4")n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)n@OverloadResolution
ByLambdaReturnTypen@kotlin.internal.InlineOnlynpublic inline fun <R> FloatArray.minOfWith(comparator:
Comparator<in R>, selector: (Float) -> R): R {n if (isEmpty()) throw NoSuchElementException()n var
minValue = selector(this[0])n for (i in 1..lastIndex) {n val v = selector(this[i])n if
(comparator.compare(minValue, v) > 0) {n minValue = v\n }n }n return minValue\n}\n\n/**n
Returns the smallest value according to the provided [comparator] among all values produced by [selector]
function applied to each element in the array. @throws NoSuchElementException if the array is empty.
*\/n@SinceKotlin("1.4")n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)n@OverloadResolution
ByLambdaReturnTypen@kotlin.internal.InlineOnlynpublic inline fun <R> DoubleArray.minOfWith(comparator:
Comparator<in R>, selector: (Double) -> R): R {n if (isEmpty()) throw NoSuchElementException()n var
minValue = selector(this[0])n for (i in 1..lastIndex) {n val v = selector(this[i])n if
(comparator.compare(minValue, v) > 0) {n minValue = v\n }n }n return minValue\n}\n\n/**n
Returns the smallest value according to the provided [comparator] among all values produced by [selector]
function applied to each element in the array. @throws NoSuchElementException if the array is empty.
*\/n@SinceKotlin("1.4")n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)n@OverloadResolution
ByLambdaReturnTypen@kotlin.internal.InlineOnlynpublic inline fun <R> BooleanArray.minOfWith(comparator:
Comparator<in R>, selector: (Boolean) -> R): R {n if (isEmpty()) throw NoSuchElementException()n var
minValue = selector(this[0])n for (i in 1..lastIndex) {n val v = selector(this[i])n if
(comparator.compare(minValue, v) > 0) {n minValue = v\n }n }n return minValue\n}\n\n/**n
Returns the smallest value according to the provided [comparator] among all values produced by [selector]
function applied to each element in the array. @throws NoSuchElementException if the array is empty.
*\/n@SinceKotlin("1.4")n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)n@OverloadResolution
ByLambdaReturnTypen@kotlin.internal.InlineOnlynpublic inline fun <R> CharArray.minOfWith(comparator:
Comparator<in R>, selector: (Char) -> R): R {n if (isEmpty()) throw NoSuchElementException()n var
minValue = selector(this[0])n for (i in 1..lastIndex) {n val v = selector(this[i])n if

```

```

(comparator.compare(minValue, v) > 0) {\n      minValue = v\n    }\n  }\n  return minValue\n}\n\n/**\n *
Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R> Array<out
T>.minOfWithOrNull(comparator: Comparator<in R>, selector: (T) -> R): R? {\n  if (isEmpty()) return null\n
var minValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if
(comparator.compare(minValue, v) > 0) {\n      minValue = v\n    }\n  }\n  return minValue\n}\n\n/**\n *
Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>
ByteArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (Byte) -> R): R? {\n  if (isEmpty()) return
null\n  var minValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if
(comparator.compare(minValue, v) > 0) {\n      minValue = v\n    }\n  }\n  return minValue\n}\n\n/**\n *
Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>
ShortArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (Short) -> R): R? {\n  if (isEmpty())
return null\n  var minValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if
(comparator.compare(minValue, v) > 0) {\n      minValue = v\n    }\n  }\n  return minValue\n}\n\n/**\n *
Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>
IntArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (Int) -> R): R? {\n  if (isEmpty()) return
null\n  var minValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if
(comparator.compare(minValue, v) > 0) {\n      minValue = v\n    }\n  }\n  return minValue\n}\n\n/**\n *
Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>
LongArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (Long) -> R): R? {\n  if (isEmpty()) return
null\n  var minValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if
(comparator.compare(minValue, v) > 0) {\n      minValue = v\n    }\n  }\n  return minValue\n}\n\n/**\n *
Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>
FloatArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (Float) -> R): R? {\n  if (isEmpty()) return
null\n  var minValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if
(comparator.compare(minValue, v) > 0) {\n      minValue = v\n    }\n  }\n  return minValue\n}\n\n/**\n *
Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>
DoubleArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (Double) -> R): R? {\n  if (isEmpty())

```

```

return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (comparator.compare(minValue, v) > 0) {\n            minValue = v\n        }\n    }\n    return minValue\n}\n\nReturns the smallest value according to the provided [comparator] * among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.\n\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> BooleanArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (Boolean) -> R): R? {\n    if (isEmpty())\n        return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (comparator.compare(minValue, v) > 0) {\n            minValue = v\n        }\n    }\n    return minValue\n}\n\nReturns the smallest value according to the provided [comparator] * among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.\n\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> CharArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (Char) -> R): R? {\n    if (isEmpty())\n        return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (comparator.compare(minValue, v) > 0) {\n            minValue = v\n        }\n    }\n    return minValue\n}\n\nReturns the smallest element or `null` if there are no elements.\n * \n * If any of elements is `NaN` returns `NaN`.\n\n*\n@SinceKotlin("1.4")\npublic fun Array<out Double>.minOrNull(): Double? {\n    if (isEmpty())\n        return null\n    var min = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        min = minOf(min, e)\n    }\n    return min\n}\n\nReturns the smallest element or `null` if there are no elements.\n * \n * If any of elements is `NaN` returns `NaN`.\n\n*\n@SinceKotlin("1.4")\npublic fun Array<out Float>.minOrNull(): Float? {\n    if (isEmpty())\n        return null\n    var min = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        min = minOf(min, e)\n    }\n    return min\n}\n\nReturns the smallest element or `null` if there are no elements.\n\n*\n@SinceKotlin("1.4")\npublic fun <T : Comparable<T>> Array<out T>.minOrNull(): T? {\n    if (isEmpty())\n        return null\n    var min = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (min > e) min = e\n    }\n    return min\n}\n\nReturns the smallest element or `null` if there are no elements.\n\n*\n@SinceKotlin("1.4")\npublic fun ByteArray.minOrNull(): Byte? {\n    if (isEmpty())\n        return null\n    var min = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (min > e) min = e\n    }\n    return min\n}\n\nReturns the smallest element or `null` if there are no elements.\n\n*\n@SinceKotlin("1.4")\npublic fun ShortArray.minOrNull(): Short? {\n    if (isEmpty())\n        return null\n    var min = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (min > e) min = e\n    }\n    return min\n}\n\nReturns the smallest element or `null` if there are no elements.\n\n*\n@SinceKotlin("1.4")\npublic fun IntArray.minOrNull(): Int? {\n    if (isEmpty())\n        return null\n    var min = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (min > e) min = e\n    }\n    return min\n}\n\nReturns the smallest element or `null` if there are no elements.\n\n*\n@SinceKotlin("1.4")\npublic fun LongArray.minOrNull(): Long? {\n    if (isEmpty())\n        return null\n    var min = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (min > e) min = e\n    }\n    return min\n}\n\nReturns the smallest element or `null` if there are no elements.\n * \n * If any of elements is `NaN` returns `NaN`.\n\n*\n@SinceKotlin("1.4")\npublic fun FloatArray.minOrNull(): Float? {\n    if (isEmpty())\n        return null\n    var min = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        min = minOf(min, e)\n    }\n    return min\n}\n\nReturns the smallest element or `null` if there are no elements.\n * \n * If any of elements is `NaN` returns `NaN`.\n\n*\n@SinceKotlin("1.4")\npublic fun DoubleArray.minOrNull(): Double? {\n    if (isEmpty())\n        return null\n    var min = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        min = minOf(min, e)\n    }\n    return min\n}\n\nReturns the smallest element or `null` if there are no elements.\n\n*\n@SinceKotlin("1.4")\npublic fun CharArray.minOrNull(): Char? {\n    if (isEmpty())\n        return null\n    var min = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (min > e) min = e\n    }\n    return min\n}\n\n@Deprecated("Use minWithOrNull instead.")\nReplaceWith("this.minWithOrNull(comparator)")\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5", hiddenSince = "1.6")\npublic fun <T> Array<out T>.minWith(comparator: Comparator<in T>): T? {\n

```



```

samples.collections.Collections.Aggregates.noneWithPredicate\n */\npublic inline fun DoubleArray.none(predicate:
(Double) -> Boolean): Boolean {\n  for (element in this) if (predicate(element)) return false\n  return
true\n}\n\n/**\n * Returns `true` if no elements match the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.noneWithPredicate\n */\npublic inline fun
BooleanArray.none(predicate: (Boolean) -> Boolean): Boolean {\n  for (element in this) if (predicate(element))
return false\n  return true\n}\n\n/**\n * Returns `true` if no elements match the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.noneWithPredicate\n */\npublic inline fun CharArray.none(predicate:
(Char) -> Boolean): Boolean {\n  for (element in this) if (predicate(element)) return false\n  return
true\n}\n\n/**\n * Performs the given [action] on each element and returns the array itself afterwards.\n
*/\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.onEach(action: (T) ->
Unit): Array<out T> {\n  return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given
[action] on each element and returns the array itself afterwards.\n
*/\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.onEach(action: (Byte) ->
Unit): ByteArray {\n  return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given
[action] on each element and returns the array itself afterwards.\n
*/\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.onEach(action: (Short) ->
Unit): ShortArray {\n  return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given
[action] on each element and returns the array itself afterwards.\n
*/\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.onEach(action: (Int) -> Unit):
IntArray {\n  return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given [action] on
each element and returns the array itself afterwards.\n
*/\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.onEach(action: (Long) ->
Unit): LongArray {\n  return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given
[action] on each element and returns the array itself afterwards.\n
*/\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.onEach(action: (Float) ->
Unit): FloatArray {\n  return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given
[action] on each element and returns the array itself afterwards.\n
*/\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.onEach(action: (Double) ->
Unit): DoubleArray {\n  return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given
[action] on each element and returns the array itself afterwards.\n
*/\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.onEach(action: (Boolean)
-> Unit): BooleanArray {\n  return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given
[action] on each element and returns the array itself afterwards.\n
*/\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.onEach(action: (Char) ->
Unit): CharArray {\n  return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given
[action] on each element, providing sequential index with the element,\n * and returns the array itself afterwards.\n
*/\n@param [action] function that takes the index of an element and the element itself\n * and performs the action on
the element.\n */\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out
T>.onEachIndexed(action: (index: Int, T) -> Unit): Array<out
T> {\n  return apply { forEachIndexed(action)
}\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element,\n * and
returns the array itself afterwards.\n * @param [action] function that takes the index of an element and the element
itself\n * and performs the action on the element.\n */\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic
inline fun ByteArray.onEachIndexed(action: (index: Int, Byte) -> Unit): ByteArray {\n  return apply {
forEachIndexed(action) }\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index
with the element,\n * and returns the array itself afterwards.\n * @param [action] function that takes the index of an
element and the element itself\n * and performs the action on the element.\n
*/\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.onEachIndexed(action:
(index: Int, Short) -> Unit): ShortArray {\n  return apply { forEachIndexed(action) }\n}\n\n/**\n * Performs the

```

given [action] on each element, providing sequential index with the element, \n * and returns the array itself afterwards.\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the element.\n */\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.onEachIndexed(action: (index: Int, Int) -> Unit): IntArray {\n return apply { forEachIndexed(action) }\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element,\n * and returns the array itself afterwards.\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the element.\n */\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.onEachIndexed(action: (index: Int, Long) -> Unit): LongArray {\n return apply { forEachIndexed(action) }\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element,\n * and returns the array itself afterwards.\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the element.\n */\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.onEachIndexed(action: (index: Int, Float) -> Unit): FloatArray {\n return apply { forEachIndexed(action) }\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element,\n * and returns the array itself afterwards.\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the element.\n */\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.onEachIndexed(action: (index: Int, Double) -> Unit): DoubleArray {\n return apply { forEachIndexed(action) }\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element,\n * and returns the array itself afterwards.\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the element.\n */\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.onEachIndexed(action: (index: Int, Boolean) -> Unit): BooleanArray {\n return apply { forEachIndexed(action) }\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element,\n * and returns the array itself afterwards.\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the element.\n */\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.onEachIndexed(action: (index: Int, Char) -> Unit): CharArray {\n return apply { forEachIndexed(action) }\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun <S, T : S> Array<out T>.reduce(operation: (acc: S, T) -> S): S {\n if (isEmpty())\n throw UnsupportedOperationException("Empty array can't be reduced.")\n var accumulator: S = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun ByteArray.reduce(operation: (acc: Byte, Byte) -> Byte): Byte {\n if (isEmpty())\n throw UnsupportedOperationException("Empty array can't be reduced.")\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun ShortArray.reduce(operation: (acc: Short, Short) -> Short): Short {\n if (isEmpty())\n throw

```

UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = this[0]\n    for (index in 1..lastIndex) {\n        accumulator = operation(accumulator, this[index])\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun IntArray.reduce(operation: (acc: Int, Int) -> Int): Int {\n    if (isEmpty())\n        throw
UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = this[0]\n    for (index in 1..lastIndex) {\n        accumulator = operation(accumulator, this[index])\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun LongArray.reduce(operation: (acc: Long, Long) -> Long): Long {\n    if (isEmpty())\n        throw
UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = this[0]\n    for (index in 1..lastIndex) {\n        accumulator = operation(accumulator, this[index])\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun FloatArray.reduce(operation: (acc: Float, Float) -> Float): Float {\n    if (isEmpty())\n        throw
UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = this[0]\n    for (index in 1..lastIndex) {\n        accumulator = operation(accumulator, this[index])\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun DoubleArray.reduce(operation: (acc: Double, Double) -> Double): Double {\n    if (isEmpty())\n        throw
UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = this[0]\n    for (index in 1..lastIndex) {\n        accumulator = operation(accumulator, this[index])\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun BooleanArray.reduce(operation: (acc: Boolean, Boolean) -> Boolean): Boolean {\n    if (isEmpty())\n        throw
UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = this[0]\n    for (index in 1..lastIndex) {\n        accumulator = operation(accumulator, this[index])\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun CharArray.reduce(operation: (acc: Char, Char) -> Char): Char {\n    if (isEmpty())\n        throw

```

```

UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = this[0]\n    for (index in 1..lastIndex) {\n        accumulator = operation(accumulator, this[index])\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n *\npublic inline fun <S, T : S> Array<out T>.reduceIndexed(operation: (index: Int, acc: S, T) -> S): S {\n    if (isEmpty())\n        throw UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator: S = this[0]\n    for (index in 1..lastIndex) {\n        accumulator = operation(index, accumulator, this[index])\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n *\npublic inline fun ByteArray.reduceIndexed(operation: (index: Int, acc: Byte, Byte) -> Byte): Byte {\n    if (isEmpty())\n        throw UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = this[0]\n    for (index in 1..lastIndex) {\n        accumulator = operation(index, accumulator, this[index])\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n *\npublic inline fun ShortArray.reduceIndexed(operation: (index: Int, acc: Short, Short) -> Short): Short {\n    if (isEmpty())\n        throw UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = this[0]\n    for (index in 1..lastIndex) {\n        accumulator = operation(index, accumulator, this[index])\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n *\npublic inline fun IntArray.reduceIndexed(operation: (index: Int, acc: Int, Int) -> Int): Int {\n    if (isEmpty())\n        throw UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = this[0]\n    for (index in 1..lastIndex) {\n        accumulator = operation(index, accumulator, this[index])\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n *\npublic inline fun LongArray.reduceIndexed(operation: (index: Int, acc: Long, Long) -> Long): Long {\n    if (isEmpty())\n        throw UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = this[0]\n    for (index in 1..lastIndex) {\n        accumulator = operation(index, accumulator, this[index])\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to

```

right\n * to current accumulator value and each element with its index in the original array.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun
FloatArray.reduceIndexed(operation: (index: Int, acc: Float, Float) -> Float): Float {\n if (isEmpty())\n throw UnsupportedOperationException("Empty array can't be reduced.")\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun
DoubleArray.reduceIndexed(operation: (index: Int, acc: Double, Double) -> Double): Double {\n if (isEmpty())\n throw UnsupportedOperationException("Empty array can't be reduced.")\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun
BooleanArray.reduceIndexed(operation: (index: Int, acc: Boolean, Boolean) -> Boolean): Boolean {\n if (isEmpty())\n throw UnsupportedOperationException("Empty array can't be reduced.")\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun
CharArray.reduceIndexed(operation: (index: Int, acc: Char, Char) -> Char): Char {\n if (isEmpty())\n throw UnsupportedOperationException("Empty array can't be reduced.")\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n */\n@SinceKotlin("1.4")\npublic inline fun <S, T : S> Array<out T>.reduceIndexedOrNull(operation: (index: Int, acc: S, T) -> S): S? {\n if (isEmpty())\n return null\n var accumulator: S = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n */\n@SinceKotlin("1.4")\npublic inline fun ByteArray.reduceIndexedOrNull(operation: (index: Int, acc: Byte,

```

Byte) -> Byte): Byte? {\n  if (isEmpty())\n    return null\n  var accumulator = this[0]\n  for (index in 1..lastIndex) {\n    accumulator = operation(index, accumulator, this[index])\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n\n*\n@SinceKotlin("1.4")\npublic inline fun ShortArray.reduceIndexedOrNull(operation: (index: Int, acc: Short, Short) -> Short): Short? {\n  if (isEmpty())\n    return null\n  var accumulator = this[0]\n  for (index in 1..lastIndex) {\n    accumulator = operation(index, accumulator, this[index])\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n\n*\n@SinceKotlin("1.4")\npublic inline fun IntArray.reduceIndexedOrNull(operation: (index: Int, acc: Int, Int) -> Int): Int? {\n  if (isEmpty())\n    return null\n  var accumulator = this[0]\n  for (index in 1..lastIndex) {\n    accumulator = operation(index, accumulator, this[index])\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n\n*\n@SinceKotlin("1.4")\npublic inline fun LongArray.reduceIndexedOrNull(operation: (index: Int, acc: Long, Long) -> Long): Long? {\n  if (isEmpty())\n    return null\n  var accumulator = this[0]\n  for (index in 1..lastIndex) {\n    accumulator = operation(index, accumulator, this[index])\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n\n*\n@SinceKotlin("1.4")\npublic inline fun FloatArray.reduceIndexedOrNull(operation: (index: Int, acc: Float, Float) -> Float): Float? {\n  if (isEmpty())\n    return null\n  var accumulator = this[0]\n  for (index in 1..lastIndex) {\n    accumulator = operation(index, accumulator, this[index])\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n\n*\n@SinceKotlin("1.4")\npublic inline fun DoubleArray.reduceIndexedOrNull(operation: (index: Int, acc: Double, Double) -> Double): Double? {\n  if (isEmpty())\n    return null\n  var accumulator = this[0]\n  for (index in 1..lastIndex) {\n    accumulator = operation(index, accumulator, this[index])\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n\n*\n@SinceKotlin("1.4")\npublic inline fun BooleanArray.reduceIndexedOrNull(operation: (index: Int, acc: Boolean, Boolean) -> Boolean): Boolean? {\n  if (isEmpty())\n    return null\n  var accumulator = this[0]\n  for (index in 1..lastIndex) {\n    accumulator = operation(index, accumulator, this[index])\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element

```

with its index in the original array.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n * \n * @SinceKotlin("1.4")\n * \n * @WasExperimental(ExperimentalStdlibApi::class)\n * \n * public inline fun CharArray.reduceIndexedOrNull(operation: (index: Int, acc: Char, Char) -> Char): Char? {\n * \n * if (isEmpty())\n * \n * return null\n * \n * var accumulator = this[0]\n * \n * for (index in 1..lastIndex) {\n * \n * accumulator = operation(index, accumulator, this[index])\n * \n * }\n * \n * return accumulator\n * \n * }\n * \n * \n * \n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n * \n * @SinceKotlin("1.4")\n * \n * @WasExperimental(ExperimentalStdlibApi::class)\n * \n * public inline fun <S, T : S> Array<out T>.reduceOrNull(operation: (acc: S, T) -> S): S? {\n * \n * if (isEmpty())\n * \n * return null\n * \n * var accumulator: S = this[0]\n * \n * for (index in 1..lastIndex) {\n * \n * accumulator = operation(accumulator, this[index])\n * \n * }\n * \n * return accumulator\n * \n * }\n * \n * \n * \n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n * \n * @SinceKotlin("1.4")\n * \n * @WasExperimental(ExperimentalStdlibApi::class)\n * \n * public inline fun ByteArray.reduceOrNull(operation: (acc: Byte, Byte) -> Byte): Byte? {\n * \n * if (isEmpty())\n * \n * return null\n * \n * var accumulator = this[0]\n * \n * for (index in 1..lastIndex) {\n * \n * accumulator = operation(accumulator, this[index])\n * \n * }\n * \n * return accumulator\n * \n * }\n * \n * \n * \n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n * \n * @SinceKotlin("1.4")\n * \n * @WasExperimental(ExperimentalStdlibApi::class)\n * \n * public inline fun ShortArray.reduceOrNull(operation: (acc: Short, Short) -> Short): Short? {\n * \n * if (isEmpty())\n * \n * return null\n * \n * var accumulator = this[0]\n * \n * for (index in 1..lastIndex) {\n * \n * accumulator = operation(accumulator, this[index])\n * \n * }\n * \n * return accumulator\n * \n * }\n * \n * \n * \n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n * \n * @SinceKotlin("1.4")\n * \n * @WasExperimental(ExperimentalStdlibApi::class)\n * \n * public inline fun IntArray.reduceOrNull(operation: (acc: Int, Int) -> Int): Int? {\n * \n * if (isEmpty())\n * \n * return null\n * \n * var accumulator = this[0]\n * \n * for (index in 1..lastIndex) {\n * \n * accumulator = operation(accumulator, this[index])\n * \n * }\n * \n * return accumulator\n * \n * }\n * \n * \n * \n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n * \n * @SinceKotlin("1.4")\n * \n * @WasExperimental(ExperimentalStdlibApi::class)\n * \n * public inline fun LongArray.reduceOrNull(operation: (acc: Long, Long) -> Long): Long? {\n * \n * if (isEmpty())\n * \n * return null\n * \n * var accumulator = this[0]\n * \n * for (index in 1..lastIndex) {\n * \n * accumulator = operation(accumulator, this[index])\n * \n * }\n * \n * return accumulator\n * \n * }\n * \n * \n * \n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n * \n * @SinceKotlin("1.4")\n * \n * @WasExperimental(ExperimentalStdlibApi::class)\n * \n * public inline fun FloatArray.reduceOrNull(operation: (acc: Float, Float) -> Float): Float? {\n * \n * if (isEmpty())\n * \n * return null\n * \n * var accumulator = this[0]\n * \n * for (index in 1..lastIndex) {\n * \n * accumulator = operation(accumulator, this[index])\n * \n * }\n * \n * return accumulator\n * \n * }\n * \n * \n * \n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n * \n * @SinceKotlin("1.4")\n * \n * @WasExperimental(ExperimentalStdlibApi::class)\n * \n * public inline fun


```
(index >= 0) {\n    accumulator = operation(get(index--), accumulator)\n } \n return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an\n * expected way,\n * please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n */\n\npublic inline fun IntArray.reduceRight(operation: (Int, acc: Int) -> Int): Int {\n    var index = lastIndex\n    if (index < 0) throw\n    UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = get(index--)\n    while\n    (index >= 0) {\n        accumulator = operation(get(index--), accumulator)\n    }\n    return\n    accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to\n * left\n * to each element and current accumulator value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an\n * expected way,\n * please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample\n * samples.collections.Collections.Aggregates.reduceRight\n */\n\npublic inline fun LongArray.reduceRight(operation: (Long, acc: Long) -> Long): Long {\n    var index = lastIndex\n    if (index < 0) throw\n    UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = get(index--)\n    while\n    (index >= 0) {\n        accumulator = operation(get(index--), accumulator)\n    }\n    return\n    accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to\n * left\n * to each element and current accumulator value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an\n * expected way,\n * please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample\n * samples.collections.Collections.Aggregates.reduceRight\n */\n\npublic inline fun FloatArray.reduceRight(operation:\n    (Float, acc: Float) -> Float): Float {\n    var index = lastIndex\n    if (index < 0) throw\n    UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = get(index--)\n    while\n    (index >= 0) {\n        accumulator = operation(get(index--), accumulator)\n    }\n    return\n    accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to\n * left\n * to each element and current accumulator value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an\n * expected way,\n * please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample\n * samples.collections.Collections.Aggregates.reduceRight\n */\n\npublic inline fun DoubleArray.reduceRight(operation: (Double, acc: Double) -> Double): Double {\n    var index = lastIndex\n    if (index < 0) throw\n    UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator =\n    get(index--)\n    while (index >= 0) {\n        accumulator = operation(get(index--), accumulator)\n    }\n    return\n    accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to\n * left\n * to each element and current accumulator value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an\n * expected way,\n * please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample\n * samples.collections.Collections.Aggregates.reduceRight\n */\n\npublic inline fun BooleanArray.reduceRight(operation: (Boolean, acc: Boolean) -> Boolean): Boolean {\n    var index = lastIndex\n    if (index < 0) throw\n    UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator =\n    get(index--)\n    while (index >= 0) {\n        accumulator = operation(get(index--), accumulator)\n    }\n    return\n    accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to\n * left\n * to each element and current accumulator value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an\n * expected way,\n * please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample
```

```

samples.collections.Collections.Aggregates.reduceRight\n *\npublic inline fun CharArray.reduceRight(operation:
(Char, acc: Char) -> Char): Char {\n  var index = lastIndex\n  if (index < 0) throw
UnsupportedOperationException("Empty array can't be reduced.")\n  var accumulator = get(index--)\n  while
(index >= 0) {\n    accumulator = operation(get(index--), accumulator)\n  }\n  return accumulator\n}\n\n/**\n *
Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with
its index in the original array and current accumulator value.\n * \n * Throws an exception if this array is empty. If
the array can be empty in an expected way,\n * please use [reduceRightIndexedOrNull] instead. It returns `null`
when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, the element
itself and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRight\n *\npublic inline fun <S, T : S> Array<out
T>.reduceRightIndexed(operation: (index: Int, T, acc: S) -> S): S {\n  var index = lastIndex\n  if (index < 0) throw
UnsupportedOperationException("Empty array can't be reduced.")\n  var accumulator: S = get(index--)\n  while
(index >= 0) {\n    accumulator = operation(index, get(index), accumulator)\n    --index\n  }\n  return
accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to
left\n * to each element with its index in the original array and current accumulator value.\n * \n * Throws an
exception if this array is empty. If the array can be empty in an expected way,\n * please use
[reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation]
function that takes the index of an element, the element itself and current accumulator value,\n * and calculates the
next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n *\npublic
inline fun ByteArray.reduceRightIndexed(operation: (index: Int, Byte, acc: Byte) -> Byte): Byte {\n  var index =
lastIndex\n  if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")\n  var
accumulator = get(index--)\n  while (index >= 0) {\n    accumulator = operation(index, get(index),
accumulator)\n    --index\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the last
element and applying [operation] from right to left\n * to each element with its index in the original array and
current accumulator value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an
expected way,\n * please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n
* @param [operation] function that takes the index of an element, the element itself and current accumulator
value,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRight\n *\npublic inline fun
ShortArray.reduceRightIndexed(operation: (index: Int, Short, acc: Short) -> Short): Short {\n  var index =
lastIndex\n  if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")\n  var
accumulator = get(index--)\n  while (index >= 0) {\n    accumulator = operation(index, get(index),
accumulator)\n    --index\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the last
element and applying [operation] from right to left\n * to each element with its index in the original array and
current accumulator value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an
expected way,\n * please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n
* @param [operation] function that takes the index of an element, the element itself and current accumulator
value,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRight\n *\npublic inline fun
IntArray.reduceRightIndexed(operation: (index: Int, Int, acc: Int) -> Int): Int {\n  var index = lastIndex\n  if (index
< 0) throw UnsupportedOperationException("Empty array can't be reduced.")\n  var accumulator = get(index--)\n
while (index >= 0) {\n    accumulator = operation(index, get(index), accumulator)\n    --index\n  }\n  return
accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to
left\n * to each element with its index in the original array and current accumulator value.\n * \n * Throws an
exception if this array is empty. If the array can be empty in an expected way,\n * please use
[reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation]
function that takes the index of an element, the element itself and current accumulator value,\n * and calculates the
next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n *\npublic

```

```

inline fun LongArray.reduceRightIndexed(operation: (index: Int, Long, acc: Long) -> Long): Long {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

```

Accumulates value starting with the last element and applying [operation] from right to left to each element with its index in the original array and current accumulator value.
 Throws an exception if this array is empty. If the array can be empty in an expected way, please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.
 @param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.
 @sample

```

samples.collections.Collections.Aggregates.reduceRight

```

```

public inline fun FloatArray.reduceRightIndexed(operation: (index: Int, Float, acc: Float) -> Float): Float {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

```

Accumulates value starting with the last element and applying [operation] from right to left to each element with its index in the original array and current accumulator value.
 Throws an exception if this array is empty. If the array can be empty in an expected way, please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.
 @param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.
 @sample

```

samples.collections.Collections.Aggregates.reduceRight

```

```

public inline fun DoubleArray.reduceRightIndexed(operation: (index: Int, Double, acc: Double) -> Double): Double {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

```

Accumulates value starting with the last element and applying [operation] from right to left to each element with its index in the original array and current accumulator value.
 Throws an exception if this array is empty. If the array can be empty in an expected way, please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.
 @param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.
 @sample

```

samples.collections.Collections.Aggregates.reduceRight

```

```

public inline fun BooleanArray.reduceRightIndexed(operation: (index: Int, Boolean, acc: Boolean) -> Boolean): Boolean {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

```

Accumulates value starting with the last element and applying [operation] from right to left to each element with its index in the original array and current accumulator value.
 Throws an exception if this array is empty. If the array can be empty in an expected way, please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.
 @param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.
 @sample

```

samples.collections.Collections.Aggregates.reduceRight

```

```

public inline fun CharArray.reduceRightIndexed(operation: (index: Int, Char, acc: Char) -> Char): Char {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

```

Accumulates value starting with the last element and applying [operation] from right to left to each element with its index in the original array and current accumulator value.
 Returns `null` if the array is empty.
 @param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.
 @sample

```

samples.collections.Collections.Aggregates.reduceRightOrNull

```

```

@SinceKotlin("1.4")
public inline fun <S,

```

```

T : S> Array<out T>.reduceRightIndexedOrNull(operation: (index: Int, T, acc: S) -> S): S? {\n  var index =
lastIndex\n  if (index < 0) return null\n  var accumulator: S = get(index--)\n  while (index >= 0) {\n
accumulator = operation(index, get(index), accumulator)\n    --index\n  }\n  return accumulator\n}\n\n/*\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with
its index in the original array and current accumulator value.\n * Returns `null` if the array is empty.\n * \n *
@param [operation] function that takes the index of an element, the element itself and current accumulator value,\n
* and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n */\n\n@SinceKotlin("1.4")\npublic inline fun
ByteArray.reduceRightIndexedOrNull(operation: (index: Int, Byte, acc: Byte) -> Byte): Byte? {\n  var index =
lastIndex\n  if (index < 0) return null\n  var accumulator = get(index--)\n  while (index >= 0) {\n
accumulator = operation(index, get(index), accumulator)\n    --index\n  }\n  return accumulator\n}\n\n/*\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with
its index in the original array and current accumulator value.\n * Returns `null` if the array is empty.\n * \n *
@param [operation] function that takes the index of an element, the element itself and current accumulator value,\n
* and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n */\n\n@SinceKotlin("1.4")\npublic inline fun
ShortArray.reduceRightIndexedOrNull(operation: (index: Int, Short, acc: Short) -> Short): Short? {\n  var index =
lastIndex\n  if (index < 0) return null\n  var accumulator = get(index--)\n  while (index >= 0) {\n
accumulator = operation(index, get(index), accumulator)\n    --index\n  }\n  return accumulator\n}\n\n/*\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with
its index in the original array and current accumulator value.\n * Returns `null` if the array is empty.\n * \n *
@param [operation] function that takes the index of an element, the element itself and current accumulator value,\n
* and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n */\n\n@SinceKotlin("1.4")\npublic inline fun
IntArray.reduceRightIndexedOrNull(operation: (index: Int, Int, acc: Int) -> Int): Int? {\n  var index = lastIndex\n
if (index < 0) return null\n  var accumulator = get(index--)\n  while (index >= 0) {\n    accumulator =
operation(index, get(index), accumulator)\n    --index\n  }\n  return accumulator\n}\n\n/*\n * Accumulates
value starting with the last element and applying [operation] from right to left\n * to each element with its index in
the original array and current accumulator value.\n * Returns `null` if the array is empty.\n * \n * @param
[operation] function that takes the index of an element, the element itself and current accumulator value,\n * and
calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n */\n\n@SinceKotlin("1.4")\npublic inline fun
LongArray.reduceRightIndexedOrNull(operation: (index: Int, Long, acc: Long) -> Long): Long? {\n  var index =
lastIndex\n  if (index < 0) return null\n  var accumulator = get(index--)\n  while (index >= 0) {\n
accumulator = operation(index, get(index), accumulator)\n    --index\n  }\n  return accumulator\n}\n\n/*\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with
its index in the original array and current accumulator value.\n * Returns `null` if the array is empty.\n * \n *
@param [operation] function that takes the index of an element, the element itself and current accumulator value,\n
* and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n */\n\n@SinceKotlin("1.4")\npublic inline fun
FloatArray.reduceRightIndexedOrNull(operation: (index: Int, Float, acc: Float) -> Float): Float? {\n  var index =
lastIndex\n  if (index < 0) return null\n  var accumulator = get(index--)\n  while (index >= 0) {\n
accumulator = operation(index, get(index), accumulator)\n    --index\n  }\n  return accumulator\n}\n\n/*\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with
its index in the original array and current accumulator value.\n * Returns `null` if the array is empty.\n * \n *
@param [operation] function that takes the index of an element, the element itself and current accumulator value,\n
* and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n */\n\n@SinceKotlin("1.4")\npublic inline fun

```

```

DoubleArray.reduceRightIndexedOrNull(operation: (index: Int, Double, acc: Double) -> Double): Double? {\n  var
index = lastIndex\n  if (index < 0) return null\n  var accumulator = get(index--)\n  while (index >= 0) {\n
accumulator = operation(index, get(index), accumulator)\n    --index\n  }\n  return accumulator\n}\n\n/**\n *
Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with
its index in the original array and current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n *
@param [operation] function that takes the index of an element, the element itself and current accumulator value,\n
* and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n */\n\n@SinceKotlin("1.4")\npublic inline fun
BooleanArray.reduceRightIndexedOrNull(operation: (index: Int, Boolean, acc: Boolean) -> Boolean): Boolean? {\n
var index = lastIndex\n  if (index < 0) return null\n  var accumulator = get(index--)\n  while (index >= 0) {\n
accumulator = operation(index, get(index), accumulator)\n    --index\n  }\n  return accumulator\n}\n\n/**\n *
Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with
its index in the original array and current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n *
@param [operation] function that takes the index of an element, the element itself and current accumulator value,\n
* and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n */\n\n@SinceKotlin("1.4")\npublic inline fun
CharArray.reduceRightIndexedOrNull(operation: (index: Int, Char, acc: Char) -> Char): Char? {\n  var index =
lastIndex\n  if (index < 0) return null\n  var accumulator = get(index--)\n  while (index >= 0) {\n
accumulator = operation(index, get(index), accumulator)\n    --index\n  }\n  return accumulator\n}\n\n/**\n *
Accumulates value starting with the last element and applying [operation] from right to left\n * to each element and
current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that
takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n */\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun <S, T : S>
Array<out T>.reduceRightOrNull(operation: (T, acc: S) -> S): S? {\n  var index = lastIndex\n  if (index < 0)
return null\n  var accumulator: S = get(index--)\n  while (index >= 0) {\n    accumulator = operation(get(index--
), accumulator)\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and
applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Returns `null` if
the array is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n *
and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n */\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun
ByteArray.reduceRightOrNull(operation: (Byte, acc: Byte) -> Byte): Byte? {\n  var index = lastIndex\n  if (index
< 0) return null\n  var accumulator = get(index--)\n  while (index >= 0) {\n    accumulator =
operation(get(index--), accumulator)\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the
last element and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n *
Returns `null` if the array is empty.\n * \n * @param [operation] function that takes an element and current
accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n */\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun
ShortArray.reduceRightOrNull(operation: (Short, acc: Short) -> Short): Short? {\n  var index = lastIndex\n  if
(index < 0) return null\n  var accumulator = get(index--)\n  while (index >= 0) {\n    accumulator =
operation(get(index--), accumulator)\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the
last element and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n *
Returns `null` if the array is empty.\n * \n * @param [operation] function that takes an element and current
accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n */\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun

```

```

IntArray.reduceRightOrNull(operation: (Int, acc: Int) -> Int): Int? {\n  var index = lastIndex\n  if (index < 0)
return null\n  var accumulator = get(index--)\n  while (index >= 0) {\n    accumulator = operation(get(index--),
accumulator)\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and
applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Returns `null` if
the array is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n *
and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n
*/\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun
LongArray.reduceRightOrNull(operation: (Long, acc: Long) -> Long): Long? {\n  var index = lastIndex\n  if
(index < 0) return null\n  var accumulator = get(index--)\n  while (index >= 0) {\n    accumulator =
operation(get(index--), accumulator)\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the
last element and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n *
Returns `null` if the array is empty.\n * \n * @param [operation] function that takes an element and current
accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n
*/\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun
FloatArray.reduceRightOrNull(operation: (Float, acc: Float) -> Float): Float? {\n  var index = lastIndex\n  if
(index < 0) return null\n  var accumulator = get(index--)\n  while (index >= 0) {\n    accumulator =
operation(get(index--), accumulator)\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the
last element and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n *
Returns `null` if the array is empty.\n * \n * @param [operation] function that takes an element and current
accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n
*/\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun
DoubleArray.reduceRightOrNull(operation: (Double, acc: Double) -> Double): Double? {\n  var index =
lastIndex\n  if (index < 0) return null\n  var accumulator = get(index--)\n  while (index >= 0) {\n
accumulator = operation(get(index--), accumulator)\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value
starting with the last element and applying [operation] from right to left\n * to each element and current accumulator
value.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes an element and
current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n
*/\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun
BooleanArray.reduceRightOrNull(operation: (Boolean, acc: Boolean) -> Boolean): Boolean? {\n  var index =
lastIndex\n  if (index < 0) return null\n  var accumulator = get(index--)\n  while (index >= 0) {\n
accumulator = operation(get(index--), accumulator)\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value
starting with the last element and applying [operation] from right to left\n * to each element and current accumulator
value.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes an element and
current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n
*/\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun
CharArray.reduceRightOrNull(operation: (Char, acc: Char) -> Char): Char? {\n  var index = lastIndex\n  if (index
< 0) return null\n  var accumulator = get(index--)\n  while (index >= 0) {\n    accumulator =
operation(get(index--), accumulator)\n  }\n  return accumulator\n}\n\n/**\n * Returns a list containing successive
accumulation values generated by applying [operation] from left to right\n * to each element and current
accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should
not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation]
function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n *
@sample samples.collections.Collections.Aggregates.runningFold\n
*/\n@SinceKotlin("1.4")\npublic inline fun

```

```

<T, R> Array<out T>.runningFold(initial: R, operation: (acc: R, T) -> R): List<R> {\n  if (isEmpty()) return
listOf(initial)\n  val result = ArrayList<R>(size + 1).apply { add(initial) }\n  var accumulator = initial\n  for
(element in this) {\n    accumulator = operation(accumulator, element)\n    result.add(accumulator)\n  }\n
return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying
[operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n
* Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the
previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an
element, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningFold\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <R> ByteArray.runningFold(initial: R,
operation: (acc: R, Byte) -> R): List<R> {\n  if (isEmpty()) return listOf(initial)\n  val result = ArrayList<R>(size
+ 1).apply { add(initial) }\n  var accumulator = initial\n  for (element in this) {\n    accumulator =
operation(accumulator, element)\n    result.add(accumulator)\n  }\n  return result\n}\n\n/**\n * Returns a list
containing successive accumulation values generated by applying [operation] from left to right\n * to each element
and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation]
function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param
[operation] function that takes current accumulator value and an element, and calculates the next accumulator
value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <R> ShortArray.runningFold(initial: R,
operation: (acc: R, Short) -> R): List<R> {\n  if (isEmpty()) return listOf(initial)\n  val result = ArrayList<R>(size
+ 1).apply { add(initial) }\n  var accumulator = initial\n  for (element in this) {\n    accumulator =
operation(accumulator, element)\n    result.add(accumulator)\n  }\n  return result\n}\n\n/**\n * Returns a list
containing successive accumulation values generated by applying [operation] from left to right\n * to each element
and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation]
function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param
[operation] function that takes current accumulator value and an element, and calculates the next accumulator
value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <R> IntArray.runningFold(initial: R,
operation: (acc: R, Int) -> R): List<R> {\n  if (isEmpty()) return listOf(initial)\n  val result = ArrayList<R>(size +
1).apply { add(initial) }\n  var accumulator = initial\n  for (element in this) {\n    accumulator =
operation(accumulator, element)\n    result.add(accumulator)\n  }\n  return result\n}\n\n/**\n * Returns a list
containing successive accumulation values generated by applying [operation] from left to right\n * to each element
and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation]
function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param
[operation] function that takes current accumulator value and an element, and calculates the next accumulator
value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <R> LongArray.runningFold(initial: R,
operation: (acc: R, Long) -> R): List<R> {\n  if (isEmpty()) return listOf(initial)\n  val result = ArrayList<R>(size
+ 1).apply { add(initial) }\n  var accumulator = initial\n  for (element in this) {\n    accumulator =
operation(accumulator, element)\n    result.add(accumulator)\n  }\n  return result\n}\n\n/**\n * Returns a list
containing successive accumulation values generated by applying [operation] from left to right\n * to each element
and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation]
function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param
[operation] function that takes current accumulator value and an element, and calculates the next accumulator
value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <R> FloatArray.runningFold(initial: R,
operation: (acc: R, Float) -> R): List<R> {\n  if (isEmpty()) return listOf(initial)\n  val result = ArrayList<R>(size
+ 1).apply { add(initial) }\n  var accumulator = initial\n  for (element in this) {\n    accumulator =

```

operation(accumulator, element)\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n

```
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <R> DoubleArray.runningFold(initial: R, operation: (acc: R, Double) -> R): List<R> {\n    if (isEmpty()) return listOf(initial)\n    val result = ArrayList<R>(size + 1).apply { add(initial) }\n    var accumulator = initial\n    for (element in this) {\n        accumulator = operation(accumulator, element)\n        result.add(accumulator)\n    }\n    return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n
```

```
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <R> BooleanArray.runningFold(initial: R, operation: (acc: R, Boolean) -> R): List<R> {\n    if (isEmpty()) return listOf(initial)\n    val result = ArrayList<R>(size + 1).apply { add(initial) }\n    var accumulator = initial\n    for (element in this) {\n        accumulator = operation(accumulator, element)\n        result.add(accumulator)\n    }\n    return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n
```

```
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <R> CharArray.runningFold(initial: R, operation: (acc: R, Char) -> R): List<R> {\n    if (isEmpty()) return listOf(initial)\n    val result = ArrayList<R>(size + 1).apply { add(initial) }\n    var accumulator = initial\n    for (element in this) {\n        accumulator = operation(accumulator, element)\n        result.add(accumulator)\n    }\n    return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n
```

```
*\n@SinceKotlin("1.4")\npublic inline fun <T, R> Array<out T>.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): List<R> {\n    if (isEmpty()) return listOf(initial)\n    val result = ArrayList<R>(size + 1).apply { add(initial) }\n    var accumulator = initial\n    for (index in indices) {\n        accumulator = operation(index, accumulator, this[index])\n        result.add(accumulator)\n    }\n    return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n
```

```
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <R> ByteArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, Byte) -> R): List<R> {\n    if (isEmpty()) return listOf(initial)\n    val result = ArrayList<R>(size + 1).apply { add(initial) }\n    var accumulator = initial\n    for (index in indices) {\n        accumulator = operation(index, accumulator, this[index])\n        result.add(accumulator)\n    }\n    return result\n}\n\n/**\n * Returns a list containing successive accumulation
```

values generated by applying [operation] from left to right to each element, its index in the original array and current accumulator value that starts with [initial] value. Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list. @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value. @sample samples.collections.Collections.Aggregates.runningFold

```

*\/n@SinceKotlin("1.4")n@kotlin.internal.InlineOnlynpublic inline fun <R>
ShortArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, Short) -> R): List<R> {n  if (isEmpty())
return listOf(initial)n  val result = ArrayList<R>(size + 1).apply { add(initial) }n  var accumulator = initialn
for (index in indices) {n    accumulator = operation(index, accumulator, this[index])n
result.add(accumulator)n  }n  return resultn}n/n/**n * Returns a list containing successive accumulation
values generated by applying [operation] from left to right to each element, its index in the original array and
current accumulator value that starts with [initial] value. Note that `acc` value passed to [operation] function
should not be mutated; otherwise it would affect the previous value in resulting list. n * @param [operation]
function that takes the index of an element, current accumulator value and the element itself, and calculates the
next accumulator value. n * @sample samples.collections.Collections.Aggregates.runningFoldn
*\/n@SinceKotlin("1.4")n@kotlin.internal.InlineOnlynpublic inline fun <R>
IntArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, Int) -> R): List<R> {n  if (isEmpty()) return
listOf(initial)n  val result = ArrayList<R>(size + 1).apply { add(initial) }n  var accumulator = initialn  for
(index in indices) {n    accumulator = operation(index, accumulator, this[index])n    result.add(accumulator)n
}n  return resultn}n/n/**n * Returns a list containing successive accumulation values generated by applying
[operation] from left to right to each element, its index in the original array and current accumulator value that
starts with [initial] value. Note that `acc` value passed to [operation] function should not be mutated; n *
otherwise it would affect the previous value in resulting list. n * @param [operation] function that takes the
index of an element, current accumulator value and the element itself, and calculates the next accumulator
value. n * @sample samples.collections.Collections.Aggregates.runningFoldn
*\/n@SinceKotlin("1.4")n@kotlin.internal.InlineOnlynpublic inline fun <R>
LongArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, Long) -> R): List<R> {n  if (isEmpty())
return listOf(initial)n  val result = ArrayList<R>(size + 1).apply { add(initial) }n  var accumulator = initialn
for (index in indices) {n    accumulator = operation(index, accumulator, this[index])n
result.add(accumulator)n  }n  return resultn}n/n/**n * Returns a list containing successive accumulation
values generated by applying [operation] from left to right to each element, its index in the original array and
current accumulator value that starts with [initial] value. Note that `acc` value passed to [operation] function
should not be mutated; otherwise it would affect the previous value in resulting list. n * @param [operation]
function that takes the index of an element, current accumulator value and the element itself, and calculates the
next accumulator value. n * @sample samples.collections.Collections.Aggregates.runningFoldn
*\/n@SinceKotlin("1.4")n@kotlin.internal.InlineOnlynpublic inline fun <R>
FloatArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, Float) -> R): List<R> {n  if (isEmpty())
return listOf(initial)n  val result = ArrayList<R>(size + 1).apply { add(initial) }n  var accumulator = initialn
for (index in indices) {n    accumulator = operation(index, accumulator, this[index])n
result.add(accumulator)n  }n  return resultn}n/n/**n * Returns a list containing successive accumulation
values generated by applying [operation] from left to right to each element, its index in the original array and
current accumulator value that starts with [initial] value. Note that `acc` value passed to [operation] function
should not be mutated; otherwise it would affect the previous value in resulting list. n * @param [operation]
function that takes the index of an element, current accumulator value and the element itself, and calculates the
next accumulator value. n * @sample samples.collections.Collections.Aggregates.runningFoldn
*\/n@SinceKotlin("1.4")n@kotlin.internal.InlineOnlynpublic inline fun <R>
DoubleArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, Double) -> R): List<R> {n  if
(isEmpty()) return listOf(initial)n  val result = ArrayList<R>(size + 1).apply { add(initial) }n  var accumulator =

```

```

initial\n for (index in indices) {\n accumulator = operation(index, accumulator, this[index])\n
result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation
values generated by applying [operation] from left to right\n * to each element, its index in the original array and
current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function
should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation]
function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the
next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n
*\n*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <R>
BooleanArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, Boolean) -> R): List<R> {\n if
(isEmpty()) return listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator =
initial\n for (index in indices) {\n accumulator = operation(index, accumulator, this[index])\n
result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation
values generated by applying [operation] from left to right\n * to each element, its index in the original array and
current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function
should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation]
function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the
next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n
*\n*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <R>
CharArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, Char) -> R): List<R> {\n if (isEmpty())
return listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n
for (index in indices) {\n accumulator = operation(index, accumulator, this[index])\n
result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation
values generated by applying [operation] from left to right\n * to each element and current accumulator value that
starts with the first element of this array.\n * \n * Note that `acc` value passed to [operation] function should not be
mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that
takes current accumulator value and the element, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningReduce\n
*\n*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun <S, T : S>
Array<out T>.runningReduce(operation: (acc: S, T) -> S): List<S> {\n if (isEmpty()) return emptyList()\n var
accumulator: S = this[0]\n val result = ArrayList<S>(size).apply { add(accumulator) }\n for (index in 1 until
size) {\n accumulator = operation(accumulator, this[index])\n result.add(accumulator)\n }\n return
result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from
left to right\n * to each element and current accumulator value that starts with the first element of this array.\n * \n *
@param [operation] function that takes current accumulator value and an element, and calculates the next
accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n
*\n*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.runningReduce(operation:
(acc: Byte, Byte) -> Byte): List<Byte> {\n if (isEmpty()) return emptyList()\n var accumulator = this[0]\n val
result = ArrayList<Byte>(size).apply { add(accumulator) }\n for (index in 1 until size) {\n accumulator =
operation(accumulator, this[index])\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list
containing successive accumulation values generated by applying [operation] from left to right\n * to each element
and current accumulator value that starts with the first element of this array.\n * \n * @param [operation] function
that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningReduce\n
*\n*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.runningReduce(operation:
(acc: Short, Short) -> Short): List<Short> {\n if (isEmpty()) return emptyList()\n var accumulator = this[0]\n
val result = ArrayList<Short>(size).apply { add(accumulator) }\n for (index in 1 until size) {\n accumulator =
operation(accumulator, this[index])\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list
containing successive accumulation values generated by applying [operation] from left to right\n * to each element

```

and current accumulator value that starts with the first element of this array.

```

    * @param [operation] function
    that takes current accumulator value and an element, and calculates the next accumulator value.
    * @sample
    samples.collections.Collections.Aggregates.runningReduce
    */
    @SinceKotlin("1.4")
    @kotlin.internal.InlineOnly
    public inline fun IntArray.runningReduce(operation: (acc: Int, Int) -> Int): List<Int> {
        if (isEmpty()) return emptyList()
        var accumulator = this[0]
        val result = ArrayList<Int>(size).apply { add(accumulator) }
        for (index in 1 until size) {
            accumulator = operation(accumulator, this[index])
            result.add(accumulator)
        }
        return result
    }
    * Returns a list
    containing successive accumulation values generated by applying [operation] from left to right
    * to each element
    and current accumulator value that starts with the first element of this array.
    * @param [operation] function
    that takes current accumulator value and an element, and calculates the next accumulator value.
    * @sample
    samples.collections.Collections.Aggregates.runningReduce
    */
    @SinceKotlin("1.4")
    @kotlin.internal.InlineOnly
    public inline fun LongArray.runningReduce(operation: (acc: Long, Long) -> Long): List<Long> {
        if (isEmpty()) return emptyList()
        var accumulator = this[0]
        val result = ArrayList<Long>(size).apply { add(accumulator) }
        for (index in 1 until size) {
            accumulator = operation(accumulator, this[index])
            result.add(accumulator)
        }
        return result
    }
    * Returns a list
    containing successive accumulation values generated by applying [operation] from left to right
    * to each element
    and current accumulator value that starts with the first element of this array.
    * @param [operation] function
    that takes current accumulator value and an element, and calculates the next accumulator value.
    * @sample
    samples.collections.Collections.Aggregates.runningReduce
    */
    @SinceKotlin("1.4")
    @kotlin.internal.InlineOnly
    public inline fun FloatArray.runningReduce(operation: (acc: Float, Float) -> Float): List<Float> {
        if (isEmpty()) return emptyList()
        var accumulator = this[0]
        val result = ArrayList<Float>(size).apply { add(accumulator) }
        for (index in 1 until size) {
            accumulator = operation(accumulator, this[index])
            result.add(accumulator)
        }
        return result
    }
    * Returns a list
    containing successive accumulation values generated by applying [operation] from left to right
    * to each element
    and current accumulator value that starts with the first element of this array.
    * @param [operation] function
    that takes current accumulator value and an element, and calculates the next accumulator value.
    * @sample
    samples.collections.Collections.Aggregates.runningReduce
    */
    @SinceKotlin("1.4")
    @kotlin.internal.InlineOnly
    public inline fun DoubleArray.runningReduce(operation: (acc: Double, Double) -> Double): List<Double> {
        if (isEmpty()) return emptyList()
        var accumulator = this[0]
        val result = ArrayList<Double>(size).apply { add(accumulator) }
        for (index in 1 until size) {
            accumulator = operation(accumulator, this[index])
            result.add(accumulator)
        }
        return result
    }
    * Returns a list containing successive accumulation values generated by applying [operation] from left to right
    * to each element and current accumulator value that starts with the first element of this array.
    * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
    * @sample
    samples.collections.Collections.Aggregates.runningReduce
    */
    @SinceKotlin("1.4")
    @kotlin.internal.InlineOnly
    public inline fun BooleanArray.runningReduce(operation: (acc: Boolean, Boolean) -> Boolean): List<Boolean> {
        if (isEmpty()) return emptyList()
        var accumulator = this[0]
        val result = ArrayList<Boolean>(size).apply { add(accumulator) }
        for (index in 1 until size) {
            accumulator = operation(accumulator, this[index])
            result.add(accumulator)
        }
        return result
    }
    * Returns a list containing successive accumulation values generated by applying [operation] from left to right
    * to each element and current accumulator value that starts with the first element of this array.
    * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
    * @sample
    samples.collections.Collections.Aggregates.runningReduce
    */
    @SinceKotlin("1.4")
    @kotlin.internal.InlineOnly
    public inline fun CharArray.runningReduce(operation: (acc: Char, Char) -> Char): List<Char> {
        if (isEmpty()) return emptyList()
        var accumulator = this[0]
        val result = ArrayList<Char>(size).apply { add(accumulator) }
        for (index in 1 until size) {
            accumulator = operation(accumulator, this[index])
            result.add(accumulator)
        }
        return result
    }
    * Returns a list
    containing successive accumulation values generated by applying [operation] from left to right
    * to each element,

```

its index in the original array and current accumulator value that starts with the first element of this array.

Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.

@param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.

@sample samples.collections.Collections.Aggregates.runningReduce

```

@SinceKotlin("1.4")
public inline fun <S, T : S> Array<out T>.runningReduceIndexed(operation: (index: Int, acc: S, T) -> S): List<S> {
    if (isEmpty()) return emptyList()
    var accumulator: S = this[0]
    val result = ArrayList<S>(size).apply { add(accumulator) }
    for (index in 1 until size) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

```

Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original array and current accumulator value that starts with the first element of this array.

@param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.

@sample samples.collections.Collections.Aggregates.runningReduce

```

@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun
ByteArray.runningReduceIndexed(operation: (index: Int, acc: Byte, Byte) -> Byte): List<Byte> {
    if (isEmpty())
return emptyList()
    var accumulator = this[0]
    val result = ArrayList<Byte>(size).apply { add(accumulator) }
    for (index in 1 until size) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

```

Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original array and current accumulator value that starts with the first element of this array.

@param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.

@sample samples.collections.Collections.Aggregates.runningReduce

```

@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun
ShortArray.runningReduceIndexed(operation: (index: Int, acc: Short, Short) -> Short): List<Short> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    val result = ArrayList<Short>(size).apply { add(accumulator) }
    for (index in 1 until size) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

```

Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original array and current accumulator value that starts with the first element of this array.

@param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.

@sample samples.collections.Collections.Aggregates.runningReduce

```

@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun
IntArray.runningReduceIndexed(operation: (index: Int, acc: Int, Int) -> Int): List<Int> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    val result = ArrayList<Int>(size).apply { add(accumulator) }
    for (index in 1 until size) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

```

Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original array and current accumulator value that starts with the first element of this array.

@param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.

@sample samples.collections.Collections.Aggregates.runningReduce

```

@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun
LongArray.runningReduceIndexed(operation: (index: Int, acc: Long, Long) -> Long): List<Long> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    val result = ArrayList<Long>(size).apply { add(accumulator) }
    for (index in 1 until size) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

```

Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original array and current accumulator value that starts with the first element of this array.

@param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next

```

accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun
FloatArray.runningReduceIndexed(operation: (index: Int, acc: Float, Float) -> Float): List<Float> {\n  if
(isEmpty()) return emptyList()\n  var accumulator = this[0]\n  val result = ArrayList<Float>(size).apply {
add(accumulator) }\n  for (index in 1 until size) {\n    accumulator = operation(index, accumulator, this[index])\n
    result.add(accumulator)\n  }\n  return result\n}\n\n/**\n * Returns a list containing successive accumulation
values generated by applying [operation] from left to right\n * to each element, its index in the original array and
current accumulator value that starts with the first element of this array.\n * \n * @param [operation] function that
takes the index of an element, current accumulator value\n * and the element itself, and calculates the next
accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun
DoubleArray.runningReduceIndexed(operation: (index: Int, acc: Double, Double) -> Double): List<Double> {\n  if
(isEmpty()) return emptyList()\n  var accumulator = this[0]\n  val result = ArrayList<Double>(size).apply {
add(accumulator) }\n  for (index in 1 until size) {\n    accumulator = operation(index, accumulator, this[index])\n
    result.add(accumulator)\n  }\n  return result\n}\n\n/**\n * Returns a list containing successive accumulation
values generated by applying [operation] from left to right\n * to each element, its index in the original array and
current accumulator value that starts with the first element of this array.\n * \n * @param [operation] function that
takes the index of an element, current accumulator value\n * and the element itself, and calculates the next
accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun
BooleanArray.runningReduceIndexed(operation: (index: Int, acc: Boolean, Boolean) -> Boolean): List<Boolean>
{\n  if (isEmpty()) return emptyList()\n  var accumulator = this[0]\n  val result =
ArrayList<Boolean>(size).apply { add(accumulator) }\n  for (index in 1 until size) {\n    accumulator =
operation(index, accumulator, this[index])\n    result.add(accumulator)\n  }\n  return result\n}\n\n/**\n *
Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to
each element, its index in the original array and current accumulator value that starts with the first element of this
array.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and
the element itself, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningReduce\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun
CharArray.runningReduceIndexed(operation: (index: Int, acc: Char, Char) -> Char): List<Char> {\n  if (isEmpty())
return emptyList()\n  var accumulator = this[0]\n  val result = ArrayList<Char>(size).apply { add(accumulator)
}\n  for (index in 1 until size) {\n    accumulator = operation(index, accumulator, this[index])\n
    result.add(accumulator)\n  }\n  return result\n}\n\n/**\n * Returns a list containing successive accumulation
values generated by applying [operation] from left to right\n * to each element and current accumulator value that
starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n *
otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current
accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.scan\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun <T, R>
Array<out T>.scan(initial: R, operation: (acc: R, T) -> R): List<R> {\n  return runningFold(initial,
operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation]
from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note
that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous
value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element,
and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
inline fun <R> ByteArray.scan(initial: R, operation: (acc: R, Byte) -> R): List<R> {\n  return runningFold(initial,

```

operation)\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n *\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <R> ShortArray.scan(initial: R, operation: (acc: R, Short) -> R): List<R> {\n return runningFold(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n *\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <R> IntArray.scan(initial: R, operation: (acc: R, Int) -> R): List<R> {\n return runningFold(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n *\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <R> LongArray.scan(initial: R, operation: (acc: R, Long) -> R): List<R> {\n return runningFold(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n *\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <R> FloatArray.scan(initial: R, operation: (acc: R, Float) -> R): List<R> {\n return runningFold(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n *\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <R> DoubleArray.scan(initial: R, operation: (acc: R, Double) -> R): List<R> {\n return runningFold(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n *\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <R> BooleanArray.scan(initial: R, operation: (acc: R, Boolean) -> R): List<R> {\n return runningFold(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial]

value.
Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.
@param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
@sample samples.collections.Collections.Aggregates.scan

```

*\/n@SinceKotlin("1.4")n@WasExperimental(ExperimentalStdlibApi::class)n@kotlin.internal.InlineOnlynpublic inline fun <R> CharArray.scan(initial: R, operation: (acc: R, Char) -> R): List<R> {n return runningFold(initial, operation)n}n/n/**n * Returns a list containing successive accumulation values generated by applying [operation] from left to rightn * to each element, its index in the original array and current accumulator value that starts with [initial] value.n * Note that `acc` value passed to [operation] function should not be mutated;n * otherwise it would affect the previous value in resulting list.n * @param [operation] function that takes the index of an element, current accumulator valuen * and the element itself, and calculates the next accumulator value.n * @sample samples.collections.Collections.Aggregates.scan
*\/n@SinceKotlin("1.4")n@WasExperimental(ExperimentalStdlibApi::class)npublic inline fun <T, R> Array<out T>.scanIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): List<R> {n return runningFoldIndexed(initial, operation)n}n/n/**n * Returns a list containing successive accumulation values generated by applying [operation] from left to rightn * to each element, its index in the original array and current accumulator value that starts with [initial] value.n * Note that `acc` value passed to [operation] function should not be mutated;n * otherwise it would affect the previous value in resulting list.n * @param [operation] function that takes the index of an element, current accumulator valuen * and the element itself, and calculates the next accumulator value.n * @sample samples.collections.Collections.Aggregates.scan
*\/n@SinceKotlin("1.4")n@WasExperimental(ExperimentalStdlibApi::class)n@kotlin.internal.InlineOnlynpublic inline fun <R> ByteArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Byte) -> R): List<R> {n return runningFoldIndexed(initial, operation)n}n/n/**n * Returns a list containing successive accumulation values generated by applying [operation] from left to rightn * to each element, its index in the original array and current accumulator value that starts with [initial] value.n * Note that `acc` value passed to [operation] function should not be mutated;n * otherwise it would affect the previous value in resulting list.n * @param [operation] function that takes the index of an element, current accumulator valuen * and the element itself, and calculates the next accumulator value.n * @sample samples.collections.Collections.Aggregates.scan
*\/n@SinceKotlin("1.4")n@WasExperimental(ExperimentalStdlibApi::class)n@kotlin.internal.InlineOnlynpublic inline fun <R> ShortArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Short) -> R): List<R> {n return runningFoldIndexed(initial, operation)n}n/n/**n * Returns a list containing successive accumulation values generated by applying [operation] from left to rightn * to each element, its index in the original array and current accumulator value that starts with [initial] value.n * Note that `acc` value passed to [operation] function should not be mutated;n * otherwise it would affect the previous value in resulting list.n * @param [operation] function that takes the index of an element, current accumulator valuen * and the element itself, and calculates the next accumulator value.n * @sample samples.collections.Collections.Aggregates.scan
*\/n@SinceKotlin("1.4")n@WasExperimental(ExperimentalStdlibApi::class)n@kotlin.internal.InlineOnlynpublic inline fun <R> IntArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Int) -> R): List<R> {n return runningFoldIndexed(initial, operation)n}n/n/**n * Returns a list containing successive accumulation values generated by applying [operation] from left to rightn * to each element, its index in the original array and current accumulator value that starts with [initial] value.n * Note that `acc` value passed to [operation] function should not be mutated;n * otherwise it would affect the previous value in resulting list.n * @param [operation] function that takes the index of an element, current accumulator valuen * and the element itself, and calculates the next accumulator value.n * @sample samples.collections.Collections.Aggregates.scan
*\/n@SinceKotlin("1.4")n@WasExperimental(ExperimentalStdlibApi::class)n@kotlin.internal.InlineOnlynpublic inline fun <R> LongArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Long) -> R): List<R> {n return runningFoldIndexed(initial, operation)n}n/n/**n * Returns a list containing successive accumulation values generated by applying [operation] from left to rightn * to each element, its index in the original array and current

```

accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n

```
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <R> FloatArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Float) -> R): List<R> {\n    return runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n
```

```
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <R> DoubleArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Double) -> R): List<R> {\n    return runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n
```

```
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <R> BooleanArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Boolean) -> R): List<R> {\n    return runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n
```

```
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <R> CharArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Char) -> R): List<R> {\n    return runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@Deprecated("Use sumOf instead.")\nReplaceWith("this.sumOf(selector)")\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic inline fun <T> Array<out T>.sumBy(selector: (T) -> Int): Int {\n    var sum: Int = 0\n    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@Deprecated("Use sumOf instead.")\nReplaceWith("this.sumOf(selector)")\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic inline fun ByteArray.sumBy(selector: (Byte) -> Int): Int {\n    var sum: Int = 0\n    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@Deprecated("Use sumOf instead.")\nReplaceWith("this.sumOf(selector)")\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic inline fun ShortArray.sumBy(selector: (Short) -> Int): Int {\n    var sum: Int = 0\n    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@Deprecated("Use sumOf instead.")\nReplaceWith("this.sumOf(selector)")\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic inline fun IntArray.sumBy(selector: (Int) -> Int): Int {\n    var sum: Int = 0\n    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@Deprecated("Use sumOf instead.")
```

```

ReplaceWith("this.sumOf(selector)")@DeprecatedSinceKotlin(warningSince = `1.5`)public inline fun
LongArray.sumBy(selector: (Long) -> Int): Int {
    var sum: Int = 0
    for (element in this) {
        sum +=
selector(element)
    }
    return sum
}
* Returns the sum of all values produced by [selector] function
applied to each element in the array.
*/@Deprecated("Use sumOf instead.")
ReplaceWith("this.sumOf(selector)")@DeprecatedSinceKotlin(warningSince = `1.5`)public inline fun
FloatArray.sumBy(selector: (Float) -> Int): Int {
    var sum: Int = 0
    for (element in this) {
        sum +=
selector(element)
    }
    return sum
}
* Returns the sum of all values produced by [selector] function
applied to each element in the array.
*/@Deprecated("Use sumOf instead.")
ReplaceWith("this.sumOf(selector)")@DeprecatedSinceKotlin(warningSince = `1.5`)public inline fun
DoubleArray.sumBy(selector: (Double) -> Int): Int {
    var sum: Int = 0
    for (element in this) {
        sum +=
selector(element)
    }
    return sum
}
* Returns the sum of all values produced by [selector] function
applied to each element in the array.
*/@Deprecated("Use sumOf instead.")
ReplaceWith("this.sumOf(selector)")@DeprecatedSinceKotlin(warningSince = `1.5`)public inline fun
BooleanArray.sumBy(selector: (Boolean) -> Int): Int {
    var sum: Int = 0
    for (element in this) {
        sum +=
selector(element)
    }
    return sum
}
* Returns the sum of all values produced by [selector] function
applied to each element in the array.
*/@Deprecated("Use sumOf instead.")
ReplaceWith("this.sumOf(selector)")@DeprecatedSinceKotlin(warningSince = `1.5`)public inline fun
CharArray.sumBy(selector: (Char) -> Int): Int {
    var sum: Int = 0
    for (element in this) {
        sum +=
selector(element)
    }
    return sum
}
* Returns the sum of all values produced by [selector] function
applied to each element in the array.
*/@Deprecated("Use sumOf instead.")
ReplaceWith("this.sumOf(selector)")@DeprecatedSinceKotlin(warningSince = `1.5`)public inline fun <T>
Array<out T>.sumByDouble(selector: (T) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
* Returns the sum of all values produced by
[selector] function applied to each element in the array.
*/@Deprecated("Use sumOf instead.")
ReplaceWith("this.sumOf(selector)")@DeprecatedSinceKotlin(warningSince = `1.5`)public inline fun
ByteArray.sumByDouble(selector: (Byte) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
* Returns the sum of all values produced by
[selector] function applied to each element in the array.
*/@Deprecated("Use sumOf instead.")
ReplaceWith("this.sumOf(selector)")@DeprecatedSinceKotlin(warningSince = `1.5`)public inline fun
ShortArray.sumByDouble(selector: (Short) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
* Returns the sum of all values produced by
[selector] function applied to each element in the array.
*/@Deprecated("Use sumOf instead.")
ReplaceWith("this.sumOf(selector)")@DeprecatedSinceKotlin(warningSince = `1.5`)public inline fun
IntArray.sumByDouble(selector: (Int) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
* Returns the sum of all values produced by [selector]
function applied to each element in the array.
*/@Deprecated("Use sumOf instead.")
ReplaceWith("this.sumOf(selector)")@DeprecatedSinceKotlin(warningSince = `1.5`)public inline fun
LongArray.sumByDouble(selector: (Long) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
* Returns the sum of all values produced by
[selector] function applied to each element in the array.
*/@Deprecated("Use sumOf instead.")
ReplaceWith("this.sumOf(selector)")@DeprecatedSinceKotlin(warningSince = `1.5`)public inline fun
FloatArray.sumByDouble(selector: (Float) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
* Returns the sum of all values produced by
[selector] function applied to each element in the array.
*/@Deprecated("Use sumOf instead.")
ReplaceWith("this.sumOf(selector)")@DeprecatedSinceKotlin(warningSince = `1.5`)public inline fun
DoubleArray.sumByDouble(selector: (Double) -> Double): Double {
    var sum: Double = 0.0
    for (element in
this) {
        sum += selector(element)
    }
    return sum
}
* Returns the sum of all values produced by
[selector] function applied to each element in the array.
*/@Deprecated("Use sumOf instead.")

```

```

ReplaceWith("this.sumOf(selector)")\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic inline fun
BooleanArray.sumByDouble(selector: (Boolean) -> Double): Double {\n    var sum: Double = 0.0\n    for (element
in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced
by [selector] function applied to each element in the array.\n */\n@Deprecated("Use sumOf instead.")
ReplaceWith("this.sumOf(selector)")\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic inline fun
CharArray.sumByDouble(selector: (Char) -> Double): Double {\n    var sum: Double = 0.0\n    for (element in this)
{\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the array.\n */\n\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfDouble")\n@kotlin.internal.InlineOnly\npublic inline fun
<T> Array<out T>.sumOf(selector: (T) -> Double): Double {\n    var sum: Double = 0.toDouble()\n    for (element
in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced
by [selector] function applied to each element in the array.\n */\n\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfDouble")\n@kotlin.internal.InlineOnly\npublic inline fun
ByteArray.sumOf(selector: (Byte) -> Double): Double {\n    var sum: Double = 0.toDouble()\n    for (element in
this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the array.\n */\n\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfDouble")\n@kotlin.internal.InlineOnly\npublic inline fun
ShortArray.sumOf(selector: (Short) -> Double): Double {\n    var sum: Double = 0.toDouble()\n    for (element in
this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the array.\n */\n\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfDouble")\n@kotlin.internal.InlineOnly\npublic inline fun
IntArray.sumOf(selector: (Int) -> Double): Double {\n    var sum: Double = 0.toDouble()\n    for (element in this)
{\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the array.\n */\n\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfDouble")\n@kotlin.internal.InlineOnly\npublic inline fun
LongArray.sumOf(selector: (Long) -> Double): Double {\n    var sum: Double = 0.toDouble()\n    for (element in
this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the array.\n */\n\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfDouble")\n@kotlin.internal.InlineOnly\npublic inline fun
FloatArray.sumOf(selector: (Float) -> Double): Double {\n    var sum: Double = 0.toDouble()\n    for (element in
this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the array.\n */\n\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfDouble")\n@kotlin.internal.InlineOnly\npublic inline fun
DoubleArray.sumOf(selector: (Double) -> Double): Double {\n    var sum: Double = 0.toDouble()\n    for (element
in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced
by [selector] function applied to each element in the array.\n */\n\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfDouble")\n@kotlin.internal.InlineOnly\npublic inline fun
BooleanArray.sumOf(selector: (Boolean) -> Double): Double {\n    var sum: Double = 0.toDouble()\n    for
(element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values
produced by [selector] function applied to each element in the array.\n */

```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution  
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfDouble")\n@kotlin.internal.InlineOnly\npublic inline fun  
CharArray.sumOf(selector: (Char) -> Double): Double {\n    var sum: Double = 0.toDouble()\n    for (element in  
this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by  
[selector] function applied to each element in the array.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution  
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfInt")\n@kotlin.internal.InlineOnly\npublic inline fun <T>  
Array<out T>.sumOf(selector: (T) -> Int): Int {\n    var sum: Int = 0.toInt()\n    for (element in this) {\n        sum +=  
selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function  
applied to each element in the array.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution  
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfInt")\n@kotlin.internal.InlineOnly\npublic inline fun  
ByteArray.sumOf(selector: (Byte) -> Int): Int {\n    var sum: Int = 0.toInt()\n    for (element in this) {\n        sum +=  
selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function  
applied to each element in the array.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution  
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfInt")\n@kotlin.internal.InlineOnly\npublic inline fun  
ShortArray.sumOf(selector: (Short) -> Int): Int {\n    var sum: Int = 0.toInt()\n    for (element in this) {\n        sum +=  
selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function  
applied to each element in the array.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution  
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfInt")\n@kotlin.internal.InlineOnly\npublic inline fun  
IntArray.sumOf(selector: (Int) -> Int): Int {\n    var sum: Int = 0.toInt()\n    for (element in this) {\n        sum +=  
selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function  
applied to each element in the array.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution  
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfInt")\n@kotlin.internal.InlineOnly\npublic inline fun  
LongArray.sumOf(selector: (Long) -> Int): Int {\n    var sum: Int = 0.toInt()\n    for (element in this) {\n        sum +=  
selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function  
applied to each element in the array.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution  
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfInt")\n@kotlin.internal.InlineOnly\npublic inline fun  
FloatArray.sumOf(selector: (Float) -> Int): Int {\n    var sum: Int = 0.toInt()\n    for (element in this) {\n        sum +=  
selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function  
applied to each element in the array.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution  
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfInt")\n@kotlin.internal.InlineOnly\npublic inline fun  
DoubleArray.sumOf(selector: (Double) -> Int): Int {\n    var sum: Int = 0.toInt()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector]  
function applied to each element in the array.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution  
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfInt")\n@kotlin.internal.InlineOnly\npublic inline fun  
BooleanArray.sumOf(selector: (Boolean) -> Int): Int {\n    var sum: Int = 0.toInt()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector]  
function applied to each element in the array.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution  
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfInt")\n@kotlin.internal.InlineOnly\npublic inline fun  
CharArray.sumOf(selector: (Char) -> Int): Int {\n    var sum: Int = 0.toInt()\n    for (element in this) {\n        sum +=
```

```

selector(element)\n } \n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the array.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfLong")\n@kotlin.internal.InlineOnly\npublic inline fun
<T> Array<out T>.sumOf(selector: (T) -> Long): Long {\n var sum: Long = 0.toLong()\n for (element in this)
{\n sum += selector(element)\n } \n return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the array.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfLong")\n@kotlin.internal.InlineOnly\npublic inline fun
ByteArray.sumOf(selector: (Byte) -> Long): Long {\n var sum: Long = 0.toLong()\n for (element in this) {\n
sum += selector(element)\n } \n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector]
function applied to each element in the array.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfLong")\n@kotlin.internal.InlineOnly\npublic inline fun
ShortArray.sumOf(selector: (Short) -> Long): Long {\n var sum: Long = 0.toLong()\n for (element in this) {\n
sum += selector(element)\n } \n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector]
function applied to each element in the array.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfLong")\n@kotlin.internal.InlineOnly\npublic inline fun
IntArray.sumOf(selector: (Int) -> Long): Long {\n var sum: Long = 0.toLong()\n for (element in this) {\n
sum += selector(element)\n } \n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector]
function applied to each element in the array.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfLong")\n@kotlin.internal.InlineOnly\npublic inline fun
LongArray.sumOf(selector: (Long) -> Long): Long {\n var sum: Long = 0.toLong()\n for (element in this) {\n
sum += selector(element)\n } \n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector]
function applied to each element in the array.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfLong")\n@kotlin.internal.InlineOnly\npublic inline fun
FloatArray.sumOf(selector: (Float) -> Long): Long {\n var sum: Long = 0.toLong()\n for (element in this) {\n
sum += selector(element)\n } \n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector]
function applied to each element in the array.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfLong")\n@kotlin.internal.InlineOnly\npublic inline fun
DoubleArray.sumOf(selector: (Double) -> Long): Long {\n var sum: Long = 0.toLong()\n for (element in this)
{\n sum += selector(element)\n } \n return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the array.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfLong")\n@kotlin.internal.InlineOnly\npublic inline fun
BooleanArray.sumOf(selector: (Boolean) -> Long): Long {\n var sum: Long = 0.toLong()\n for (element in this)
{\n sum += selector(element)\n } \n return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the array.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfLong")\n@kotlin.internal.InlineOnly\npublic inline fun
CharArray.sumOf(selector: (Char) -> Long): Long {\n var sum: Long = 0.toLong()\n for (element in this) {\n
sum += selector(element)\n } \n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector]
function applied to each element in the array.\n
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution

```

ByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfUInt\")\n@WasExperimental(ExperimentalUnsignedType
s::class)\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.sumOf(selector: (T) -> UInt): UInt {\n
var sum: UInt = 0.toUInt()\n for (element in this) {\n sum += selector(element)\n }\n return
sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the
array.\n

*\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfUInt\")\n@WasExperimental(ExperimentalUnsignedType
s::class)\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.sumOf(selector: (Byte) -> UInt): UInt {\n
var sum: UInt = 0.toUInt()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n

*\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfUInt\")\n@WasExperimental(ExperimentalUnsignedType
s::class)\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.sumOf(selector: (Short) -> UInt): UInt {\n
var sum: UInt = 0.toUInt()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n

*\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfUInt\")\n@WasExperimental(ExperimentalUnsignedType
s::class)\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.sumOf(selector: (Int) -> UInt): UInt {\n
var sum: UInt = 0.toUInt()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n

*\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfUInt\")\n@WasExperimental(ExperimentalUnsignedType
s::class)\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.sumOf(selector: (Long) -> UInt): UInt {\n
var sum: UInt = 0.toUInt()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n

*\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfUInt\")\n@WasExperimental(ExperimentalUnsignedType
s::class)\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.sumOf(selector: (Float) -> UInt): UInt {\n
var sum: UInt = 0.toUInt()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n

*\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfUInt\")\n@WasExperimental(ExperimentalUnsignedType
s::class)\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.sumOf(selector: (Double) -> UInt): UInt {\n
var sum: UInt = 0.toUInt()\n for (element in this) {\n sum += selector(element)\n }\n return
sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the
array.\n

*\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfUInt\")\n@WasExperimental(ExperimentalUnsignedType
s::class)\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.sumOf(selector: (Boolean) -> UInt): UInt
{\n
var sum: UInt = 0.toUInt()\n for (element in this) {\n sum += selector(element)\n }\n return
sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the
array.\n

*\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfUInt\")\n@WasExperimental(ExperimentalUnsignedType
s::class)\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.sumOf(selector: (Char) -> UInt): UInt {\n
var sum: UInt = 0.toUInt()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n

*\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution

ByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfULong\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.sumOf(selector: (T) -> ULong): ULong {\n var sum: ULong = 0.toULong()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/*\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n

\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfULong\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.sumOf(selector: (Byte) -> ULong): ULong {\n var sum: ULong = 0.toULong()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n

\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfULong\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.sumOf(selector: (Short) -> ULong): ULong {\n var sum: ULong = 0.toULong()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n

\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfULong\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.sumOf(selector: (Int) -> ULong): ULong {\n var sum: ULong = 0.toULong()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n

\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfULong\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.sumOf(selector: (Long) -> ULong): ULong {\n var sum: ULong = 0.toULong()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n

\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfULong\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.sumOf(selector: (Float) -> ULong): ULong {\n var sum: ULong = 0.toULong()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n

\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfULong\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.sumOf(selector: (Double) -> ULong): ULong {\n var sum: ULong = 0.toULong()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n

\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfULong\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.sumOf(selector: (Boolean) -> ULong): ULong {\n var sum: ULong = 0.toULong()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n

*\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution

```

ByLambdaReturnType\n@kotlin.jvm.JvmName("\sumOfULong")\n@WasExperimental(ExperimentalUnsignedTy
pes::class)\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.sumOf(selector: (Char) -> ULong): ULong
{\n    var sum: ULong = 0.toULong()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return
sum}\n\n/**\n * Returns an original collection containing all the non-`null` elements, throwing an
[IllegalArgumentException] if there are any `null` elements.\n */\npublic fun <T : Any>
Array<T?>.requireNonNulls(): Array<T> {\n    for (element in this) {\n        if (element == null) {\n            throw
IllegalArgumentException("\null element found in $this.")\n        }\n    }\n    @Suppress("\UNCHECKED_CAST")\n    return this as Array<T>}\n\n/**\n * Splits the original array into pair
of lists,\n * where *first* list contains elements for which [predicate] yielded `true`,\n * while *second* list contains
elements for which [predicate] yielded `false`.\n * \n * @sample
samples.collections.Arrays.Transformations.partitionArrayOfPrimitives\n */\npublic inline fun <T> Array<out
T>.partition(predicate: (T) -> Boolean): Pair<List<T>, List<T>> {\n    val first = ArrayList<T>()\n    val second =
ArrayList<T>()\n    for (element in this) {\n        if (predicate(element)) {\n            first.add(element)\n        } else {\n
            second.add(element)\n        }\n    }\n    return Pair(first, second)\n}\n\n/**\n * Splits the original array into pair
of lists,\n * where *first* list contains elements for which [predicate] yielded `true`,\n * while *second* list contains
elements for which [predicate] yielded `false`.\n * \n * @sample
samples.collections.Arrays.Transformations.partitionArrayOfPrimitives\n */\npublic inline fun
ByteArray.partition(predicate: (Byte) -> Boolean): Pair<List<Byte>, List<Byte>> {\n    val first =
ArrayList<Byte>()\n    val second = ArrayList<Byte>()\n    for (element in this) {\n        if (predicate(element)) {\n
            first.add(element)\n        } else {\n            second.add(element)\n        }\n    }\n    return Pair(first,
second)\n}\n\n/**\n * Splits the original array into pair of lists,\n * where *first* list contains elements for which
[predicate] yielded `true`,\n * while *second* list contains elements for which [predicate] yielded `false`.\n * \n *
@sample samples.collections.Arrays.Transformations.partitionArrayOfPrimitives\n */\npublic inline fun
ShortArray.partition(predicate: (Short) -> Boolean): Pair<List<Short>, List<Short>> {\n    val first =
ArrayList<Short>()\n    val second = ArrayList<Short>()\n    for (element in this) {\n        if (predicate(element)) {\n
            first.add(element)\n        } else {\n            second.add(element)\n        }\n    }\n    return Pair(first,
second)\n}\n\n/**\n * Splits the original array into pair of lists,\n * where *first* list contains elements for which
[predicate] yielded `true`,\n * while *second* list contains elements for which [predicate] yielded `false`.\n * \n *
@sample samples.collections.Arrays.Transformations.partitionArrayOfPrimitives\n */\npublic inline fun
IntArray.partition(predicate: (Int) -> Boolean): Pair<List<Int>, List<Int>> {\n    val first = ArrayList<Int>()\n    val
second = ArrayList<Int>()\n    for (element in this) {\n        if (predicate(element)) {\n            first.add(element)\n
        } else {\n            second.add(element)\n        }\n    }\n    return Pair(first, second)\n}\n\n/**\n * Splits the original
array into pair of lists,\n * where *first* list contains elements for which [predicate] yielded `true`,\n * while
*second* list contains elements for which [predicate] yielded `false`.\n * \n * @sample
samples.collections.Arrays.Transformations.partitionArrayOfPrimitives\n */\npublic inline fun
LongArray.partition(predicate: (Long) -> Boolean): Pair<List<Long>, List<Long>> {\n    val first =
ArrayList<Long>()\n    val second = ArrayList<Long>()\n    for (element in this) {\n        if (predicate(element)) {\n
            first.add(element)\n        } else {\n            second.add(element)\n        }\n    }\n    return Pair(first,
second)\n}\n\n/**\n * Splits the original array into pair of lists,\n * where *first* list contains elements for which
[predicate] yielded `true`,\n * while *second* list contains elements for which [predicate] yielded `false`.\n * \n *
@sample samples.collections.Arrays.Transformations.partitionArrayOfPrimitives\n */\npublic inline fun
FloatArray.partition(predicate: (Float) -> Boolean): Pair<List<Float>, List<Float>> {\n    val first =
ArrayList<Float>()\n    val second = ArrayList<Float>()\n    for (element in this) {\n        if (predicate(element)) {\n
            first.add(element)\n        } else {\n            second.add(element)\n        }\n    }\n    return Pair(first,
second)\n}\n\n/**\n * Splits the original array into pair of lists,\n * where *first* list contains elements for which
[predicate] yielded `true`,\n * while *second* list contains elements for which [predicate] yielded `false`.\n * \n *
@sample samples.collections.Arrays.Transformations.partitionArrayOfPrimitives\n */\npublic inline fun
DoubleArray.partition(predicate: (Double) -> Boolean): Pair<List<Double>, List<Double>> {\n    val first =

```



```

samples.collections.Iterables.Operations.zipIterableWithTransform\n * \npublic inline fun <T, R, V> Array<out
T>.zip(other: Array<out R>, transform: (a: T, b: R) -> V): List<V> {\n  val size = minOf(size, other.size)\n  val
list = ArrayList<V>(size)\n  for (i in 0 until size) {\n    list.add(transform(this[i], other[i]))\n  }\n  return
list}\n\n\n\n * Returns a list of values built from the elements of `this` array and the [other] array with the same
index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length
of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n
*\npublic inline fun <R, V> ByteArray.zip(other: Array<out R>, transform: (a: Byte, b: R) -> V): List<V> {\n  val
size = minOf(size, other.size)\n  val list = ArrayList<V>(size)\n  for (i in 0 until size) {\n
list.add(transform(this[i], other[i]))\n  }\n  return list}\n\n\n\n * Returns a list of values built from the elements
of `this` array and the [other] array with the same index\n * using the provided [transform] function applied to each
pair of elements.\n * The returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n * \npublic inline fun <R, V>
ShortArray.zip(other: Array<out R>, transform: (a: Short, b: R) -> V): List<V> {\n  val size = minOf(size,
other.size)\n  val list = ArrayList<V>(size)\n  for (i in 0 until size) {\n    list.add(transform(this[i], other[i]))\n
}\n  return list}\n\n\n\n * Returns a list of values built from the elements of `this` array and the [other] array
with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned
list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n * \npublic inline fun <R, V>
IntArray.zip(other: Array<out R>, transform: (a: Int, b: R) -> V): List<V> {\n  val size = minOf(size, other.size)\n
val list = ArrayList<V>(size)\n  for (i in 0 until size) {\n    list.add(transform(this[i], other[i]))\n  }\n  return
list}\n\n\n\n * Returns a list of values built from the elements of `this` array and the [other] array with the same
index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length
of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n
*\npublic inline fun <R, V> LongArray.zip(other: Array<out R>, transform: (a: Long, b: R) -> V): List<V> {\n
val size = minOf(size, other.size)\n  val list = ArrayList<V>(size)\n  for (i in 0 until size) {\n
list.add(transform(this[i], other[i]))\n  }\n  return list}\n\n\n\n * Returns a list of values built from the elements
of `this` array and the [other] array with the same index\n * using the provided [transform] function applied to each
pair of elements.\n * The returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n * \npublic inline fun <R, V>
FloatArray.zip(other: Array<out R>, transform: (a: Float, b: R) -> V): List<V> {\n  val size = minOf(size,
other.size)\n  val list = ArrayList<V>(size)\n  for (i in 0 until size) {\n    list.add(transform(this[i], other[i]))\n
}\n  return list}\n\n\n\n * Returns a list of values built from the elements of `this` array and the [other] array
with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned
list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n * \npublic inline fun <R, V>
DoubleArray.zip(other: Array<out R>, transform: (a: Double, b: R) -> V): List<V> {\n  val size = minOf(size,
other.size)\n  val list = ArrayList<V>(size)\n  for (i in 0 until size) {\n    list.add(transform(this[i], other[i]))\n
}\n  return list}\n\n\n\n * Returns a list of values built from the elements of `this` array and the [other] array
with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned
list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n * \npublic inline fun <R, V>
BooleanArray.zip(other: Array<out R>, transform: (a: Boolean, b: R) -> V): List<V> {\n  val size = minOf(size,
other.size)\n  val list = ArrayList<V>(size)\n  for (i in 0 until size) {\n    list.add(transform(this[i], other[i]))\n
}\n  return list}\n\n\n\n * Returns a list of values built from the elements of `this` array and the [other] array
with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned
list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n * \npublic inline fun <R, V>
CharArray.zip(other: Array<out R>, transform: (a: Char, b: R) -> V): List<V> {\n  val size = minOf(size,

```

```

other.size)\n    val list = ArrayList<V>(size)\n    for (i in 0 until size) {\n        list.add(transform(this[i], other[i]))\n    }\n    return list\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun <T, R> Array<out T>.zip(other: Iterable<R>): List<Pair<T, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun <R> ByteArray.zip(other: Iterable<R>): List<Pair<Byte, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun <R> ShortArray.zip(other: Iterable<R>): List<Pair<Short, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun <R> IntArray.zip(other: Iterable<R>): List<Pair<Int, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun <R> LongArray.zip(other: Iterable<R>): List<Pair<Long, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun <R> FloatArray.zip(other: Iterable<R>): List<Pair<Float, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun <R> DoubleArray.zip(other: Iterable<R>): List<Pair<Double, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun <R> BooleanArray.zip(other: Iterable<R>): List<Pair<Boolean, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun <R> CharArray.zip(other: Iterable<R>): List<Pair<Char, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] collection with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n */\npublic inline fun <T, R, V> Array<out T>.zip(other: Iterable<R>, transform: (a: T, b: R) -> V): List<V> {\n    val arraySize = size\n    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))\n    var i = 0\n    for (element in other) {\n        if (i >= arraySize) break\n        list.add(transform(this[i++], element))\n    }\n    return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] collection with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n */\npublic inline fun <R, V> ByteArray.zip(other: Iterable<R>, transform: (a: Byte, b: R) -> V): List<V> {\n    val arraySize = size\n    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))\n    var i = 0\n    for (element in other) {\n        if (i >= arraySize) break\n        list.add(transform(this[i++], element))\n    }\n    return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] collection with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n */

```

```

*public inline fun <R, V> ShortArray.zip(other: Iterable<R>, transform: (a: Short, b: R) -> V): List<V> {
    val arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

```

* Returns a list of values built from the elements of `this` array and the [other] collection with the same index * using the provided [transform] function applied to each pair of elements. * The returned list has length of the shortest collection. * @sample

```

samples.collections.Iterables.Operations.zipIterableWithTransform

```

```

*public inline fun <R, V>
IntArray.zip(other: Iterable<R>, transform: (a: Int, b: R) -> V): List<V> {
    val arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

```

* Returns a list of values built from the elements of `this` array and the [other] collection with the same index * using the provided [transform] function applied to each pair of elements. * The returned list has length of the shortest collection. * @sample

```

samples.collections.Iterables.Operations.zipIterableWithTransform

```

```

*public inline fun <R, V> LongArray.zip(other: Iterable<R>, transform: (a: Long, b: R) -> V): List<V> {
    val arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

```

* Returns a list of values built from the elements of `this` array and the [other] collection with the same index * using the provided [transform] function applied to each pair of elements. * The returned list has length of the shortest collection. * @sample

```

samples.collections.Iterables.Operations.zipIterableWithTransform

```

```

*public inline fun <R, V>
FloatArray.zip(other: Iterable<R>, transform: (a: Float, b: R) -> V): List<V> {
    val arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

```

* Returns a list of values built from the elements of `this` array and the [other] collection with the same index * using the provided [transform] function applied to each pair of elements. * The returned list has length of the shortest collection. * @sample

```

samples.collections.Iterables.Operations.zipIterableWithTransform

```

```

*public inline fun <R, V> DoubleArray.zip(other: Iterable<R>, transform: (a: Double, b: R) -> V): List<V> {
    val arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

```

* Returns a list of values built from the elements of `this` array and the [other] collection with the same index * using the provided [transform] function applied to each pair of elements. * The returned list has length of the shortest collection. * @sample

```

samples.collections.Iterables.Operations.zipIterableWithTransform

```

```

*public inline fun <R, V>
BooleanArray.zip(other: Iterable<R>, transform: (a: Boolean, b: R) -> V): List<V> {
    val arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

```

* Returns a list of values built from the elements of `this` array and the [other] collection with the same index * using the provided [transform] function applied to each pair of elements. * The returned list has length of the shortest collection. * @sample

```

samples.collections.Iterables.Operations.zipIterableWithTransform

```

```

*public inline fun <R, V> CharArray.zip(other: Iterable<R>, transform: (a: Char, b: R) -> V): List<V> {
    val arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

```

* Returns a list of pairs built from the elements of `this` array and the [other] array with the same index. * The returned list has length of the shortest collection. * @sample

```

samples.collections.Iterables.Operations.zipIterable

```

```

*public infix fun ByteArray.zip(other: ByteArray):
List<Pair<Byte, Byte>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

```

* Returns a list of pairs built from the elements of `this` array and the [other] array with the same index. * The returned list has length of the shortest collection. * @sample

```

samples.collections.Iterables.Operations.zipIterable

```

```

*public infix fun

```

```

ShortArray.zip(other: ShortArray): List<Pair<Short, Short>> {\n  return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun IntArray.zip(other: IntArray):
List<Pair<Int, Int>> {\n  return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the
elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest
collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun
LongArray.zip(other: LongArray): List<Pair<Long, Long>> {\n  return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n *
Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The
returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun FloatArray.zip(other: FloatArray):
List<Pair<Float, Float>> {\n  return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from
the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest
collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun
DoubleArray.zip(other: DoubleArray): List<Pair<Double, Double>> {\n  return zip(other) { t1, t2 -> t1 to t2
}\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the same
index.\n * The returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun BooleanArray.zip(other: BooleanArray):
List<Pair<Boolean, Boolean>> {\n  return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built
from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the
shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun
CharArray.zip(other: CharArray): List<Pair<Char, Char>> {\n  return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n *
Returns a list of values built from the elements of `this` array and the [other] array with the same index\n * using the
provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest
array.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n */\npublic inline fun
<V> ByteArray.zip(other: ByteArray, transform: (a: Byte, b: Byte) -> V): List<V> {\n  val size = minOf(size,
other.size)\n  val list = ArrayList<V>(size)\n  for (i in 0 until size) {\n    list.add(transform(this[i], other[i]))\n
}\n  return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array
with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned
list has length of the shortest array.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n */\npublic inline fun <V>
ShortArray.zip(other: ShortArray, transform: (a: Short, b: Short) -> V): List<V> {\n  val size = minOf(size,
other.size)\n  val list = ArrayList<V>(size)\n  for (i in 0 until size) {\n    list.add(transform(this[i], other[i]))\n
}\n  return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array
with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned
list has length of the shortest array.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n */\npublic inline fun <V> IntArray.zip(other:
IntArray, transform: (a: Int, b: Int) -> V): List<V> {\n  val size = minOf(size, other.size)\n  val list =
ArrayList<V>(size)\n  for (i in 0 until size) {\n    list.add(transform(this[i], other[i]))\n  }\n  return
list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array with the same
index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length
of the shortest array.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n
*/\npublic inline fun <V> LongArray.zip(other: LongArray, transform: (a: Long, b: Long) -> V): List<V> {\n  val
size = minOf(size, other.size)\n  val list = ArrayList<V>(size)\n  for (i in 0 until size) {\n
list.add(transform(this[i], other[i]))\n  }\n  return list\n}\n\n/**\n * Returns a list of values built from the elements
of `this` array and the [other] array with the same index\n * using the provided [transform] function applied to each
pair of elements.\n * The returned list has length of the shortest array.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterableWithTransform\n */\npublic inline fun <V>

```



```

}\n  if (limit >= 0 && count > limit) buffer.append(truncated)\n  buffer.append(postfix)\n  return
buffer\n}\n\n/**\n * Appends the string from all the elements separated using [separator] and using the given
[prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value of
[limit], in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string (which
defaults to "...").\n * \n * @sample samples.collections.Collections.Transformations.joinTo\n * \npublic fun <A :
Appendable> IntArray.joinTo(buffer: A, separator: CharSequence = "\", "\", prefix: CharSequence = "\"", postfix:
CharSequence = "\"", limit: Int = -1, truncated: CharSequence = "...", transform: ((Int) -> CharSequence)? = null):
A {\n  buffer.append(prefix)\n  var count = 0\n  for (element in this) {\n    if (++count > 1)
buffer.append(separator)\n    if (limit < 0 || count <= limit) {\n      if (transform != null)\nbuffer.append(transform(element))\n      else\n        buffer.append(element.toString())\n    } else break\n  }\n  if (limit >= 0 && count > limit) buffer.append(truncated)\n  buffer.append(postfix)\n  return
buffer\n}\n\n/**\n * Appends the string from all the elements separated using [separator] and using the given
[prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value of
[limit], in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string (which
defaults to "...").\n * \n * @sample samples.collections.Collections.Transformations.joinTo\n * \npublic fun <A :
Appendable> LongArray.joinTo(buffer: A, separator: CharSequence = "\", "\", prefix: CharSequence = "\"", postfix:
CharSequence = "\"", limit: Int = -1, truncated: CharSequence = "...", transform: ((Long) -> CharSequence)? =
null): A {\n  buffer.append(prefix)\n  var count = 0\n  for (element in this) {\n    if (++count > 1)
buffer.append(separator)\n    if (limit < 0 || count <= limit) {\n      if (transform != null)\nbuffer.append(transform(element))\n      else\n        buffer.append(element.toString())\n    } else break\n  }\n  if (limit >= 0 && count > limit) buffer.append(truncated)\n  buffer.append(postfix)\n  return
buffer\n}\n\n/**\n * Appends the string from all the elements separated using [separator] and using the given
[prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value of
[limit], in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string (which
defaults to "...").\n * \n * @sample samples.collections.Collections.Transformations.joinTo\n * \npublic fun <A :
Appendable> FloatArray.joinTo(buffer: A, separator: CharSequence = "\", "\", prefix: CharSequence = "\"", postfix:
CharSequence = "\"", limit: Int = -1, truncated: CharSequence = "...", transform: ((Float) -> CharSequence)? =
null): A {\n  buffer.append(prefix)\n  var count = 0\n  for (element in this) {\n    if (++count > 1)
buffer.append(separator)\n    if (limit < 0 || count <= limit) {\n      if (transform != null)\nbuffer.append(transform(element))\n      else\n        buffer.append(element.toString())\n    } else break\n  }\n  if (limit >= 0 && count > limit) buffer.append(truncated)\n  buffer.append(postfix)\n  return
buffer\n}\n\n/**\n * Appends the string from all the elements separated using [separator] and using the given
[prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value of
[limit], in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string (which
defaults to "...").\n * \n * @sample samples.collections.Collections.Transformations.joinTo\n * \npublic fun <A :
Appendable> DoubleArray.joinTo(buffer: A, separator: CharSequence = "\", "\", prefix: CharSequence = "\"", postfix:
CharSequence = "\"", limit: Int = -1, truncated: CharSequence = "...", transform: ((Double) -> CharSequence)? =
null): A {\n  buffer.append(prefix)\n  var count = 0\n  for (element in this) {\n    if (++count > 1)
buffer.append(separator)\n    if (limit < 0 || count <= limit) {\n      if (transform != null)\nbuffer.append(transform(element))\n      else\n        buffer.append(element.toString())\n    } else break\n  }\n  if (limit >= 0 && count > limit) buffer.append(truncated)\n  buffer.append(postfix)\n  return
buffer\n}\n\n/**\n * Appends the string from all the elements separated using [separator] and using the given
[prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value of
[limit], in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string (which
defaults to "...").\n * \n * @sample samples.collections.Collections.Transformations.joinTo\n * \npublic fun <A :
Appendable> BooleanArray.joinTo(buffer: A, separator: CharSequence = "\", "\", prefix: CharSequence = "\"", postfix:
CharSequence = "\"", limit: Int = -1, truncated: CharSequence = "...", transform: ((Boolean) ->
CharSequence)? = null): A {\n  buffer.append(prefix)\n  var count = 0\n  for (element in this) {\n    if (++count

```

```

> 1) buffer.append(separator)\n    if (limit < 0 || count <= limit) {\n        if (transform != null)\n        buffer.append(transform(element))\n        else\n        buffer.append(element.toString())\n    } else break\n}\n if (limit >= 0 && count > limit) buffer.append(truncated)\n buffer.append(postfix)\n return
buffer\n}\n\n/**\n * Appends the string from all the elements separated using [separator] and using the given
[prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value of
[limit], in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string (which
defaults to "...").\n * \n * @sample samples.collections.Collections.Transformations.joinTo\n * \n\npublic fun <A :
Appendable> CharArray.joinTo(buffer: A, separator: CharSequence = "\", \"", prefix: CharSequence = "\", postfix:
CharSequence = "\", limit: Int = -1, truncated: CharSequence = "...\", transform: ((Char) -> CharSequence)? =
null): A {\n    buffer.append(prefix)\n    var count = 0\n    for (element in this) {\n        if (++count > 1)
buffer.append(separator)\n        if (limit < 0 || count <= limit) {\n            if (transform != null)\n            buffer.append(transform(element))\n            else\n            buffer.append(element)\n        } else break\n    }\n    if
(limit >= 0 && count > limit) buffer.append(truncated)\n    buffer.append(postfix)\n    return buffer\n}\n\n/**\n * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if
supplied.\n * \n * If the collection could be huge, you can specify a non-negative value of [limit], in which case only
the first [limit]\n * elements will be appended, followed by the [truncated] string (which defaults to "...").\n * \n *
@sample samples.collections.Collections.Transformations.joinToString\n * \n\npublic fun <T> Array<out
T>.joinToString(separator: CharSequence = "\", \"", prefix: CharSequence = "\", postfix: CharSequence = "\", limit:
Int = -1, truncated: CharSequence = "...\", transform: ((T) -> CharSequence)? = null): String {\n    return
joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()\n}\n\n/**\n * Creates a
string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n * \n
* If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first
[limit]\n * elements will be appended, followed by the [truncated] string (which defaults to "...").\n * \n *
@sample samples.collections.Collections.Transformations.joinToString\n * \n\npublic fun ByteArray.joinToString(separator:
CharSequence = "\", \"", prefix: CharSequence = "\", postfix: CharSequence = "\", limit: Int = -1, truncated:
CharSequence = "...\", transform: ((Byte) -> CharSequence)? = null): String {\n    return joinTo(StringBuilder(),
separator, prefix, postfix, limit, truncated, transform).toString()\n}\n\n/**\n * Creates a string from all the elements
separated using [separator] and using the given [prefix] and [postfix] if supplied.\n * \n * If the collection could be
huge, you can specify a non-negative value of [limit], in which case only the first [limit]\n * elements will be
appended, followed by the [truncated] string (which defaults to "...").\n * \n * @sample
samples.collections.Collections.Transformations.joinToString\n * \n\npublic fun ShortArray.joinToString(separator:
CharSequence = "\", \"", prefix: CharSequence = "\", postfix: CharSequence = "\", limit: Int = -1, truncated:
CharSequence = "...\", transform: ((Short) -> CharSequence)? = null): String {\n    return joinTo(StringBuilder(),
separator, prefix, postfix, limit, truncated, transform).toString()\n}\n\n/**\n * Creates a string from all the elements
separated using [separator] and using the given [prefix] and [postfix] if supplied.\n * \n * If the collection could be
huge, you can specify a non-negative value of [limit], in which case only the first [limit]\n * elements will be
appended, followed by the [truncated] string (which defaults to "...").\n * \n * @sample
samples.collections.Collections.Transformations.joinToString\n * \n\npublic fun IntArray.joinToString(separator:
CharSequence = "\", \"", prefix: CharSequence = "\", postfix: CharSequence = "\", limit: Int = -1, truncated:
CharSequence = "...\", transform: ((Int) -> CharSequence)? = null): String {\n    return joinTo(StringBuilder(),
separator, prefix, postfix, limit, truncated, transform).toString()\n}\n\n/**\n * Creates a string from all the elements
separated using [separator] and using the given [prefix] and [postfix] if supplied.\n * \n * If the collection could be
huge, you can specify a non-negative value of [limit], in which case only the first [limit]\n * elements will be
appended, followed by the [truncated] string (which defaults to "...").\n * \n * @sample
samples.collections.Collections.Transformations.joinToString\n * \n\npublic fun LongArray.joinToString(separator:
CharSequence = "\", \"", prefix: CharSequence = "\", postfix: CharSequence = "\", limit: Int = -1, truncated:
CharSequence = "...\", transform: ((Long) -> CharSequence)? = null): String {\n    return joinTo(StringBuilder(),
separator, prefix, postfix, limit, truncated, transform).toString()\n}\n\n/**\n * Creates a string from all the elements

```

separated using [separator] and using the given [prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string (which defaults to \"...\").\n * \n * @sample

```

samples.collections.Collections.Transformations.joinToString\n *
\npublic fun FloatArray.joinToString(separator:
CharSequence = "\", \"\", prefix: CharSequence = \"\", postfix: CharSequence = \"\", limit: Int = -1, truncated:
CharSequence = \"...\", transform: ((Float) -> CharSequence)? = null): String {\n  return joinTo(StringBuilder(),
separator, prefix, postfix, limit, truncated, transform).toString()\n}\n\n/**\n * Creates a string from all the elements
separated using [separator] and using the given [prefix] and [postfix] if supplied.\n * \n * If the collection could be
huge, you can specify a non-negative value of [limit], in which case only the first [limit]\n * elements will be
appended, followed by the [truncated] string (which defaults to \"...\").\n * \n * @sample
samples.collections.Collections.Transformations.joinToString\n *
\npublic fun DoubleArray.joinToString(separator:
CharSequence = "\", \"\", prefix: CharSequence = \"\", postfix: CharSequence = \"\", limit: Int = -1, truncated:
CharSequence = \"...\", transform: ((Double) -> CharSequence)? = null): String {\n  return joinTo(StringBuilder(),
separator, prefix, postfix, limit, truncated, transform).toString()\n}\n\n/**\n * Creates a string from all the elements
separated using [separator] and using the given [prefix] and [postfix] if supplied.\n * \n * If the collection could be
huge, you can specify a non-negative value of [limit], in which case only the first [limit]\n * elements will be
appended, followed by the [truncated] string (which defaults to \"...\").\n * \n * @sample
samples.collections.Collections.Transformations.joinToString\n *
\npublic fun
BooleanArray.joinToString(separator: CharSequence = "\", \"\", prefix: CharSequence = \"\", postfix: CharSequence =
\"\", limit: Int = -1, truncated: CharSequence = \"...\", transform: ((Boolean) -> CharSequence)? = null): String {\n
return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()\n}\n\n/**\n * Creates
a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n * \n
* If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first
[limit]\n * elements will be appended, followed by the [truncated] string (which defaults to \"...\").\n * \n * @sample
samples.collections.Collections.Transformations.joinToString\n *
\npublic fun CharArray.joinToString(separator:
CharSequence = "\", \"\", prefix: CharSequence = \"\", postfix: CharSequence = \"\", limit: Int = -1, truncated:
CharSequence = \"...\", transform: ((Char) -> CharSequence)? = null): String {\n  return joinTo(StringBuilder(),
separator, prefix, postfix, limit, truncated, transform).toString()\n}\n\n/**\n * Creates an [Iterable] instance that
wraps the original array returning its elements when being iterated.\n *
\npublic fun <T> Array<out T>.asIterable():
Iterable<T> {\n  if (isEmpty()) return emptyList()\n  return Iterable { this.iterator() }\n}\n\n/**\n * Creates an
[Iterable] instance that wraps the original array returning its elements when being iterated.\n *
\npublic fun
ByteArray.asIterable(): Iterable<Byte> {\n  if (isEmpty()) return emptyList()\n  return Iterable { this.iterator()
}\n}\n\n/**\n * Creates an [Iterable] instance that wraps the original array returning its elements when being
iterated.\n *
\npublic fun ShortArray.asIterable(): Iterable<Short> {\n  if (isEmpty()) return emptyList()\n  return
Iterable { this.iterator() }\n}\n\n/**\n * Creates an [Iterable] instance that wraps the original array returning its
elements when being iterated.\n *
\npublic fun IntArray.asIterable(): Iterable<Int> {\n  if (isEmpty()) return
emptyList()\n  return Iterable { this.iterator() }\n}\n\n/**\n * Creates an [Iterable] instance that wraps the original
array returning its elements when being iterated.\n *
\npublic fun LongArray.asIterable(): Iterable<Long> {\n  if
(isEmpty()) return emptyList()\n  return Iterable { this.iterator() }\n}\n\n/**\n * Creates an [Iterable] instance that
wraps the original array returning its elements when being iterated.\n *
\npublic fun FloatArray.asIterable():
Iterable<Float> {\n  if (isEmpty()) return emptyList()\n  return Iterable { this.iterator() }\n}\n\n/**\n * Creates an
[Iterable] instance that wraps the original array returning its elements when being iterated.\n *
\npublic fun
DoubleArray.asIterable(): Iterable<Double> {\n  if (isEmpty()) return emptyList()\n  return Iterable {
this.iterator() }\n}\n\n/**\n * Creates an [Iterable] instance that wraps the original array returning its elements when
being iterated.\n *
\npublic fun BooleanArray.asIterable(): Iterable<Boolean> {\n  if (isEmpty()) return
emptyList()\n  return Iterable { this.iterator() }\n}\n\n/**\n * Creates an [Iterable] instance that wraps the original
array returning its elements when being iterated.\n *
\npublic fun CharArray.asIterable(): Iterable<Char> {\n  if
(isEmpty()) return emptyList()\n  return Iterable { this.iterator() }\n}\n\n/**\n * Creates a [Sequence] instance that

```

```

wraps the original array returning its elements when being iterated.\n * \n * @sample
samples.collections.Sequences.Building.sequenceFromArray\n * \n\npublic fun <T> Array<out T>.asSequence():
Sequence<T> {\n if (isEmpty()) return emptySequence()\n return Sequence { this.iterator() }\n}\n\n/**\n *
Creates a [Sequence] instance that wraps the original array returning its elements when being iterated.\n * \n *
@sample samples.collections.Sequences.Building.sequenceFromArray\n * \n\npublic fun ByteArray.asSequence():
Sequence<Byte> {\n if (isEmpty()) return emptySequence()\n return Sequence { this.iterator() }\n}\n\n/**\n *
Creates a [Sequence] instance that wraps the original array returning its elements when being iterated.\n * \n *
@sample samples.collections.Sequences.Building.sequenceFromArray\n * \n\npublic fun ShortArray.asSequence():
Sequence<Short> {\n if (isEmpty()) return emptySequence()\n return Sequence { this.iterator() }\n}\n\n/**\n *
Creates a [Sequence] instance that wraps the original array returning its elements when being iterated.\n * \n *
@sample samples.collections.Sequences.Building.sequenceFromArray\n * \n\npublic fun IntArray.asSequence():
Sequence<Int> {\n if (isEmpty()) return emptySequence()\n return Sequence { this.iterator() }\n}\n\n/**\n *
Creates a [Sequence] instance that wraps the original array returning its elements when being iterated.\n * \n *
@sample samples.collections.Sequences.Building.sequenceFromArray\n * \n\npublic fun LongArray.asSequence():
Sequence<Long> {\n if (isEmpty()) return emptySequence()\n return Sequence { this.iterator() }\n}\n\n/**\n *
Creates a [Sequence] instance that wraps the original array returning its elements when being iterated.\n * \n *
@sample samples.collections.Sequences.Building.sequenceFromArray\n * \n\npublic fun FloatArray.asSequence():
Sequence<Float> {\n if (isEmpty()) return emptySequence()\n return Sequence { this.iterator() }\n}\n\n/**\n *
Creates a [Sequence] instance that wraps the original array returning its elements when being iterated.\n * \n *
@sample samples.collections.Sequences.Building.sequenceFromArray\n * \n\npublic fun DoubleArray.asSequence():
Sequence<Double> {\n if (isEmpty()) return emptySequence()\n return Sequence { this.iterator() }\n}\n\n/**\n *
Creates a [Sequence] instance that wraps the original array returning its elements when being iterated.\n * \n *
@sample samples.collections.Sequences.Building.sequenceFromArray\n * \n\npublic fun
BooleanArray.asSequence(): Sequence<Boolean> {\n if (isEmpty()) return emptySequence()\n return Sequence
{ this.iterator() }\n}\n\n/**\n * Creates a [Sequence] instance that wraps the original array returning its elements
when being iterated.\n * \n * @sample samples.collections.Sequences.Building.sequenceFromArray\n * \n\npublic
fun CharArray.asSequence(): Sequence<Char> {\n if (isEmpty()) return emptySequence()\n return Sequence {
this.iterator() }\n}\n\n/**\n * Returns an average value of elements in the array.\n
*\n\n@kotlin.jvm.JvmName("averageOfByte")\n\npublic fun Array<out Byte>.average(): Double {\n var sum:
Double = 0.0\n var count: Int = 0\n for (element in this) {\n sum += element\n ++count\n }\n return
if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns an average value of elements in the array.\n
*\n\n@kotlin.jvm.JvmName("averageOfShort")\n\npublic fun Array<out Short>.average(): Double {\n var sum:
Double = 0.0\n var count: Int = 0\n for (element in this) {\n sum += element\n ++count\n }\n return
if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns an average value of elements in the array.\n
*\n\n@kotlin.jvm.JvmName("averageOfInt")\n\npublic fun Array<out Int>.average(): Double {\n var sum: Double
= 0.0\n var count: Int = 0\n for (element in this) {\n sum += element\n ++count\n }\n return if (count
== 0) Double.NaN else sum / count\n}\n\n/**\n * Returns an average value of elements in the array.\n
*\n\n@kotlin.jvm.JvmName("averageOfLong")\n\npublic fun Array<out Long>.average(): Double {\n var sum:
Double = 0.0\n var count: Int = 0\n for (element in this) {\n sum += element\n ++count\n }\n return
if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns an average value of elements in the array.\n
*\n\n@kotlin.jvm.JvmName("averageOfFloat")\n\npublic fun Array<out Float>.average(): Double {\n var sum:
Double = 0.0\n var count: Int = 0\n for (element in this) {\n sum += element\n ++count\n }\n return
if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns an average value of elements in the array.\n
*\n\n@kotlin.jvm.JvmName("averageOfDouble")\n\npublic fun Array<out Double>.average(): Double {\n var sum:
Double = 0.0\n var count: Int = 0\n for (element in this) {\n sum += element\n ++count\n }\n return
if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns an average value of elements in the array.\n
*\n\npublic fun ByteArray.average(): Double {\n var sum: Double = 0.0\n var count: Int = 0\n for (element in
this) {\n sum += element\n ++count\n }\n return if (count == 0) Double.NaN else sum /

```

```

count\n}\n\n/**\n * Returns an average value of elements in the array.\n */\npublic fun ShortArray.average():
Double {\n    var sum: Double = 0.0\n    var count: Int = 0\n    for (element in this) {\n        sum += element\n
    ++count\n    }\n    return if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns an average value of
elements in the array.\n */\npublic fun IntArray.average(): Double {\n    var sum: Double = 0.0\n    var count: Int =
0\n    for (element in this) {\n        sum += element\n        ++count\n    }\n    return if (count == 0) Double.NaN
else sum / count\n}\n\n/**\n * Returns an average value of elements in the array.\n */\npublic fun LongArray.average():
Double {\n    var sum: Double = 0.0\n    var count: Int = 0\n    for (element in this) {\n        sum += element\n
    ++count\n    }\n    return if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns an average value of
elements in the array.\n */\npublic fun FloatArray.average(): Double {\n    var sum: Double = 0.0\n    var count: Int
= 0\n    for (element in this) {\n        sum += element\n        ++count\n    }\n    return if (count == 0) Double.NaN
else sum / count\n}\n\n/**\n * Returns an average value of elements in the array.\n */\npublic fun
DoubleArray.average(): Double {\n    var sum: Double = 0.0\n    var count: Int = 0\n    for (element in this) {\n
sum += element\n        ++count\n    }\n    return if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns
the sum of all elements in the array.\n */\n@kotlin.jvm.JvmName("sumOfByte")\npublic fun Array<out
Byte>.sum(): Int {\n    var sum: Int = 0\n    for (element in this) {\n        sum += element\n    }\n    return
sum\n}\n\n/**\n * Returns the sum of all elements in the array.\n
*/\n@kotlin.jvm.JvmName("sumOfShort")\npublic fun Array<out Short>.sum(): Int {\n    var sum: Int = 0\n    for
(element in this) {\n        sum += element\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all elements in the
array.\n */\n@kotlin.jvm.JvmName("sumOfInt")\npublic fun Array<out Int>.sum(): Int {\n    var sum: Int = 0\n
for (element in this) {\n        sum += element\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all elements in
the array.\n */\n@kotlin.jvm.JvmName("sumOfLong")\npublic fun Array<out Long>.sum(): Long {\n    var sum:
Long = 0L\n    for (element in this) {\n        sum += element\n    }\n    return sum\n}\n\n/**\n * Returns the sum of
all elements in the array.\n */\n@kotlin.jvm.JvmName("sumOfFloat")\npublic fun Array<out Float>.sum(): Float
{\n    var sum: Float = 0.0f\n    for (element in this) {\n        sum += element\n    }\n    return sum\n}\n\n/**\n
* Returns the sum of all elements in the array.\n */\n@kotlin.jvm.JvmName("sumOfDouble")\npublic fun Array<out
Double>.sum(): Double {\n    var sum: Double = 0.0\n    for (element in this) {\n        sum += element\n    }\n
return sum\n}\n\n/**\n * Returns the sum of all elements in the array.\n */\npublic fun ByteArray.sum(): Int {\n
var sum: Int = 0\n    for (element in this) {\n        sum += element\n    }\n    return sum\n}\n\n/**\n * Returns the
sum of all elements in the array.\n */\npublic fun ShortArray.sum(): Int {\n    var sum: Int = 0\n    for (element in
this) {\n        sum += element\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all elements in the array.\n
*/\npublic fun IntArray.sum(): Int {\n    var sum: Int = 0\n    for (element in this) {\n        sum += element\n    }\n
return sum\n}\n\n/**\n * Returns the sum of all elements in the array.\n */\npublic fun LongArray.sum(): Long {\n
var sum: Long = 0L\n    for (element in this) {\n        sum += element\n    }\n    return sum\n}\n\n/**\n * Returns the
sum of all elements in the array.\n */\npublic fun FloatArray.sum(): Float {\n    var sum: Float = 0.0f\n    for
(element in this) {\n        sum += element\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all elements in the
array.\n */\npublic fun DoubleArray.sum(): Double {\n    var sum: Double = 0.0\n    for (element in this) {\n
sum += element\n    }\n    return sum\n}\n\n", "/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n
*/\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("RangesKt")\n\npackage
kotlin.ranges\n\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See:
https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\nimport kotlin.random.*\n\n/**\n * Returns a
random element from this range.\n */\n * @throws IllegalArgumentException if this range is empty.\n
*/\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline fun IntRange.random(): Int {\n    return
random(Random)\n}\n\n/**\n * Returns a random element from this range.\n */\n * @throws
IllegalArgumentException if this range is empty.\n */\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic
inline fun LongRange.random(): Long {\n    return random(Random)\n}\n\n/**\n * Returns a random element from
this range.\n */\n * @throws IllegalArgumentException if this range is empty.\n
}

```

```

*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline fun CharRange.random(): Char {\n    return
random(Random)\n}\n\n/**\n * Returns a random element from this range using the specified source of
randomness.\n * \n * @throws IllegalArgumentException if this range is empty.\n
*\n@SinceKotlin("1.3")\npublic fun IntRange.random(random: Random): Int {\n    try {\n        return
random.nextInt(this)\n    } catch(e: IllegalArgumentException) {\n        throw
NoSuchElementException(e.message)\n    }\n}\n\n/**\n * Returns a random element from this range using the
specified source of randomness.\n * \n * @throws IllegalArgumentException if this range is empty.\n
*\n@SinceKotlin("1.3")\npublic fun LongRange.random(random: Random): Long {\n    try {\n        return
random.nextLong(this)\n    } catch(e: IllegalArgumentException) {\n        throw
NoSuchElementException(e.message)\n    }\n}\n\n/**\n * Returns a random element from this range using the
specified source of randomness.\n * \n * @throws IllegalArgumentException if this range is empty.\n
*\n@SinceKotlin("1.3")\npublic fun CharRange.random(random: Random): Char {\n    try {\n        return
random.nextInt(first.code, last.code + 1).toChar()\n    } catch(e: IllegalArgumentException) {\n        throw
NoSuchElementException(e.message)\n    }\n}\n\n/**\n * Returns a random element from this range, or `null` if this
range is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npubli
c inline fun IntRange.randomOrNull(): Int? {\n    return randomOrNull(Random)\n}\n\n/**\n * Returns a random
element from this range, or `null` if this range is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npubli
c inline fun LongRange.randomOrNull(): Long? {\n    return randomOrNull(Random)\n}\n\n/**\n * Returns a
random element from this range, or `null` if this range is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npubli
c inline fun CharRange.randomOrNull(): Char? {\n    return randomOrNull(Random)\n}\n\n/**\n * Returns a
random element from this range using the specified source of randomness, or `null` if this range is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun
IntRange.randomOrNull(random: Random): Int? {\n    if (isEmpty())\n        return null\n    return
random.nextInt(this)\n}\n\n/**\n * Returns a random element from this range using the specified source of
randomness, or `null` if this range is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun
LongRange.randomOrNull(random: Random): Long? {\n    if (isEmpty())\n        return null\n    return
random.nextLong(this)\n}\n\n/**\n * Returns a random element from this range using the specified source of
randomness, or `null` if this range is empty.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun
CharRange.randomOrNull(random: Random): Char? {\n    if (isEmpty())\n        return null\n    return
random.nextInt(first.code, last.code + 1).toChar()\n}\n\n/**\n * Returns `true` if this range contains the specified
[element].\n * \n * Always returns `false` if the [element] is `null`.\n
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline operator fun IntRange.contains(element:
Int?): Boolean {\n    return element != null && contains(element)\n}\n\n/**\n * Returns `true` if this range contains
the specified [element].\n * \n * Always returns `false` if the [element] is `null`.\n
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline operator fun LongRange.contains(element:
Long?): Boolean {\n    return element != null && contains(element)\n}\n\n/**\n * Returns `true` if this range
contains the specified [element].\n * \n * Always returns `false` if the [element] is `null`.\n
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline operator fun CharRange.contains(element:
Char?): Boolean {\n    return element != null && contains(element)\n}\n\n/**\n * Checks if the specified [value]
belongs to this range.\n *\n@kotlin.jvm.JvmName("intRangeContains")\npublic operator fun
ClosedRange<Int>.contains(value: Byte): Boolean {\n    return contains(value.toInt())\n}\n\n/**\n * Checks if the
specified [value] belongs to this range.\n *\n@kotlin.jvm.JvmName("longRangeContains")\npublic operator fun
ClosedRange<Long>.contains(value: Byte): Boolean {\n    return contains(value.toLong())\n}\n\n/**\n * Checks if

```

the specified [value] belongs to this range.\n */\n@kotlin.jvm.JvmName("\shortRangeContains")\npublic operator fun ClosedRange<Short>.contains(value: Byte): Boolean {\n return contains(value.toShort())\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@Deprecated("\This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.")\n@DeprecatedSinceKotlin(warningSince = "\1.3", errorSince = "\1.4", hiddenSince = "\1.5")\n@kotlin.jvm.JvmName("\doubleRangeContains")\npublic operator fun ClosedRange<Double>.contains(value: Byte): Boolean {\n return contains(value.toDouble())\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@Deprecated("\This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.")\n@DeprecatedSinceKotlin(warningSince = "\1.3", errorSince = "\1.4", hiddenSince = "\1.5")\n@kotlin.jvm.JvmName("\floatRangeContains")\npublic operator fun ClosedRange<Float>.contains(value: Byte): Boolean {\n return contains(value.toFloat())\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@Deprecated("\This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.")\n@DeprecatedSinceKotlin(warningSince = "\1.3", errorSince = "\1.4", hiddenSince = "\1.5")\n@kotlin.jvm.JvmName("\intRangeContains")\npublic operator fun ClosedRange<Int>.contains(value: Double): Boolean {\n return value.toIntExactOrNull().let { if (it != null) contains(it) else false }\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@Deprecated("\This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.")\n@DeprecatedSinceKotlin(warningSince = "\1.3", errorSince = "\1.4", hiddenSince = "\1.5")\n@kotlin.jvm.JvmName("\longRangeContains")\npublic operator fun ClosedRange<Long>.contains(value: Double): Boolean {\n return value.toLongExactOrNull().let { if (it != null) contains(it) else false }\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@Deprecated("\This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.")\n@DeprecatedSinceKotlin(warningSince = "\1.3", errorSince = "\1.4", hiddenSince = "\1.5")\n@kotlin.jvm.JvmName("\byteRangeContains")\npublic operator fun ClosedRange<Byte>.contains(value: Double): Boolean {\n return value.toByteExactOrNull().let { if (it != null) contains(it) else false }\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@Deprecated("\This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.")\n@DeprecatedSinceKotlin(warningSince = "\1.3", errorSince = "\1.4", hiddenSince = "\1.5")\n@kotlin.jvm.JvmName("\shortRangeContains")\npublic operator fun ClosedRange<Short>.contains(value: Double): Boolean {\n return value.toShortExactOrNull().let { if (it != null) contains(it) else false }\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@kotlin.jvm.JvmName("\floatRangeContains")\npublic operator fun ClosedRange<Float>.contains(value: Double): Boolean {\n return contains(value.toFloat())\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@Deprecated("\This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.")\n@DeprecatedSinceKotlin(warningSince = "\1.3", errorSince = "\1.4", hiddenSince = "\1.5")\n@kotlin.jvm.JvmName("\intRangeContains")\npublic operator fun ClosedRange<Int>.contains(value: Float): Boolean {\n return value.toIntExactOrNull().let { if (it != null) contains(it) else false }\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@Deprecated("\This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.")\n@DeprecatedSinceKotlin(warningSince = "\1.3", errorSince = "\1.4", hiddenSince = "\1.5")\n@kotlin.jvm.JvmName("\longRangeContains")\npublic operator fun ClosedRange<Long>.contains(value: Float): Boolean {\n return value.toLongExactOrNull().let { if (it != null) contains(it) else false }\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@Deprecated("\This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.")\n@DeprecatedSinceKotlin(warningSince = "\1.3", errorSince = "\1.4", hiddenSince = "\1.5")\n@kotlin.jvm.JvmName("\byteRangeContains")\npublic operator fun ClosedRange<Byte>.contains(value: Float): Boolean {\n return value.toByteExactOrNull().let { if (it != null)


```

value is greater than `this` value the returned progression is empty.\n */\npublic infix fun Byte.downTo(to: Long):
LongProgression {\n    return LongProgression.fromClosedRange(this.toLong(), to, -1L)\n}\n\n/**\n * Returns a
progression from this value down to the specified [to] value with the step -1.\n * \n * The [to] value should be less
than or equal to `this` value.\n * If the [to] value is greater than `this` value the returned progression is empty.\n
*/\npublic infix fun Short.downTo(to: Long): LongProgression {\n    return
LongProgression.fromClosedRange(this.toLong(), to, -1L)\n}\n\n/**\n * Returns a progression from this value
down to the specified [to] value with the step -1.\n * \n * The [to] value should be less than or equal to `this` value.\n
* If the [to] value is greater than `this` value the returned progression is empty.\n */\npublic infix fun Int.downTo(to:
Short): IntProgression {\n    return IntProgression.fromClosedRange(this, to.toInt(), -1)\n}\n\n/**\n * Returns a
progression from this value down to the specified [to] value with the step -1.\n * \n * The [to] value should be less
than or equal to `this` value.\n * If the [to] value is greater than `this` value the returned progression is empty.\n
*/\npublic infix fun Long.downTo(to: Short): LongProgression {\n    return LongProgression.fromClosedRange(this,
to.toLong(), -1L)\n}\n\n/**\n * Returns a progression from this value down to the specified [to] value with the step -
1.\n * \n * The [to] value should be less than or equal to `this` value.\n * If the [to] value is greater than `this` value
the returned progression is empty.\n */\npublic infix fun Byte.downTo(to: Short): IntProgression {\n    return
IntProgression.fromClosedRange(this.toInt(), to.toInt(), -1)\n}\n\n/**\n * Returns a progression from this value
down to the specified [to] value with the step -1.\n * \n * The [to] value should be less than or equal to `this` value.\n
* If the [to] value is greater than `this` value the returned progression is empty.\n */\npublic infix fun
Short.downTo(to: Short): IntProgression {\n    return IntProgression.fromClosedRange(this.toInt(), to.toInt(), -
1)\n}\n\n/**\n * Returns a progression that goes over the same range in the opposite direction with the same step.\n
*/\npublic fun IntProgression.reversed(): IntProgression {\n    return IntProgression.fromClosedRange(last, first, -
step)\n}\n\n/**\n * Returns a progression that goes over the same range in the opposite direction with the same
step.\n */\npublic fun LongProgression.reversed(): LongProgression {\n    return
LongProgression.fromClosedRange(last, first, -step)\n}\n\n/**\n * Returns a progression that goes over the same
range in the opposite direction with the same step.\n */\npublic fun CharProgression.reversed(): CharProgression {\n
return CharProgression.fromClosedRange(last, first, -step)\n}\n\n/**\n * Returns a progression that goes over the
same range with the given step.\n */\npublic infix fun IntProgression.step(step: Int): IntProgression {\n
checkStepIsPositive(step > 0, step)\n    return IntProgression.fromClosedRange(first, last, if (this.step > 0) step else -
step)\n}\n\n/**\n * Returns a progression that goes over the same range with the given step.\n */\npublic infix fun
LongProgression.step(step: Long): LongProgression {\n    checkStepIsPositive(step > 0, step)\n    return
LongProgression.fromClosedRange(first, last, if (this.step > 0) step else -step)\n}\n\n/**\n * Returns a progression
that goes over the same range with the given step.\n */\npublic infix fun CharProgression.step(step: Int):
CharProgression {\n    checkStepIsPositive(step > 0, step)\n    return CharProgression.fromClosedRange(first, last, if
(this.step > 0) step else -step)\n}\n\ninternal fun Int.toByteExactOrNull(): Byte? {\n    return if (this in
Byte.MIN_VALUE.toInt()..Byte.MAX_VALUE.toInt()) this.toByte() else null\n}\n\ninternal fun
Long.toByteExactOrNull(): Byte? {\n    return if (this in
Byte.MIN_VALUE.toLong()..Byte.MAX_VALUE.toLong()) this.toByte() else null\n}\n\ninternal fun
Short.toByteExactOrNull(): Byte? {\n    return if (this in
Byte.MIN_VALUE.toShort()..Byte.MAX_VALUE.toShort()) this.toByte() else null\n}\n\ninternal fun
Double.toByteExactOrNull(): Byte? {\n    return if (this in
Byte.MIN_VALUE.toDouble()..Byte.MAX_VALUE.toDouble()) this.toInt().toByte() else null\n}\n\ninternal fun
Float.toByteExactOrNull(): Byte? {\n    return if (this in
Byte.MIN_VALUE.toFloat()..Byte.MAX_VALUE.toFloat()) this.toInt().toByte() else null\n}\n\ninternal fun
Long.toIntExactOrNull(): Int? {\n    return if (this in Int.MIN_VALUE.toLong()..Int.MAX_VALUE.toLong())
this.toInt() else null\n}\n\ninternal fun Double.toIntExactOrNull(): Int? {\n    return if (this in
Int.MIN_VALUE.toDouble()..Int.MAX_VALUE.toDouble()) this.toInt() else null\n}\n\ninternal fun
Float.toIntExactOrNull(): Int? {\n    return if (this in Int.MIN_VALUE.toFloat()..Int.MAX_VALUE.toFloat())
this.toInt() else null\n}\n\ninternal fun Double.toLongExactOrNull(): Long? {\n    return if (this in

```

Long.MIN_VALUE.toDouble()..Long.MAX_VALUE.toDouble()) this.toLong() else null\n}\n\ninternal fun
 Float.toLongExactOrNull(): Long? {\n return if (this in
 Long.MIN_VALUE.toFloat()..Long.MAX_VALUE.toFloat()) this.toLong() else null\n}\n\ninternal fun
 Int.toShortExactOrNull(): Short? {\n return if (this in Short.MIN_VALUE.toInt()..Short.MAX_VALUE.toInt())
 this.toShort() else null\n}\n\ninternal fun Long.toShortExactOrNull(): Short? {\n return if (this in
 Short.MIN_VALUE.toLong()..Short.MAX_VALUE.toLong()) this.toShort() else null\n}\n\ninternal fun
 Double.toShortExactOrNull(): Short? {\n return if (this in
 Short.MIN_VALUE.toDouble()..Short.MAX_VALUE.toDouble()) this.toInt().toShort() else null\n}\n\ninternal fun
 Float.toShortExactOrNull(): Short? {\n return if (this in
 Short.MIN_VALUE.toFloat()..Short.MAX_VALUE.toFloat()) this.toInt().toShort() else null\n}\n\n/**\n * Returns
 a range from this value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to
 `this` value, then the returned range is empty.\n */\npublic infix fun Int.until(to: Byte): IntRange {\n return this ..
 (to.toInt() - 1).toInt()\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n *
 \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n */\npublic infix fun
 Long.until(to: Byte): LongRange {\n return this .. (to.toLong() - 1).toLong()\n}\n\n/**\n * Returns a range from
 this value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this`
 value,
 then the returned range is empty.\n */\npublic infix fun Byte.until(to: Byte): IntRange {\n return this.toInt() ..
 (to.toInt() - 1).toInt()\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n *
 \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n */\npublic infix fun
 Short.until(to: Byte): IntRange {\n return this.toInt() .. (to.toInt() - 1).toInt()\n}\n\n/**\n * Returns a range from
 this value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this`
 value,
 then the returned range is empty.\n */\npublic infix fun Char.until(to: Char): CharRange {\n if (to <= '\u0000')
 return CharRange.EMPTY\n return this .. (to - 1).toChar()\n}\n\n/**\n * Returns a range from this value up to but
 excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned
 range
 is empty.\n */\npublic infix fun Int.until(to: Int): IntRange {\n if (to <= Int.MIN_VALUE) return
 IntRange.EMPTY\n return this .. (to - 1).toInt()\n}\n\n/**\n * Returns a range from this value up to but excluding
 the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned range is
 empty.\n */\npublic infix fun Long.until(to: Int): LongRange {\n return this .. (to.toLong() -
 1).toLong()\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the
 [to] value is less than or equal to `this` value, then the returned range is empty.\n */\npublic infix fun Byte.until(to:
 Int): IntRange {\n if (to <= Int.MIN_VALUE) return IntRange.EMPTY\n return this.toInt() .. (to -
 1).toInt()\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the
 [to] value is less than or equal to `this` value, then the returned range is empty.\n */\npublic infix fun Short.until(to:
 Int): IntRange {\n if (to <= Int.MIN_VALUE) return IntRange.EMPTY\n return this.toInt() .. (to -
 1).toInt()\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the
 [to] value is less than or equal to `this` value, then the returned range is empty.\n */\npublic infix fun Int.until(to:
 Long): LongRange {\n if (to <= Long.MIN_VALUE) return LongRange.EMPTY\n return this.toLong() .. (to -
 1).toLong()\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the
 [to] value is less than or equal to `this` value, then the returned range is empty.\n */\npublic infix fun Long.until(to:
 Long): LongRange {\n if (to <= Long.MIN_VALUE) return LongRange.EMPTY\n return this .. (to -
 1).toLong()\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the
 [to] value is less than or equal to `this` value, then the returned range is empty.\n */\npublic infix fun Byte.until(to:
 Long): LongRange {\n if (to <= Long.MIN_VALUE) return LongRange.EMPTY\n return this.toLong() .. (to -
 1).toLong()\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the
 [to] value is less than or equal to `this` value, then the returned range is empty.\n */\npublic infix fun Short.until(to:
 Long): LongRange {\n if (to <= Long.MIN_VALUE) return LongRange.EMPTY\n return this.toLong() .. (to -
 1).toLong()\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the
 [to] value is less than or equal to `this` value, then the returned range is empty.\n */\npublic infix fun Int.until(to:

Short): IntRange { \n return this .. (to.toInt() - 1).toInt()\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n */\npublic infix fun Long.until(to: Short): LongRange { \n return this .. (to.toLong() - 1).toLong()\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n */\npublic infix fun Byte.until(to: Short): IntRange { \n return this.toInt() .. (to.toInt() - 1).toInt()\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n */\npublic infix fun Short.until(to: Short): IntRange { \n return this.toInt() .. (to.toInt() - 1).toInt()\n}\n\n/**\n * Ensures that this value is not less than the specified [minimumValue].\n * \n * @return this value if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.\n * \n * @sample samples.comparisons.ComparableOps.coerceAtLeastComparable\n */\npublic fun <T : Comparable<T>> T.coerceAtLeast(minimumValue: T): T { \n return if (this < minimumValue) minimumValue else this\n}\n\n/**\n * Ensures that this value is not less than the specified [minimumValue].\n * \n * @return this value if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.\n * \n * @sample samples.comparisons.ComparableOps.coerceAtLeast\n */\npublic fun Byte.coerceAtLeast(minimumValue: Byte): Byte { \n return if (this < minimumValue) minimumValue else this\n}\n\n/**\n * Ensures that this value is not less than the specified [minimumValue].\n * \n * @return this value if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.\n * \n * @sample samples.comparisons.ComparableOps.coerceAtLeast\n */\npublic fun Short.coerceAtLeast(minimumValue: Short): Short { \n return if (this < minimumValue) minimumValue else this\n}\n\n/**\n * Ensures that this value is not less than the specified [minimumValue].\n * \n * @return this value if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.\n * \n * @sample samples.comparisons.ComparableOps.coerceAtLeast\n */\npublic fun Int.coerceAtLeast(minimumValue: Int): Int { \n return if (this < minimumValue) minimumValue else this\n}\n\n/**\n * Ensures that this value is not less than the specified [minimumValue].\n * \n * @return this value if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.\n * \n * @sample samples.comparisons.ComparableOps.coerceAtLeast\n */\npublic fun Long.coerceAtLeast(minimumValue: Long): Long { \n return if (this < minimumValue) minimumValue else this\n}\n\n/**\n * Ensures that this value is not less than the specified [minimumValue].\n * \n * @return this value if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.\n * \n * @sample samples.comparisons.ComparableOps.coerceAtLeast\n */\npublic fun Float.coerceAtLeast(minimumValue: Float): Float { \n return if (this < minimumValue) minimumValue else this\n}\n\n/**\n * Ensures that this value is not less than the specified [minimumValue].\n * \n * @return this value if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.\n * \n * @sample samples.comparisons.ComparableOps.coerceAtLeast\n */\npublic fun Double.coerceAtLeast(minimumValue: Double): Double { \n return if (this < minimumValue) minimumValue else this\n}\n\n/**\n * Ensures that this value is not greater than the specified [maximumValue].\n * \n * @return this value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise.\n * \n * @sample samples.comparisons.ComparableOps.coerceAtMostComparable\n */\npublic fun <T : Comparable<T>> T.coerceAtMost(maximumValue: T): T { \n return if (this > maximumValue) maximumValue else this\n}\n\n/**\n * Ensures that this value is not greater than the specified [maximumValue].\n * \n * @return this value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise.\n * \n * @sample samples.comparisons.ComparableOps.coerceAtMost\n */\npublic fun Byte.coerceAtMost(maximumValue: Byte): Byte { \n return if (this > maximumValue) maximumValue else this\n}\n\n/**\n * Ensures that this value is not greater than the specified [maximumValue].\n * \n * @return this value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise.\n * \n * @sample samples.comparisons.ComparableOps.coerceAtMost\n */\npublic fun Short.coerceAtMost(maximumValue: Short): Short { \n return if (this > maximumValue) maximumValue else this\n}\n\n/**\n * Ensures that this value is not greater than the specified [maximumValue].\n * \n * @return this value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise.\n * \n * @sample

```

samples.comparisons.ComparableOps.coerceAtMost\n */\npublic fun Int.coerceAtMost(maximumValue: Int): Int
{\n    return if (this > maximumValue) maximumValue else this\n}\n\n/**\n * Ensures that this value is not greater
than the specified [maximumValue].\n * \n * @return this value if it's less than or equal to the [maximumValue] or
the [maximumValue] otherwise.\n * \n * @sample samples.comparisons.ComparableOps.coerceAtMost\n
*/\npublic fun Long.coerceAtMost(maximumValue: Long): Long {\n    return if (this > maximumValue)
maximumValue else this\n}\n\n/**\n * Ensures that this value is not greater than the specified [maximumValue].\n
*\n * @return this value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise.\n
*\n * @sample samples.comparisons.ComparableOps.coerceAtMost\n */\npublic fun
Float.coerceAtMost(maximumValue: Float): Float {\n    return if (this > maximumValue) maximumValue else
this\n}\n\n/**\n * Ensures that this value is not greater than the specified [maximumValue].\n * \n * @return this
value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise.\n * \n * @sample
samples.comparisons.ComparableOps.coerceAtMost\n */\npublic fun Double.coerceAtMost(maximumValue:
Double): Double {\n    return if (this > maximumValue) maximumValue else this\n}\n\n/**\n * Ensures that this
value lies in the specified range [minimumValue]..[maximumValue].\n * \n * @return this value if it's in the range,
or [minimumValue] if this value is less than [minimumValue], or [maximumValue] if this value is greater than
[maximumValue].\n * \n * @sample samples.comparisons.ComparableOps.coerceInComparable\n */\npublic fun
<T : Comparable<T>> T.coerceIn(minimumValue: T?, maximumValue: T?): T {\n    if (minimumValue !== null
&& maximumValue !== null) {\n        if (minimumValue > maximumValue) throw
IllegalArgumentOutOfRangeException("Cannot coerce value to an empty range: maximum $maximumValue is less than
minimum $minimumValue.")\n        if (this < minimumValue) return minimumValue\n        if (this >
maximumValue) return maximumValue\n    }\n    else {\n        if (minimumValue !== null && this <
minimumValue) return minimumValue\n        if (maximumValue !== null && this > maximumValue) return
maximumValue\n    }\n    return this\n}\n\n/**\n * Ensures that this value lies in the specified range
[minimumValue]..[maximumValue].\n * \n * @return this value if it's in the range, or [minimumValue] if this value
is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue].\n * \n * @sample
samples.comparisons.ComparableOps.coerceIn\n */\npublic fun Byte.coerceIn(minimumValue: Byte,
maximumValue: Byte): Byte {\n    if (minimumValue > maximumValue) throw
IllegalArgumentOutOfRangeException("Cannot coerce value to an empty range: maximum $maximumValue is less than
minimum $minimumValue.")\n    if (this < minimumValue) return minimumValue\n    if (this > maximumValue)
return maximumValue\n    return this\n}\n\n/**\n * Ensures that this value lies in the specified range
[minimumValue]..[maximumValue].\n * \n * @return this value if it's in the range, or [minimumValue] if this value
is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue].\n * \n * @sample
samples.comparisons.ComparableOps.coerceIn\n */\npublic fun Short.coerceIn(minimumValue: Short,
maximumValue: Short): Short {\n    if (minimumValue > maximumValue) throw
IllegalArgumentOutOfRangeException("Cannot coerce value to an empty range: maximum $maximumValue is less than
minimum $minimumValue.")\n    if (this < minimumValue) return minimumValue\n    if (this > maximumValue)
return maximumValue\n    return this\n}\n\n/**\n * Ensures that this value lies in the specified range
[minimumValue]..[maximumValue].\n * \n * @return this value if it's in the range, or [minimumValue] if this value
is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue].\n * \n * @sample
samples.comparisons.ComparableOps.coerceIn\n */\npublic fun Int.coerceIn(minimumValue: Int, maximumValue:
Int): Int {\n    if (minimumValue > maximumValue) throw IllegalArgumentOutOfRangeException("Cannot coerce value to an
empty range: maximum $maximumValue is less than minimum $minimumValue.")\n    if (this < minimumValue)
return minimumValue\n    if (this > maximumValue) return maximumValue\n    return this\n}\n\n/**\n * Ensures
that this value lies in the specified range [minimumValue]..[maximumValue].\n * \n * @return this value if it's in
the range, or [minimumValue] if this value is less than [minimumValue], or [maximumValue] if this value is greater
than [maximumValue].\n * \n * @sample samples.comparisons.ComparableOps.coerceIn\n */\npublic fun
Long.coerceIn(minimumValue: Long, maximumValue: Long): Long {\n    if (minimumValue > maximumValue)
throw IllegalArgumentOutOfRangeException("Cannot coerce value to an empty range: maximum $maximumValue is less than

```

```

minimum $minimumValue.\\n if (this < minimumValue) return minimumValue\\n if (this > maximumValue)
return maximumValue\\n return this\\n}\\n\\n/**\\n * Ensures that this value lies in the specified range
[minimumValue]..[maximumValue].\\n * \\n * @return this value if it's in the range, or [minimumValue] if this value
is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue].\\n * \\n * @sample
samples.comparisons.ComparableOps.coerceIn\\n */\\npublic fun Float.coerceIn(minimumValue: Float,
maximumValue: Float): Float {\\n if (minimumValue > maximumValue) throw
IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than
minimum $minimumValue.\\n")\\n if (this < minimumValue) return minimumValue\\n if (this > maximumValue)
return maximumValue\\n return this\\n}\\n\\n/**\\n * Ensures that this value lies in the specified range
[minimumValue]..[maximumValue].\\n * \\n * @return this value if it's in the range, or [minimumValue] if this value
is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue].\\n * \\n * @sample
samples.comparisons.ComparableOps.coerceIn\\n */\\npublic fun Double.coerceIn(minimumValue: Double,
maximumValue: Double): Double {\\n if (minimumValue > maximumValue) throw
IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than
minimum $minimumValue.\\n")\\n if (this < minimumValue) return minimumValue\\n if (this > maximumValue)
return maximumValue\\n return this\\n}\\n\\n/**\\n * Ensures that this value lies in the specified [range].\\n * \\n *
@return this value if it's in the [range], or `range.start` if this value is less than `range.start`, or `range.endInclusive`
if this value is greater than `range.endInclusive`.\\n * \\n * @sample
samples.comparisons.ComparableOps.coerceInFloatingPointRange\\n */\\n@SinceKotlin("1.1")\\npublic fun <T :
Comparable<T>> T.coerceIn(range: ClosedFloatingPointRange<T>): T {\\n if (range.isEmpty()) throw
IllegalArgumentException("Cannot coerce value to an empty range: $range.\\n")\\n return when {\\n // this <
start equiv to this <= start && !(this >= start)\\n range.lessThanOrEqualTo(this, range.start) &&
!range.lessThanOrEqualTo(range.start, this) -> range.start\\n // this > end equiv to this >= end && !(this <= end)\\n
range.lessThanOrEqualTo(range.endInclusive, this) && !range.lessThanOrEqualTo(this, range.endInclusive) ->
range.endInclusive\\n else -> this\\n }\\n}\\n\\n/**\\n * Ensures that this value lies in the specified [range].\\n * \\n *
@return this value if it's in the [range], or `range.start` if this value is less than `range.start`, or `range.endInclusive`
if this value is greater than `range.endInclusive`.\\n * \\n * @sample
samples.comparisons.ComparableOps.coerceInComparable\\n */\\npublic fun <T : Comparable<T>>
T.coerceIn(range: ClosedRange<T>): T {\\n if (range is ClosedFloatingPointRange) {\\n return
this.coerceIn<T>(range)\\n }\\n if (range.isEmpty()) throw IllegalArgumentException("Cannot coerce value to an
empty range: $range.\\n")\\n return when {\\n this < range.start -> range.start\\n this > range.endInclusive ->
range.endInclusive\\n else -> this\\n }\\n}\\n\\n/**\\n * Ensures that this value lies in the specified [range].\\n * \\n *
@return this value if it's in the [range], or `range.start` if this value is less than `range.start`, or `range.endInclusive`
if this value is greater than `range.endInclusive`.\\n * \\n * @sample
samples.comparisons.ComparableOps.coerceIn\\n */\\npublic fun Int.coerceIn(range: ClosedRange<Int>): Int {\\n if (range is ClosedFloatingPointRange) {\\n
return this.coerceIn<Int>(range)\\n }\\n if (range.isEmpty()) throw IllegalArgumentException("Cannot coerce
value to an empty range: $range.\\n")\\n return when {\\n this < range.start -> range.start\\n this >
range.endInclusive -> range.endInclusive\\n else -> this\\n }\\n}\\n\\n/**\\n * Ensures that this value lies in the
specified [range].\\n * \\n * @return this value if it's in the [range], or `range.start` if this value is less than
`range.start`, or `range.endInclusive` if this value is greater than `range.endInclusive`.\\n * \\n * @sample
samples.comparisons.ComparableOps.coerceIn\\n */\\npublic fun Long.coerceIn(range: ClosedRange<Long>): Long
{\\n if (range is ClosedFloatingPointRange) {\\n return this.coerceIn<Long>(range)\\n }\\n if
(range.isEmpty()) throw IllegalArgumentException("Cannot coerce value to an empty range: $range.\\n")\\n return
when {\\n this < range.start -> range.start\\n this > range.endInclusive -> range.endInclusive\\n else ->
this\\n }\\n}\\n}\\n\\n/**\\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\\n *
Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\\n
*/\\n\\n// Auto-generated file. DO NOT EDIT!\\n\\npackage kotlin\\n\\nimport kotlin.experimental.*\\nimport
kotlin.jvm.*\\n\\n@SinceKotlin("1.5")\\n@WasExperimental(ExperimentalUnsignedTypes::class)\\n@JvmInline\\npu

```

```

blic value class UByte @PublishedApi internal constructor(@PublishedApi internal val data: Byte) :
Comparable<UByte> {\n\n companion object {\n /**\n * A constant holding the minimum value an
instance of UByte can have.\n */\n public const val MIN_VALUE: UByte = UByte(0)\n /**\n *
A constant holding the maximum value an instance of UByte can have.\n */\n public const val
MAX_VALUE: UByte = UByte(-1)\n /**\n * The number of bytes used to represent an instance of
UByte in a binary form.\n */\n public const val SIZE_BYTES: Int = 1\n /**\n * The number of
bits used to represent an instance of UByte in a binary form.\n */\n public const val SIZE_BITS: Int = 8\n
}\n\n /**\n * Compares this value with the specified value for order.\n * Returns zero if this value is equal to
the specified other value, a negative number if it's less than other,\n * or a positive number if it's greater than
other.\n */\n @kotlin.internal.InlineOnly\n @Suppress("OVERRIDE_BY_INLINE")\n public override
inline operator fun compareTo(other: UByte): Int = this.toInt().compareTo(other.toInt())\n /**\n * Compares
this value with the specified value for order.\n * Returns zero if this value is equal to the specified other value, a
negative number if it's less than other,\n * or a positive number if it's greater than other.\n */\n
@kotlin.internal.InlineOnly\n public inline operator fun compareTo(other: UShort): Int =
this.toInt().compareTo(other.toInt())\n /**\n * Compares this value with the specified value for order.\n *
Returns zero if this value is equal to the specified other value, a negative number if it's less than other,\n * or a
positive number if it's greater than other.\n */\n @kotlin.internal.InlineOnly\n public inline operator fun
compareTo(other: UInt): Int = this.toUInt().compareTo(other)\n /**\n * Compares this value with the
specified value for order.\n * Returns zero if this value is equal to the specified other value, a negative number if
it's less than other,\n * or a positive number if it's greater than other.\n */\n @kotlin.internal.InlineOnly\n
public inline operator fun compareTo(other: ULong): Int = this.toULong().compareTo(other)\n /** Adds the
other value to this value. */\n @kotlin.internal.InlineOnly\n public inline operator fun plus(other: UByte): UInt =
this.toUInt().plus(other.toUInt())\n /** Adds the other value to this value. */\n @kotlin.internal.InlineOnly\n
public inline operator fun plus(other: UShort): UInt = this.toUInt().plus(other.toUInt())\n /** Adds the other value
to this value. */\n @kotlin.internal.InlineOnly\n public inline operator fun plus(other: UInt): UInt =
this.toUInt().plus(other)\n /** Adds the other value to this value. */\n @kotlin.internal.InlineOnly\n public
inline operator fun plus(other: ULong): ULong = this.toULong().plus(other)\n /** Subtracts the other value from
this value. */\n @kotlin.internal.InlineOnly\n public inline operator fun minus(other: UByte): UInt =
this.toUInt().minus(other.toUInt())\n /** Subtracts the other value from this value. */\n
@kotlin.internal.InlineOnly\n public inline operator fun minus(other: UShort): UInt =
this.toUInt().minus(other.toUInt())\n /** Subtracts the other value from this value. */\n
@kotlin.internal.InlineOnly\n public inline operator fun minus(other: UInt): UInt = this.toUInt().minus(other)\n
/** Subtracts the other value from this value. */\n @kotlin.internal.InlineOnly\n public inline operator fun
minus(other: ULong): ULong = this.toULong().minus(other)\n /** Multiplies this value by the other value. */\n
@kotlin.internal.InlineOnly\n public inline operator fun times(other: UByte): UInt =
this.toUInt().times(other.toUInt())\n /** Multiplies this value by the other value. */\n
@kotlin.internal.InlineOnly\n public inline operator fun times(other: UShort): UInt =
this.toUInt().times(other.toUInt())\n /** Multiplies this value by the other value. */\n
@kotlin.internal.InlineOnly\n public inline operator fun times(other: UInt): UInt = this.toUInt().times(other)\n
/** Multiplies this value by the other value. */\n @kotlin.internal.InlineOnly\n public inline operator fun
times(other: ULong): ULong = this.toULong().times(other)\n /** Divides this value by the other value,
truncating the result to an integer that is closer to zero. */\n @kotlin.internal.InlineOnly\n public inline operator
fun div(other: UByte): UInt = this.toUInt().div(other.toUInt())\n /** Divides this value by the other value,
truncating the result to an integer that is closer to zero. */\n @kotlin.internal.InlineOnly\n public inline operator
fun div(other: UShort): UInt = this.toUInt().div(other.toUInt())\n /** Divides this value by the other value,
truncating the result to an integer that is closer to zero. */\n @kotlin.internal.InlineOnly\n public inline operator
fun div(other: UInt): UInt = this.toUInt().div(other)\n /** Divides this value by the other value, truncating the
result to an integer that is closer to zero. */\n @kotlin.internal.InlineOnly\n public inline operator fun div(other:

```

```

ULong): ULong = this.toULong().div(other)\n\n /**\n * Calculates the remainder of truncating division of this
value by the other value.\n * \n * The result is always less than the divisor.\n */\n
@kotlin.internal.InlineOnly\n public inline operator fun rem(other: UByte): UInt =
this.toUInt().rem(other.toUInt())\n /**\n * Calculates the remainder of truncating division of this value by the
other value.\n * \n * The result is always less than the divisor.\n */\n @kotlin.internal.InlineOnly\n public
inline operator fun rem(other: UShort): UInt = this.toUInt().rem(other.toUInt())\n /**\n * Calculates the
remainder of truncating division of this value by the other value.\n * \n * The result is always less than the
divisor.\n */\n @kotlin.internal.InlineOnly\n public inline operator fun rem(other: UInt): UInt =
this.toUInt().rem(other)\n /**\n * Calculates the remainder of truncating division of this value by the other
value.\n * \n * The result is always less than the divisor.\n */\n @kotlin.internal.InlineOnly\n public
inline operator fun rem(other: ULong): ULong = this.toULong().rem(other)\n\n /**\n * Divides this value by
the other value, flooring the result to an integer that is closer to negative infinity.\n * \n * For unsigned
types, the results of flooring division and truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n
public inline fun floorDiv(other: UByte): UInt = this.toUInt().floorDiv(other.toUInt())\n /**\n * Divides this
value by the other value, flooring the result to an integer that is closer to negative infinity.\n * \n * For unsigned
types, the results of flooring division and truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n
public inline fun floorDiv(other: UShort): UInt = this.toUInt().floorDiv(other.toUInt())\n /**\n * Divides this
value by the other value, flooring the result to an integer that is closer to negative infinity.\n * \n * For unsigned
types, the results of flooring division and truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n
public inline fun floorDiv(other: UInt): UInt = this.toUInt().floorDiv(other)\n /**\n * Divides this value by the
other value, flooring the result to an integer that is closer to negative infinity.\n * \n * For unsigned types,
the results of flooring division and truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n public
inline fun floorDiv(other: ULong): ULong = this.toULong().floorDiv(other)\n\n /**\n * Calculates the
remainder of flooring division of this value by the other value.\n * \n * The result is always less than the
divisor.\n * \n * For unsigned types, the remainders of flooring division and truncating division are the same.\n
*/\n @kotlin.internal.InlineOnly\n public inline fun mod(other: UByte): UByte =
this.toUInt().mod(other.toUInt()).toUByte()\n /**\n * Calculates the remainder of flooring division of this value
by the other value.\n * \n * The result is always less than the divisor.\n * \n * For unsigned types, the
remainders of flooring division and truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n
public inline fun mod(other: UShort): UShort = this.toUInt().mod(other.toUInt()).toUShort()\n /**\n *
Calculates the remainder of flooring division of this value by the other value.\n * \n * The result is always less
than the divisor.\n * \n * For unsigned types, the remainders of flooring division and truncating division are the
same.\n */\n @kotlin.internal.InlineOnly\n public inline fun mod(other: UInt): UInt =
this.toUInt().mod(other)\n /**\n * Calculates the remainder of flooring division of this value by the other
value.\n * \n * The result is always less than the divisor.\n * \n * For unsigned types, the remainders of
flooring division and truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n public inline fun
mod(other: ULong): ULong = this.toULong().mod(other)\n\n /**\n * Returns this value incremented by one.\n
*/\n * @sample samples.misc.Builtins.inc\n */\n @kotlin.internal.InlineOnly\n public inline operator fun
inc(): UByte = UByte(data.inc())\n\n /**\n * Returns this value decremented by one.\n */\n * @sample
samples.misc.Builtins.dec\n */\n @kotlin.internal.InlineOnly\n public inline operator fun dec(): UByte =
UByte(data.dec())\n\n /**\n * Creates a range from this value to the specified [other] value. */\n
@kotlin.internal.InlineOnly\n public inline operator fun rangeTo(other: UByte): UIntRange =
UIntRange(this.toUInt(), other.toUInt())\n\n /**\n * Performs a bitwise AND operation between the two values. */\n
@kotlin.internal.InlineOnly\n public inline infix fun and(other: UByte): UByte = UByte(this.data and other.data)\n\n
/**\n * Performs a bitwise OR operation between the two values. */\n @kotlin.internal.InlineOnly\n public inline
infix fun or(other: UByte): UByte = UByte(this.data or other.data)\n\n /**\n * Performs a bitwise XOR operation
between the two values. */\n @kotlin.internal.InlineOnly\n public inline infix fun xor(other: UByte): UByte =
UByte(this.data xor other.data)\n\n /**\n * Inverts the bits in this value. */\n @kotlin.internal.InlineOnly\n public

```

```

inline fun inv(): UByte = UByte(data.inv())\n\n /**\n * Converts this [UByte] value to [Byte].\n *\n * If this value is less than or equals to [Byte.MAX_VALUE], the resulting `Byte` value represents\n * the same numerical value as this `UByte`. Otherwise the result is negative.\n *\n * The resulting `Byte` value has the same binary representation as this `UByte` value.\n */\n @kotlin.internal.InlineOnly\n public inline fun toByte(): Byte = data\n /**\n * Converts this [UByte] value to [Short].\n *\n * The resulting `Short` value represents the same numerical value as this `UByte`.\n *\n * The least significant 8 bits of the resulting `Short` value are the same as the bits of this `UByte` value,\n * whereas the most significant 8 bits are filled with zeros.\n */\n @kotlin.internal.InlineOnly\n public inline fun toShort(): Short = data.toShort() and 0xFF\n /**\n * Converts this [UByte] value to [Int].\n *\n * The resulting `Int` value represents the same numerical value as this `UByte`.\n *\n * The least significant 8 bits of the resulting `Int` value are the same as the bits of this `UByte` value,\n * whereas the most significant 24 bits are filled with zeros.\n */\n @kotlin.internal.InlineOnly\n public inline fun toInt(): Int = data.toInt() and 0xFF\n /**\n * Converts this [UByte] value to [Long].\n *\n * The resulting `Long` value represents the same numerical value as this `UByte`.\n *\n * The least significant 8 bits of the resulting `Long` value are the same as the bits of this `UByte` value,\n * whereas the most significant 56 bits are filled with zeros.\n */\n @kotlin.internal.InlineOnly\n public inline fun toLong(): Long = data.toLong() and 0xFF\n\n /** Returns this value. */\n @kotlin.internal.InlineOnly\n public inline fun toUByte(): UByte = this\n /**\n * Converts this [UByte] value to [UShort].\n *\n * The resulting `UShort` value represents the same numerical value as this `UByte`.\n *\n * The least significant 8 bits of the resulting `UShort` value are the same as the bits of this `UByte` value,\n * whereas the most significant 8 bits are filled with zeros.\n */\n @kotlin.internal.InlineOnly\n public inline fun toUShort(): UShort = UShort(data.toShort() and 0xFF)\n /**\n * Converts this [UByte] value to [UInt].\n *\n * The resulting `UInt` value represents the same numerical value as this `UByte`.\n *\n * The least significant 8 bits of the resulting `UInt` value are the same as the bits of this `UByte` value,\n * whereas the most significant 24 bits are filled with zeros.\n */\n @kotlin.internal.InlineOnly\n public inline fun toUInt(): UInt = UInt(data.toInt() and 0xFF)\n /**\n * Converts this [UByte] value to [ULong].\n *\n * The resulting `ULong` value represents the same numerical value as this `UByte`.\n *\n * The least significant 8 bits of the resulting `ULong` value are the same as the bits of this `UByte` value,\n * whereas the most significant 56 bits are filled with zeros.\n */\n @kotlin.internal.InlineOnly\n public inline fun toULong(): ULong = ULong(data.toLong() and 0xFF)\n\n /**\n * Converts this [UByte] value to [Float].\n */\n @kotlin.internal.InlineOnly\n public inline fun toFloat(): Float = this.toInt().toFloat()\n /**\n * Converts this [UByte] value to [Double].\n */\n @kotlin.internal.InlineOnly\n public inline fun toDouble(): Double = this.toInt().toDouble()\n\n public override fun toString(): String = toInt().toString()\n\n\n /**\n * Converts this [Byte] value to [UByte].\n *\n * If this value is positive, the resulting `UByte` value represents the same numerical value as this `Byte`.\n *\n * The resulting `UByte` value has the same binary representation as this `Byte` value.\n */\n @SinceKotlin("1.5")\n @WasExperimental(ExperimentalUnsignedTypes::class)\n @kotlin.internal.InlineOnly\n public inline fun Byte.toUByte(): UByte = UByte(this)\n /**\n * Converts this [Short] value to [UByte].\n *\n * If this value is positive and less than or equals to [UByte.MAX_VALUE], the resulting `UByte` value represents\n * the same numerical value as this `Short`.\n *\n * The resulting `UByte` value is represented by the least significant 8 bits of this `Short` value.\n */\n @SinceKotlin("1.5")\n @WasExperimental(ExperimentalUnsignedTypes::class)\n @kotlin.internal.InlineOnly\n public inline fun Short.toUByte(): UByte = UByte(this.toByte())\n /**\n * Converts this [Int] value to [UByte].\n *\n * If this value is positive and less than or equals to [UByte.MAX_VALUE], the resulting `UByte` value represents\n * the same numerical value as this `Int`.\n *\n * The resulting `UByte` value is represented by the least significant 8 bits of this `Int` value.\n */\n @SinceKotlin("1.5")\n @WasExperimental(ExperimentalUnsignedTypes::class)\n @kotlin.internal.InlineOnly

```

```

public inline fun Int.toUByte(): UByte = UByte(this.toByte())\n/**\n * Converts this [Long] value to [UByte].\n *\n * If this value is positive and less than or equals to [UByte.MAX_VALUE], the resulting `UByte` value\n represents\n * the same numerical value as this `Long`.\n *\n * The resulting `UByte` value is represented by the\n least significant 8 bits of this `Long` value.\n\n*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun Long.toUByte(): UByte = UByte(this.toByte())\n"/*\n * Copyright 2010-2022 JetBrains s.r.o.\n and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license\n that can be found in the license/LICENSE.txt file.\n */\n\n// Auto-generated file. DO NOT EDIT!\n\npackage\nkotlin\n\nimport kotlin.experimental.*\nimport\nkotlin.jvm.*\n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@JvmInline\npublic value class UInt @PublishedApi internal constructor(@PublishedApi internal val data: Int) :\n Comparable<UInt> {\n\n    companion object {\n\n        /**\n         * A constant holding the minimum value an\n         instance of UInt can have.\n         *\n         public const val MIN_VALUE: UInt = UInt(0)\n\n        /**\n         * A\n         constant holding the maximum value an instance of UInt can have.\n         *\n         public const val MAX_VALUE:\n UInt = UInt(-1)\n\n        /**\n         * The number of bytes used to represent an instance of UInt in a binary form.\n         *\n         public const val SIZE_BYTES: Int = 4\n\n        /**\n         * The number of bits used to represent an\n         instance of UInt in a binary form.\n         *\n         public const val SIZE_BITS: Int = 32\n    }\n\n    /**\n     *\n     * Compares this value with the specified value for order.\n     * Returns zero if this value is equal to the specified other\n     value, a negative number if it's less than other,\n     * or a positive number if it's greater than other.\n     *\n     @kotlin.internal.InlineOnly\n     public inline operator fun compareTo(other: UByte): Int =\n     this.compareTo(other.toUInt())\n\n     /**\n      *\n      * Compares this value with the specified value for order.\n      *\n      * Returns zero if this value is equal to the specified other value, a negative number if it's less than other,\n      * or a\n      positive number if it's greater than other.\n      *\n      @kotlin.internal.InlineOnly\n      public inline operator fun\n      compareTo(other: UShort): Int = this.compareTo(other.toUInt())\n\n     /**\n      *\n      * Compares this value with the\n      specified value for order.\n      * Returns zero if this value is equal to the specified other value, a negative number if\n      it's less than other,\n      * or a positive number if it's greater than other.\n      *\n      @kotlin.internal.InlineOnly\n      @Suppress("OVERRIDE_BY_INLINE")\n      public override inline operator fun compareTo(other: UInt): Int =\n      uintCompare(this.data, other.data)\n\n     /**\n      *\n      * Compares this value with the specified value for order.\n      *\n      * Returns zero if this value is equal to the specified other value, a negative number if it's less than other,\n      * or a\n      positive number if it's greater than other.\n      *\n      @kotlin.internal.InlineOnly\n      public inline operator fun\n      compareTo(other: ULong): Int = this.toULong().compareTo(other)\n\n     /**\n      *\n      * Adds the other value to this value.\n      *\n      @kotlin.internal.InlineOnly\n      public inline operator fun plus(other: UByte): UInt = this.plus(other.toUInt())\n\n     /**\n      *\n      * Adds the other value to this value.\n      *\n      @kotlin.internal.InlineOnly\n      public inline operator fun plus(other:\n      UShort): UInt = this.plus(other.toUInt())\n\n     /**\n      *\n      * Adds the other value to this value.\n      *\n      @kotlin.internal.InlineOnly\n      public inline operator fun plus(other: UInt): UInt = UInt(this.data.plus(other.data))\n\n     /**\n      *\n      * Adds the other value to this value.\n      *\n      @kotlin.internal.InlineOnly\n      public inline operator fun plus(other:\n      ULong): ULong = this.toULong().plus(other)\n\n     /**\n      *\n      * Subtracts the other value from this value.\n      *\n      @kotlin.internal.InlineOnly\n      public inline operator fun minus(other: UByte): UInt = this.minus(other.toUInt())\n\n     /**\n      *\n      * Subtracts the other value from this value.\n      *\n      @kotlin.internal.InlineOnly\n      public inline operator fun\n      minus(other: UShort): UInt = this.minus(other.toUInt())\n\n     /**\n      *\n      * Subtracts the other value from this value.\n      *\n      @kotlin.internal.InlineOnly\n      public inline operator fun minus(other: UInt): UInt =\n      UInt(this.data.minus(other.data))\n\n     /**\n      *\n      * Subtracts the other value from this value.\n      *\n      @kotlin.internal.InlineOnly\n      public inline operator fun minus(other: ULong): ULong =\n      this.toULong().minus(other)\n\n     /**\n      *\n      * Multiplies this value by the other value.\n      *\n      @kotlin.internal.InlineOnly\n      public inline operator fun times(other: UByte): UInt = this.times(other.toUInt())\n\n     /**\n      *\n      * Multiplies this value by the\n      other value.\n      *\n      @kotlin.internal.InlineOnly\n      public inline operator fun times(other: UShort): UInt =\n      this.times(other.toUInt())\n\n     /**\n      *\n      * Multiplies this value by the other value.\n      *\n      @kotlin.internal.InlineOnly\n      public inline operator fun times(other: UInt): UInt = UInt(this.data.times(other.data))\n\n     /**\n      *\n      * Multiplies this value

```

```

by the other value. */
@kotlin.internal.InlineOnly
public inline operator fun times(other: ULong): ULong =
this.toULong().times(other)
/** Divides this value by the other value, truncating the result to an integer that is
closer to zero. */
@kotlin.internal.InlineOnly
public inline operator fun div(other: UByte): UInt =
this.div(other.toUInt())
/** Divides this value by the other value, truncating the result to an integer that is closer
to zero. */
@kotlin.internal.InlineOnly
public inline operator fun div(other: UShort): UInt =
this.div(other.toUInt())
/** Divides this value by the other value, truncating the result to an integer that is closer
to zero. */
@kotlin.internal.InlineOnly
public inline operator fun div(other: UInt): UInt = uintDivide(this,
other)
/** Divides this value by the other value, truncating the result to an integer that is closer to zero. */
@kotlin.internal.InlineOnly
public inline operator fun div(other: ULong): ULong =
this.toULong().div(other)
/**
 * Calculates the remainder of truncating division of this value by the other
value.
 * The result is always less than the divisor.
 */
@kotlin.internal.InlineOnly
public
inline operator fun rem(other: UByte): UInt = this.rem(other.toUInt())
/**
 * Calculates the remainder of
truncating division of this value by the other value.
 * The result is always less than the divisor.
 */
@kotlin.internal.InlineOnly
public inline operator fun rem(other: UShort): UInt = this.rem(other.toUInt())
/**
 * Calculates the remainder of truncating division of this value by the other value.
 * The result is
always less than the divisor.
 */
@kotlin.internal.InlineOnly
public inline operator fun rem(other: UInt):
UInt = uintRemainder(this, other)
/**
 * Calculates the remainder of truncating division of this value by the
other value.
 * The result is always less than the divisor.
 */
@kotlin.internal.InlineOnly
public
inline operator fun rem(other: ULong): ULong = this.toULong().rem(other)
/**
 * Divides this value by
the other value, flooring the result to an integer that is closer to negative infinity.
 * For unsigned types,
the results of flooring division and truncating division are the same.
 */
@kotlin.internal.InlineOnly
public
inline fun floorDiv(other: UByte): UInt = this.floorDiv(other.toUInt())
/**
 * Divides this value by the
other value, flooring the result to an integer that is closer to negative infinity.
 * For unsigned types, the
results of flooring division and truncating division are the same.
 */
@kotlin.internal.InlineOnly
public
inline fun floorDiv(other: UShort): UInt = this.floorDiv(other.toUInt())
/**
 * Divides this value by the other
value, flooring the result to an integer that is closer to negative infinity.
 * For unsigned types, the results
of flooring division and truncating division are the same.
 */
@kotlin.internal.InlineOnly
public inline
fun floorDiv(other: UInt): UInt = div(other)
/**
 * Divides this value by the other value, flooring the result to
an integer that is closer to negative infinity.
 * For unsigned types, the results of flooring division and
truncating division are the same.
 */
@kotlin.internal.InlineOnly
public inline fun floorDiv(other:
ULong): ULong = this.toULong().floorDiv(other)
/**
 * Calculates the remainder of flooring division of
this value by the other value.
 * The result is always less than the divisor.
 * For unsigned
types, the remainders of flooring division and truncating division are the same.
 */
@kotlin.internal.InlineOnly
public inline fun mod(other: UByte): UByte = this.mod(other.toUInt()).toUByte()
/**
 * Calculates the remainder of flooring division of this value by the other value.
 * The result is
always less than the divisor.
 * For unsigned types, the remainders of flooring division and truncating
division are the same.
 */
@kotlin.internal.InlineOnly
public inline fun mod(other: UShort): UShort =
this.mod(other.toUInt()).toUShort()
/**
 * Calculates the remainder of flooring division of this value by the
other value.
 * The result is always less than the divisor.
 * For unsigned types, the remainders
of flooring division and truncating division are the same.
 */
@kotlin.internal.InlineOnly
public inline
fun mod(other: UInt): UInt = rem(other)
/**
 * Calculates the remainder of flooring division of this value by
the other value.
 * The result is always less than the divisor.
 * For unsigned types, the
remainders of flooring division and truncating division are the same.
 */
@kotlin.internal.InlineOnly
public
inline fun mod(other: ULong): ULong = this.toULong().mod(other)
/**
 * Returns this value
incremented by one.
 */
@sample samples.misc.Builtins.inc
@kotlin.internal.InlineOnly
public inline operator fun inc(): UInt = UInt(data.inc())
/**
 * Returns this value decremented by one.
 */
@sample samples.misc.Builtins.dec
@kotlin.internal.InlineOnly
public inline operator fun
dec(): UInt = UInt(data.dec())
/** Creates a range from this value to the specified [other] value.
 */

```



```

*\n * If this value is positive, the resulting `UInt` value represents the same numerical value as this `Byte`. \n *\n *
The least significant 8 bits of the resulting `UInt` value are the same as the bits of this `Byte` value, \n * whereas the
most significant 24 bits are filled with the sign bit of this value. \n
*\n @SinceKotlin("1.5") \n @WasExperimental(ExperimentalUnsignedTypes::class) \n @kotlin.internal.InlineOnly \n
public inline fun Byte.toUInt(): UInt = UInt(this.toInt()) \n /** \n * Converts this [Short] value to [UInt]. \n *\n * If
this value is positive, the resulting `UInt` value represents the same numerical value as this `Short`. \n *\n * The least
significant 16 bits of the resulting `UInt` value are the same as the bits of this `Short` value, \n * whereas the most
significant 16 bits are filled with the sign bit of this value. \n
*\n @SinceKotlin("1.5") \n @WasExperimental(ExperimentalUnsignedTypes::class) \n @kotlin.internal.InlineOnly \n
public inline fun Short.toUInt(): UInt = UInt(this.toInt()) \n /** \n * Converts this [Int] value to [UInt]. \n *\n * If this
value is positive, the resulting `UInt` value represents the same numerical value as this `Int`. \n *\n * The resulting
`UInt` value has the same binary representation as this `Int` value. \n
*\n @SinceKotlin("1.5") \n @WasExperimental(ExperimentalUnsignedTypes::class) \n @kotlin.internal.InlineOnly \n
public inline fun Int.toUInt(): UInt = UInt(this) \n /** \n * Converts this [Long] value to [UInt]. \n *\n * If this value
is positive and less than or equals to [UInt.MAX_VALUE], the resulting `UInt` value represents \n * the same
numerical value as this `Long`. \n *\n * The resulting `UInt` value is represented by the least significant 32 bits of
this `Long` value. \n
*\n @SinceKotlin("1.5") \n @WasExperimental(ExperimentalUnsignedTypes::class) \n @kotlin.internal.InlineOnly \n
public inline fun Long.toUInt(): UInt = UInt(this.toInt()) \n /** \n * Converts this [Float] value to [UInt]. \n *\n *
The fractional part, if any, is rounded down towards zero. \n * Returns zero if this `Float` value is negative or `NaN`,
[UInt.MAX_VALUE] if it's bigger than `UInt.MAX_VALUE`. \n
*\n @SinceKotlin("1.5") \n @WasExperimental(ExperimentalUnsignedTypes::class) \n @kotlin.internal.InlineOnly \n
public inline fun Float.toUInt(): UInt = doubleToUInt(this.toDouble()) \n /** \n * Converts this [Double] value to
[UInt]. \n *\n * The fractional part, if any, is rounded down towards zero. \n * Returns zero if this `Double` value is
negative or `NaN`, [UInt.MAX_VALUE] if it's bigger than `UInt.MAX_VALUE`. \n
*\n @SinceKotlin("1.5") \n @WasExperimental(ExperimentalUnsignedTypes::class) \n @kotlin.internal.InlineOnly \n
public inline fun Double.toUInt(): UInt = doubleToUInt(this) \n /** \n * Copyright 2010-2022 JetBrains s.r.o. and
Kotlin Programming Language contributors. \n * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file. \n */ \n // Auto-generated file. DO NOT EDIT! \n \n package
kotlin \n \n import kotlin.experimental.* \n import
kotlin.jvm.* \n \n @SinceKotlin("1.5") \n @WasExperimental(ExperimentalUnsignedTypes::class) \n @JvmInline \n \n pu
blic value class UInt @PublishedApi internal constructor(@PublishedApi internal val data: Short) :
Comparable<UInt> { \n \n     companion object { \n \n         /** \n          * A constant holding the minimum value an
instance of UInt can have. \n          */ \n         public const val MIN_VALUE: UInt = UInt(0) \n \n         /** \n
          * A constant holding the maximum value an instance of UInt can have. \n          */ \n         public const val
MAX_VALUE: UInt = UInt(-1) \n \n         /** \n          * The number of bytes used to represent an instance of
UInt in a binary form. \n          */ \n         public const val SIZE_BYTES: Int = 2 \n \n         /** \n          * The number of
bits used to represent an instance of UInt in a binary form. \n          */ \n         public const val SIZE_BITS: Int =
16 \n     } \n \n     /** \n      * Compares this value with the specified value for order. \n      * Returns zero if this value is
equal to the specified other value, a negative number if it's less than other, \n      * or a positive number if it's greater
than other. \n      */ \n     @kotlin.internal.InlineOnly \n     public inline operator fun compareTo(other: UInt): Int =
this.toInt().compareTo(other.toInt()) \n \n     /** \n      * Compares this value with the specified value for order. \n
      * Returns zero if this value is equal to the specified other value, a negative number if it's less than other, \n
      * or a positive number if it's greater than other. \n      */ \n     @kotlin.internal.InlineOnly \n
     @Suppress("OVERRIDE_BY_INLINE") \n     public override inline operator fun compareTo(other: UInt): Int =
this.toInt().compareTo(other.toInt()) \n \n     /** \n      * Compares this value with the specified value for order. \n
      * Returns zero if this value is equal to the specified other value, a negative number if it's less than other, \n
      * or a positive number if it's greater than other. \n      */ \n     @kotlin.internal.InlineOnly \n     public inline operator fun

```

```

compareTo(other: UInt): Int = this.toUInt().compareTo(other)\n\n /**\n * Compares this value with the
specified value for order.\n * Returns zero if this value is equal to the specified other value, a negative number if
it's less than other,\n * or a positive number if it's greater than other.\n */\n @kotlin.internal.InlineOnly\n
public inline operator fun compareTo(other: ULong): Int = this.toULong().compareTo(other)\n\n /** Adds the
other value to this value. *\n @kotlin.internal.InlineOnly\n
public inline operator fun plus(other: UByte): UInt =
this.toUInt().plus(other.toUInt())\n\n /** Adds the other value to this value. *\n @kotlin.internal.InlineOnly\n
public inline operator fun plus(other: UShort): UInt = this.toUInt().plus(other.toUInt())\n\n /** Adds the other value
to this value. *\n @kotlin.internal.InlineOnly\n
public inline operator fun plus(other: UInt): UInt =
this.toUInt().plus(other)\n\n /** Adds the other value to this value. *\n @kotlin.internal.InlineOnly\n
public
inline operator fun plus(other: ULong): ULong = this.toULong().plus(other)\n\n /** Subtracts the other value from
this value. *\n @kotlin.internal.InlineOnly\n
public inline operator fun minus(other: UByte): UInt =
this.toUInt().minus(other.toUInt())\n\n /** Subtracts the other value from this value. *\n
@kotlin.internal.InlineOnly\n
public inline operator fun minus(other: UShort): UInt =
this.toUInt().minus(other.toUInt())\n\n /** Subtracts the other value from this value. *\n
@kotlin.internal.InlineOnly\n
public inline operator fun minus(other: UInt): UInt = this.toUInt().minus(other)\n\n /** Subtracts the other value from this value. *\n
@kotlin.internal.InlineOnly\n
public inline operator fun
minus(other: ULong): ULong = this.toULong().minus(other)\n\n /** Multiplies this value by the other value. *\n
@kotlin.internal.InlineOnly\n
public inline operator fun times(other: UByte): UInt =
this.toUInt().times(other.toUInt())\n\n /** Multiplies this value by the other value. *\n
@kotlin.internal.InlineOnly\n
public inline operator fun times(other: UShort): UInt =
this.toUInt().times(other.toUInt())\n\n /** Multiplies this value by the other value. *\n
@kotlin.internal.InlineOnly\n
public inline operator fun times(other: UInt): UInt = this.toUInt().times(other)\n\n /** Multiplies this value by the other value. *\n
@kotlin.internal.InlineOnly\n
public inline operator fun
times(other: ULong): ULong = this.toULong().times(other)\n\n /** Divides this value by the other value,
truncating the result to an integer that is closer to zero. *\n @kotlin.internal.InlineOnly\n
public inline operator
fun div(other: UByte): UInt = this.toUInt().div(other.toUInt())\n\n /** Divides this value by the other value,
truncating the result to an integer that is closer to zero. *\n @kotlin.internal.InlineOnly\n
public inline operator
fun div(other: UShort): UInt = this.toUInt().div(other.toUInt())\n\n /** Divides this value by the other value,
truncating the result to an integer that is closer to zero. *\n @kotlin.internal.InlineOnly\n
public inline operator
fun div(other: UInt): UInt = this.toUInt().div(other)\n\n /** Divides this value by the other value, truncating the
result to an integer that is closer to zero. *\n @kotlin.internal.InlineOnly\n
public inline operator fun div(other:
ULong): ULong = this.toULong().div(other)\n\n /**\n * Calculates the remainder of truncating division of this
value by the other value.\n * \n * The result is always less than the divisor.\n */\n
@kotlin.internal.InlineOnly\n
public inline operator fun rem(other: UByte): UInt =
this.toUInt().rem(other.toUInt())\n\n /**\n * Calculates the remainder of truncating division of this value by the
other value.\n * \n * The result is always less than the divisor.\n */\n @kotlin.internal.InlineOnly\n
public
inline operator fun rem(other: UShort): UInt = this.toUInt().rem(other.toUInt())\n\n /**\n * Calculates the
remainder of truncating division of this value by the other value.\n * \n * The result is always less than the
divisor.\n */\n @kotlin.internal.InlineOnly\n
public inline operator fun rem(other: UInt): UInt =
this.toUInt().rem(other)\n\n /**\n * Calculates the remainder of truncating division of this value by the other
value.\n * \n * The result is always less than the divisor.\n */\n @kotlin.internal.InlineOnly\n
public
inline operator fun rem(other: ULong): ULong = this.toULong().rem(other)\n\n /**\n * Divides this value by
the other value, flooring the result to an integer that is closer to negative infinity.\n * \n * For unsigned types,
the results of flooring division and truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n
public inline fun floorDiv(other: UByte): UInt = this.toUInt().floorDiv(other.toUInt())\n\n /**\n * Divides this
value by the other value, flooring the result to an integer that is closer to negative infinity.\n * \n * For unsigned
types, the results of flooring division and truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n
public inline fun floorDiv(other: UShort): UInt = this.toUInt().floorDiv(other.toUInt())\n\n /**\n * Divides this

```

```

value by the other value, flooring the result to an integer that is closer to negative infinity.\n * \n * For unsigned
types, the results of flooring division and truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n
public inline fun floorDiv(other: UInt): UInt = this.toUInt().floorDiv(other)\n /**\n * Divides this value by the
other value, flooring the result to an integer that is closer to negative infinity.\n * \n * For unsigned types, the
results of flooring division and truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n public
inline fun floorDiv(other: ULong): ULong = this.toULong().floorDiv(other)\n\n /**\n * Calculates the
remainder of flooring division of this value by the other value.\n * \n * The result is always less than the
divisor.\n * \n * For unsigned types, the remainders of flooring division and truncating division are the same.\n
*/\n @kotlin.internal.InlineOnly\n public inline fun mod(other: UByte): UByte =
this.toUInt().mod(other.toUInt()).toUByte()\n /**\n * Calculates the remainder of flooring division of this value
by the other value.\n * \n * The result is always less than the divisor.\n * \n * For unsigned types, the
remainders of flooring division and truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n
public inline fun mod(other: UShort): UShort = this.toUInt().mod(other.toUInt()).toUShort()\n /**\n *
Calculates the remainder of flooring division of this value by the other value.\n * \n * The result is always less
than the divisor.\n * \n * For unsigned types, the remainders of flooring division and truncating division are the
same.\n */\n @kotlin.internal.InlineOnly\n public inline fun mod(other: UInt): UInt =
this.toUInt().mod(other)\n /**\n * Calculates the remainder of flooring division of this value by the other
value.\n * \n * The result is always less than the divisor.\n * \n * For unsigned types, the remainders of
flooring division and truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n public inline fun
mod(other: ULong): ULong = this.toULong().mod(other)\n\n /**\n * Returns this value incremented by one.\n
*/\n * @sample samples.misc.Builtins.inc\n */\n @kotlin.internal.InlineOnly\n public inline operator fun
inc(): UShort = UShort(data.inc())\n\n /**\n * Returns this value decremented by one.\n */\n * @sample
samples.misc.Builtins.dec\n */\n @kotlin.internal.InlineOnly\n public inline operator fun dec(): UShort =
UShort(data.dec())\n\n /**\n * Creates a range from this value to the specified [other] value. */\n
@kotlin.internal.InlineOnly\n public inline operator fun rangeTo(other: UShort): UIntRange =
UIntRange(this.toUInt(), other.toUInt())\n\n /**\n * Performs a bitwise AND operation between the two values. */\n
@kotlin.internal.InlineOnly\n public inline infix fun and(other: UShort): UShort = UShort(this.data and
other.data)\n\n /**\n * Performs a bitwise OR operation between the two values. */\n @kotlin.internal.InlineOnly\n
public inline infix fun or(other: UShort): UShort = UShort(this.data or other.data)\n\n /**\n * Performs a bitwise XOR
operation between the two values. */\n @kotlin.internal.InlineOnly\n public inline infix fun xor(other: UShort):
UShort = UShort(this.data xor other.data)\n\n /**\n * Inverts the bits in this value. */\n @kotlin.internal.InlineOnly\n
public inline fun inv(): UShort = UShort(data.inv())\n\n /**\n * Converts this [UShort] value to [Byte].\n * \n
* If this value is less than or equals to [Byte.MAX_VALUE], the resulting `Byte` value represents\n * the same
numerical value as this `UShort`.\n * \n * The resulting `Byte` value is represented by the least significant 8 bits
of this `UShort` value.\n * \n * Note that the resulting `Byte` value may be negative.\n */\n
@kotlin.internal.InlineOnly\n public inline fun toByte(): Byte = data.toByte()\n\n /**\n * Converts this [UShort]
value to [Short].\n * \n * If this value is less than or equals to [Short.MAX_VALUE], the resulting `Short` value
represents\n * the same numerical value as this `UShort`. Otherwise the result is negative.\n * \n * The
resulting `Short` value has the same binary representation as this `UShort` value.\n */\n
@kotlin.internal.InlineOnly\n public inline fun toShort(): Short = data\n\n /**\n * Converts this [UShort] value
to [Int].\n * \n * The resulting `Int` value represents the same numerical value as this `UShort`.\n * \n * The
least significant 16 bits of the resulting `Int` value are the same as the bits of this `UShort` value,\n * whereas
the most significant 16 bits are filled with zeros.\n */\n @kotlin.internal.InlineOnly\n public inline fun toInt(): Int
= data.toInt() and 0xFFFF\n\n /**\n * Converts this [UShort] value to [Long].\n * \n * The resulting `Long`
value represents the same numerical value as this `UShort`.\n * \n * The least significant 16 bits of the resulting
`Long` value are the same as the bits of this `UShort` value,\n * whereas the most significant 48 bits are filled
with zeros.\n */\n @kotlin.internal.InlineOnly\n public inline fun toLong(): Long = data.toLong() and
0xFFFF\n\n /**\n * Converts this [UShort] value to [UByte].\n * \n * If this value is less than or equals to

```

```

[UByte.MAX_VALUE], the resulting `UByte` value represents
 * the same numerical value as this `UShort`.
 * The resulting `UByte` value is represented by the least significant 8 bits of this `UShort` value.
 @kotlin.internal.InlineOnly
 public inline fun toUByte(): UByte = data.toUByte()
 /** Returns this value.
 @kotlin.internal.InlineOnly
 public inline fun toUShort(): UShort = this
 /** Converts this [UShort]
 value to [UInt].
 * The resulting `UInt` value represents the same numerical value as this `UShort`.
 * The least significant 16 bits of the resulting `UInt` value are the same as the bits of this `UShort` value,
 * whereas the most significant 16 bits are filled with zeros.
 @kotlin.internal.InlineOnly
 public inline
 fun toUInt(): UInt = UInt(data.toInt() and 0xFFFF)
 /** Converts this [UShort] value to [ULong].
 * The resulting `ULong` value represents the same numerical value as this `UShort`.
 * The least
 significant 16 bits of the resulting `ULong` value are the same as the bits of this `UShort` value,
 * whereas the
 most significant 48 bits are filled with zeros.
 @kotlin.internal.InlineOnly
 public inline fun toULong():
 ULong = ULong(data.toLong() and 0xFFFF)
 /** Converts this [UShort] value to [Float].
 * The
 resulting `Float` value represents the same numerical value as this `UShort`.
 @kotlin.internal.InlineOnly
 public inline fun toFloat(): Float = this.toInt().toFloat()
 /** Converts this
 [UShort] value to [Double].
 * The resulting `Double` value represents the same numerical value as this
 `UShort`.
 @kotlin.internal.InlineOnly
 public inline fun toDouble(): Double =
 this.toInt().toDouble()
 public override fun toString(): String = toInt().toString()
 /** Converts this
 [Byte] value to [UShort].
 * If this value is positive, the resulting `UShort` value represents the same numerical
 value as this `Byte`.
 * The least significant 8 bits of the resulting `UShort` value are the same as the bits of this
 `Byte` value,
 * whereas the most significant 8 bits are filled with the sign bit of this value.
 @kotlin.internal.InlineOnly
 public inline fun Byte.toUShort(): UShort = UShort(this.toShort())
 /** Converts this [Short] value to
 [UShort].
 * If this value is positive, the resulting `UShort` value represents the same numerical value as this
 `Short`.
 * The resulting `UShort` value has the same binary representation as this `Short` value.
 @kotlin.internal.InlineOnly
 public inline fun Short.toUShort(): UShort = UShort(this)
 /** Converts this [Int] value to [UShort].
 * If
 this value is positive and less than or equals to [UShort.MAX_VALUE], the resulting `UShort` value represents
 * the same numerical value as this `Int`.
 * The resulting `UShort` value is represented by the least significant 16
 bits of this `Int` value.
 @kotlin.internal.InlineOnly
 public inline fun Int.toUShort(): UShort = UShort(this.toShort())
 /** Converts this [Long] value to
 [UShort].
 * If this value is positive and less than or equals to [UShort.MAX_VALUE], the resulting `UShort`
 value represents
 * the same numerical value as this `Long`.
 * The resulting `UShort` value is represented by
 the least significant 16 bits of this `Long` value.
 @kotlin.internal.InlineOnly
 public inline fun Long.toUShort(): UShort = UShort(this.toShort())
 /** Copyright 2010-2022 JetBrains s.r.o.
 and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license
 that can be found in the license/LICENSE.txt file.
 @kotlin.internal.AutoGeneratedFile
 DO NOT EDIT!
 package
 kotlin.ranges
 /** A range of values of type `Char`.
 @kotlin.internal.InlineOnly
 public class CharRange(start: Char, endInclusive:
 Char) : CharProgression(start, endInclusive, 1), ClosedRange<Char> {
 override val start: Char get() = first
 override val endInclusive: Char get() = last
 override fun contains(value: Char): Boolean = first <= value &&
 value <= last
 /** Checks whether the range is empty.
 * The range is empty if its start value is
 greater than the end value.
 override fun isEmpty(): Boolean = first > last
 override fun equals(other:
 Any?): Boolean =
 other is CharRange && (isEmpty() && other.isEmpty()) ||
 first == other.first && last
 == other.last
 override fun hashCode(): Int =
 if (isEmpty()) -1 else (31 * first.code + last.code)
 override fun toString(): String = "$first..$last"
 companion object {
 /** An empty range of values of
 type Char.
 public val EMPTY: CharRange = CharRange(1.toChar(), 0.toChar())
 }
 }
 /** A
 range of values of type `Int`.
 @kotlin.internal.InlineOnly
 public class IntRange(start: Int, endInclusive: Int) : IntProgression(start,

```

```

endInclusive, 1), ClosedRange<Int> {\n  override val start: Int get() = first\n  override val endInclusive: Int get() =
last\n\n  override fun contains(value: Int): Boolean = first <= value && value <= last\n\n  /** \n   * Checks
whether the range is empty.\n   * \n   * The range is empty if its start value is greater than the end value.\n   */\n
override fun isEmpty(): Boolean = first > last\n\n  override fun equals(other: Any?): Boolean =\n    other is
IntRange && (isEmpty() && other.isEmpty()) ||\n    first == other.first && last == other.last)\n\n  override fun
hashCode(): Int =\n    if (isEmpty()) -1 else (31 * first + last)\n\n  override fun toString(): String =
"\$first..$last"\n\n  companion object {\n    /** An empty range of values of type Int. */\n    public val
EMPTY: IntRange = IntRange(1, 0)\n  }\n}\n\n/** \n * A range of values of type `Long`.\n */\npublic class
LongRange(start: Long, endInclusive: Long) : LongProgression(start, endInclusive, 1), ClosedRange<Long> {\n
  override val start: Long get() = first\n  override val endInclusive: Long get() = last\n\n  override fun
contains(value: Long): Boolean = first <= value && value <= last\n\n  /** \n   * Checks whether the range is
empty.\n   * \n   * The range is empty if its start value is greater than the end value.\n   */\n
  override fun isEmpty(): Boolean = first > last\n\n  override fun equals(other: Any?): Boolean =\n    other is LongRange &&
(isEmpty() && other.isEmpty()) ||\n    first == other.first && last == other.last)\n\n  override fun hashCode(): Int
= \n    if (isEmpty()) -1 else (31 * (first xor (first ushr 32)) + (last xor (last ushr 32)))\n\n  override fun
toString(): String = "\$first..$last"\n\n  companion object {\n    /** An empty range of values of type Long. */\n
    public val EMPTY: LongRange = LongRange(1, 0)\n  }\n}\n\n"/\n * Copyright 2010-2021 JetBrains s.r.o.
and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license
that can be found in the license/LICENSE.txt file.\n
*/\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("CollectionsKt")\n@file:OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n\npackage kotlin.collections\n\nimport kotlin.contracts.*\nimport
kotlin.random.Random\n\ninternal object EmptyIterator : ListIterator<Nothing> {\n  override fun hasNext():
Boolean = false\n  override fun hasPrevious(): Boolean = false\n  override fun nextIndex(): Int = 0\n  override
fun previousIndex(): Int = -1\n  override fun next(): Nothing = throw NoSuchElementException()\n  override fun
previous(): Nothing = throw NoSuchElementException()\n}\n\ninternal object EmptyList : List<Nothing>,
Serializable, RandomAccess {\n  private const val serialVersionUID: Long = -7390468764508069838L\n\n  override fun equals(other: Any?): Boolean = other is List<*> && other.isEmpty()\n  override fun hashCode(): Int
= 1\n  override fun toString(): String = "[]"\n\n  override val size: Int get() = 0\n  override fun isEmpty():
Boolean = true\n  override fun contains(element: Nothing): Boolean = false\n  override fun containsAll(elements:
Collection<Nothing>): Boolean = elements.isEmpty()\n\n  override fun get(index: Int): Nothing = throw
IndexOutOfBoundsException("Empty list doesn't contain element at index $index.")\n  override fun
indexOf(element: Nothing): Int = -1\n  override fun lastIndexOf(element: Nothing): Int = -1\n\n  override fun
iterator(): Iterator<Nothing> = EmptyIterator\n  override fun listIterator(): ListIterator<Nothing> = EmptyIterator\n
  override fun listIterator(index: Int): ListIterator<Nothing> {\n    if (index != 0) throw
IndexOutOfBoundsException("Index: $index")\n    return EmptyIterator\n  }\n\n  override fun
subList(fromIndex: Int, toIndex: Int): List<Nothing> {\n    if (fromIndex == 0 && toIndex == 0) return this\n
throw IndexOutOfBoundsException("fromIndex: $fromIndex, toIndex: $toIndex")\n  }\n\n  private fun
readResolve(): Any = EmptyList\n}\n\ninternal fun <T> Array<out T>.asCollection(): Collection<T> =
ArrayAsCollection(this, isVarargs = false)\n\nprivate class ArrayAsCollection<T>(val values: Array<out T>, val
isVarargs: Boolean) : Collection<T> {\n  override val size: Int get() = values.size\n  override fun isEmpty():
Boolean = values.isEmpty()\n  override fun contains(element: T): Boolean = values.contains(element)\n  override
fun containsAll(elements: Collection<T>): Boolean = elements.all { contains(it) }\n  override fun iterator():
Iterator<T> = values.iterator()\n  // override hidden toArray implementation to prevent copying of values array\n
public fun toArray(): Array<out Any?> = values.copyToArrayOfAny(isVarargs)\n}\n\n/** \n * Returns an empty
read-only list. The returned list is serializable (JVM).\n */\n * @sample
samples.collections.Collections.Lists.emptyReadOnlyList\n */\npublic fun <T> emptyList(): List<T> =
EmptyList\n\n/** \n * Returns a new read-only list of given elements. The returned list is serializable (JVM).\n */
 * @sample samples.collections.Collections.Lists.readOnlyList\n */\npublic fun <T> listOf(vararg elements: T):

```

```

List<T> = if (elements.size > 0) elements.asList() else emptyList()\n\n/**\n * Returns an empty read-only list. The
returned list is serializable (JVM).\n * @sample samples.collections.Collections.Lists.emptyReadOnlyList\n
*/\n@kotlin.internal.InlineOnly\npublic inline fun <T> listOf(): List<T> = emptyList()\n\n/**\n * Returns an empty
new [MutableList].\n * @sample samples.collections.Collections.Lists.emptyMutableList\n
*/\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline fun <T> mutableListOf(): MutableList<T> =
ArrayList()\n\n/**\n * Returns an empty new [ArrayList].\n * @sample
samples.collections.Collections.Lists.emptyArrayList\n
*/\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline fun <T> arrayListOf(): ArrayList<T> =
ArrayList()\n\n/**\n * Returns a new [MutableList] with the given elements.\n * @sample
samples.collections.Collections.Lists.mutableList\n */\npublic fun <T> mutableListOf(vararg elements: T):
MutableList<T> =\n    if (elements.size == 0) ArrayList() else ArrayList(ArrayAsCollection(elements, isVarargs =
true))\n\n/**\n * Returns a new [ArrayList] with the given elements.\n * @sample
samples.collections.Collections.Lists.arrayList\n */\npublic fun <T> arrayListOf(vararg elements: T): ArrayList<T>
=\n    if (elements.size == 0) ArrayList() else ArrayList(ArrayAsCollection(elements, isVarargs = true))\n\n/**\n * Returns a new read-only list either of single given element, if it is not null, or empty list if the element is null. The
returned list is serializable (JVM).\n * @sample samples.collections.Collections.Lists.listOfNotNull\n */\npublic fun
<T : Any> listOfNotNull(element: T?): List<T> = if (element != null) listOf(element) else emptyList()\n\n/**\n * Returns a new read-only list only of those given elements, that are not null. The returned list is serializable
(JVM).\n * @sample samples.collections.Collections.Lists.listOfNotNull\n */\npublic fun <T : Any>
listOfNotNull(vararg elements: T?): List<T> = elements.filterNotNull()\n\n/**\n * Creates a new read-only list with
the specified [size], where each element is calculated by calling the specified\n * [init] function.\n * \n * The
function [init] is called for each list element sequentially starting from the first one.\n * It should return the value for
a list element given its index.\n * \n * @sample samples.collections.Collections.Lists.readOnlyListFromInitializer\n
*/\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline fun <T> List(size: Int, init: (index: Int) -> T):
List<T> = MutableList(size, init)\n\n/**\n * Creates a new mutable list with the specified [size], where each element
is calculated by calling the specified\n * [init] function.\n * \n * The function [init] is called for each list element
sequentially starting from the first one.\n * It should return the value for a list element given its index.\n * \n *
@sample samples.collections.Collections.Lists.mutableListFromInitializer\n
*/\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline fun <T> MutableList(size: Int, init: (index:
Int) -> T): MutableList<T> {\n    val list = ArrayList<T>(size)\n    repeat(size) { index -> list.add(init(index)) }\n
return list}\n\n/**\n * Builds a new read-only [List] by populating a [MutableList] using the given
[builderAction]\n * and returning a read-only list with the same elements.\n * \n * The list passed as a receiver to the
[builderAction] is valid only inside that function.\n * Using it outside of the function produces an unspecified
behavior.\n * \n * The returned list is serializable (JVM).\n * \n * @sample
samples.collections.Builders.Lists.buildListSample\n
*/\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
inline fun <E> buildList(@BuilderInference builderAction: MutableList<E>().->Unit): List<E> {\n    contract {
callsInPlace(builderAction, InvocationKind.EXACTLY_ONCE) }\n    return
buildListInternal(builderAction)\n}\n\n@PublishedApi\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\ninternal
expect inline fun <E> buildListInternal(builderAction: MutableList<E>().->Unit): List<E>\n\n/**\n * Builds a
new read-only [List] by populating a [MutableList] using the given [builderAction]\n * and returning a read-only list
with the same elements.\n * \n * The list passed as a receiver to the [builderAction] is valid only inside that
function.\n * Using it outside of the function produces an unspecified behavior.\n * \n * The returned list is
serializable (JVM).\n * \n * [capacity] is used to hint the expected number of elements added in the
[builderAction].\n * \n * @throws IllegalArgumentException if the given [capacity] is negative.\n * \n * @sample
samples.collections.Builders.Lists.buildListSampleWithCapacity\n
*/\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
inline fun <E> buildList(capacity: Int, @BuilderInference builderAction: MutableList<E>().->Unit): List<E> {\n

```

```

contract { callsInPlace(builderAction, InvocationKind.EXACTLY_ONCE) }
return buildListInternal(capacity, builderAction)
@PublishedApi
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
internal expect inline fun <E> buildListInternal(capacity: Int, builderAction: MutableList<E>.() -> Unit): List<E>
Returns an [IntRange] of the valid indices for this collection.
@sample
samples.collections.Collections.Collections.indicesOfCollection
public val Collection<*>.indices: IntRange
get() = 0..size - 1
Returns the index of the last item in the list or -1 if the list is empty.
@sample
samples.collections.Collections.Collections.lastIndexOfList
public val <T> List<T>.lastIndex: Int
get() = this.size - 1
Returns `true` if the collection is not empty.
@sample
samples.collections.Collections.Collections.collectionIsNotEmpty
public inline fun <T> Collection<T>.isNotEmpty(): Boolean = !isEmpty()
Returns `true` if this nullable collection is either null or empty.
@sample
samples.collections.Collections.Collections.collectionOrNullOrNullEmpty
public inline fun <T> Collection<T>?.isNullOrNullEmpty(): Boolean {
    contract {
        returns(false) implies (this@isNullOrNullEmpty != null)
    }
    return this == null || this.isEmpty()
}
Returns this Collection if it's not `null` and the empty list otherwise.
@sample
samples.collections.Collections.Collections.collectionOrNullEmpty
public inline fun <T> Collection<T>?.orEmpty(): Collection<T> = this ?: emptyList()
Returns this List if it's not `null` and the empty list otherwise.
@sample
samples.collections.Collections.Collections.listOrNullEmpty
public inline fun <T> List<T>?.orEmpty(): List<T> = this ?: emptyList()
Returns this collection if it's not empty or the result of calling [defaultValue] function if the collection is empty.
@sample
samples.collections.Collections.Collections.collectionIfEmpty
public inline fun <C, R> C.ifEmpty(defaultValue: () -> R): R where C : Collection<*>, C : R =
    if (isEmpty()) defaultValue() else this
Checks if all elements in the specified collection are contained in this collection.
Allows to overcome type-safety restriction of `containsAll` that requires to pass a collection of type `Collection<E>`.
@sample
samples.collections.Collections.Collections.collectionContainsAll
public inline fun <@kotlin.internal.OnlyInputTypes T> Collection<T>.containsAll(elements: Collection<T>): Boolean = this.containsAll(elements)
Returns a new list with the elements of this list randomly shuffled using the specified [random] instance as the source of randomness.
@sample
samples.collections.Collections.Collections.collectionShuffled
public fun <T> Iterable<T>.shuffled(random: Random): List<T> = toMutableList().apply { shuffle(random) }
optimizeReadOnlyList() = when (size) {
    0 -> emptyList()
    1 -> listOf(this[0])
    else -> this
}
Searches this list or its range for the provided [element] using the binary search algorithm.
The list is expected to be sorted into ascending order according to the Comparable natural ordering of its elements, otherwise the result is undefined.
If the list contains multiple elements equal to the specified [element], there is no guarantee which one will be found.
`null` value is considered to be less than any non-null value.
@return the index of the element, if it is contained in the list within the specified range; otherwise, the inverted insertion point `(-insertion point - 1)`.
The insertion point is defined as the index at which the element should be inserted, so that the list (or the specified subrange of list) still remains sorted.
@sample
samples.collections.Collections.Collections.binarySearchOnComparable
@sample
samples.collections.Collections.Collections.binarySearchWithBoundaries
public fun <T : Comparable<T>> List<T>.binarySearch(element: T?, fromIndex: Int = 0, toIndex: Int = size): Int {
    rangeCheck(size, fromIndex, toIndex)
    var low = fromIndex
    var high = toIndex - 1
    while (low <= high) {
        val mid = (low + high).ushr(1) // safe from overflows
        val midVal = get(mid)
        val cmp = compareValues(midVal, element)
        if (cmp < 0) low = mid + 1
        else if (cmp > 0) high = mid - 1
        else return mid // key found
    }
    return -(low + 1) // key not found
}
Searches this list or its range for the provided [element] using the binary search algorithm.
The list is expected to be sorted into ascending order according to the specified [comparator], otherwise the result is undefined.
If the list contains

```

multiple elements equal to the specified [element], there is no guarantee which one will be found. `null` value is considered to be less than any non-null value. `@return` the index of the element, if it is contained in the list within the specified range; otherwise, the inverted insertion point `(-insertion point - 1)`. The insertion point is defined as the index at which the element should be inserted, so that the list (or the specified subrange of list) still remains sorted according to the specified [comparator]. `@sample`

```

samples.collections.Collections.Lists.binarySearchWithComparator
public fun <T>
List<T>.binarySearch(element: T, comparator: Comparator<in T>, fromIndex: Int = 0, toIndex: Int = size): Int {
    rangeCheck(size, fromIndex, toIndex)
    var low = fromIndex
    var high = toIndex - 1
    while (low <= high) {
        val mid = (low + high).ushr(1) // safe from overflows
        val midVal = get(mid)
        val cmp = comparator.compare(midVal, element)
        if (cmp < 0) low = mid + 1
        else if (cmp > 0) high = mid - 1
        else return mid // key found
    }
    return -(low + 1) // key not found
}

```

Searches this list or its range for an element having the key returned by the specified [selector] function equal to the provided [key] value using the binary search algorithm. The list is expected to be sorted into ascending order according to the Comparable natural ordering of keys of its elements. otherwise the result is undefined. If the list contains multiple elements with the specified [key], there is no guarantee which one will be found. `null` value is considered to be less than any non-null value. `@return` the index of the element with the specified [key], if it is contained in the list within the specified range; otherwise, the inverted insertion point `(-insertion point - 1)`. The insertion point is defined as the index at which the element should be inserted, so that the list (or the specified subrange of list) still remains sorted. `@sample`

```

samples.collections.Collections.Lists.binarySearchByKey
public inline fun <T, K : Comparable<K>>
List<T>.binarySearchBy(
    key: K?,
    fromIndex: Int = 0,
    toIndex: Int = size,
    crossinline selector: (T) -> K?): Int =
    binarySearch(fromIndex, toIndex) { compareValues(selector(it), key) }

```

do not introduce this overload --- too rare

```

public fun <T, K> List<T>.binarySearchBy(key: K, comparator: Comparator<K>,
    fromIndex: Int = 0, toIndex: Int = size(), selector: (T) -> K): Int =
    binarySearch(fromIndex, toIndex) {
        comparator.compare(selector(it), key)
    }

```

Searches this list or its range for an element for which the given [comparison] function returns zero using the binary search algorithm. The list is expected to be sorted so that the signs of the [comparison] function's return values ascend on the list elements, i.e. negative values come before zero and zeroes come before positive values. Otherwise, the result is undefined. If the list contains multiple elements for which [comparison] returns zero, there is no guarantee which one will be found. `@param` comparison function that returns zero when called on the list element being searched. On the elements coming before the target element, the function must return negative values; on the elements coming after the target element, the function must return positive values. `@return` the index of the found element, if it is contained in the list within the specified range; otherwise, the inverted insertion point `(-insertion point - 1)`. The insertion point is defined as the index at which the element should be inserted, so that the list (or the specified subrange of list) still remains sorted. `@sample`

```

samples.collections.Collections.Lists.binarySearchWithComparisonFunction
public fun <T>
List<T>.binarySearch(fromIndex: Int = 0, toIndex: Int = size, comparison: (T) -> Int): Int {
    rangeCheck(size, fromIndex, toIndex)
    var low = fromIndex
    var high = toIndex - 1
    while (low <= high) {
        val mid = (low + high).ushr(1) // safe from overflows
        val midVal = get(mid)
        val cmp = comparison(midVal)
        if (cmp < 0) low = mid + 1
        else if (cmp > 0) high = mid - 1
        else return mid // key found
    }
    return -(low + 1) // key not found
}

```

Checks that `from` and `to` are in the range of `[0..size]` and throws an appropriate exception, if they aren't.

```

private fun rangeCheck(size: Int,
    fromIndex: Int, toIndex: Int) {
    when {
        fromIndex > toIndex -> throw
        IllegalArgumentException("fromIndex ($fromIndex) is greater than toIndex ($toIndex).")
        fromIndex < 0 ->
        throw IndexOutOfBoundsException("fromIndex ($fromIndex) is less than zero.")
        toIndex > size -> throw
        IndexOutOfBoundsException("toIndex ($toIndex) is greater than size ($size).")
    }
}

```

`@PublishedApi`
`@SinceKotlin("1.3")`
`internal` expect fun checkIndexOverflow(index: Int): Int
`@PublishedApi`
`@SinceKotlin("1.3")`
`internal` expect fun checkCountOverflow(count: Int): Int

```

Int\n\n\n@PublishedApi\n@SinceKotlin("1.3")\n\ninternal fun throwIndexOverflow() { throw
ArithmeticException("Index overflow has happened.") }\n\n\n@PublishedApi\n@SinceKotlin("1.3")\n\ninternal fun
throwCountOverflow() { throw ArithmeticException("Count overflow has happened.") }\n\n\n"/*\n * Copyright
2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed
by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("MapsKt")\n@file:OptIn(kotlin.experiment
al.ExperimentalTypeInference::class)\n\npackage kotlin.collections\n\nimport kotlin.contracts.*\n\nprivate object
EmptyMap : Map<Any?, Nothing>, Serializable {\n    private const val serialVersionUID: Long =
8246714829545688274\n\n    override fun equals(other: Any?): Boolean = other is Map<*, *> &&
other.isEmpty()\n    override fun hashCode(): Int = 0\n    override fun toString(): String = "{}"\n    override val
size: Int get() = 0\n    override fun isEmpty(): Boolean = true\n\n    override fun containsKey(key: Any?): Boolean =
false\n    override fun containsValue(value: Nothing): Boolean = false\n    override fun get(key: Any?): Nothing? =
null\n    override val entries: Set<Map.Entry<Any?, Nothing>> get() = EmptySet\n    override val keys: Set<Any?>
get() = EmptySet\n    override val values: Collection<Nothing> get() = EmptyList\n\n    private fun readResolve():
Any = EmptyMap\n}\n\n/**\n * Returns an empty read-only map of specified type.\n * \n * The returned map is
serializable (JVM).\n * @sample samples.collections.Maps.Instantiation.emptyReadOnlyMap\n */\n\npublic fun <K,
V> emptyMap(): Map<K, V> = @Suppress("UNCHECKED_CAST") (EmptyMap as Map<K, V>)\n\n/**\n *
Returns a new read-only map with the specified contents, given as a list of pairs\n * where the first value is the key
and the second is the value.\n * \n * If multiple pairs have the same key, the resulting map will contain the value
from the last of those pairs.\n * \n * Entries of the map are iterated in the order they were specified.\n * \n * The
returned map is serializable (JVM).\n * \n * @sample samples.collections.Maps.Instantiation.mapFromPairs\n */\n\npublic fun <K, V> mapOf(vararg pairs: Pair<K, V>): Map<K, V> =\n    if (pairs.size > 0)
pairs.toMap(LinkedHashMap(mapCapacity(pairs.size))) else emptyMap()\n\n/**\n * Returns an empty read-only
map.\n * \n * The returned map is serializable (JVM).\n * @sample
samples.collections.Maps.Instantiation.emptyReadOnlyMap\n */\n\n@kotlin.internal.InlineOnly\n\npublic inline fun
<K, V> mapOf(): Map<K, V> = emptyMap()\n\n/**\n * Returns an empty new [MutableMap].\n * \n * The returned
map preserves the entry iteration order.\n * @sample samples.collections.Maps.Instantiation.emptyMutableMap\n */\n\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\n\npublic inline fun <K, V> mutableMapOf():
MutableMap<K, V> = LinkedHashMap()\n\n/**\n * Returns a new [MutableMap] with the specified contents, given
as a list of pairs\n * where the first component is the key and the second is the value.\n * \n * If multiple pairs have
the same key, the resulting map will contain the value from the last of those pairs.\n * \n * Entries of the map are
iterated in the order they were specified.\n * \n * @sample
samples.collections.Maps.Instantiation.mutableMapFromPairs\n * @sample
samples.collections.Maps.Instantiation.emptyMutableMap\n */\n\npublic fun <K, V> mutableMapOf(vararg pairs:
Pair<K, V>): MutableMap<K, V> =\n    LinkedHashMap<K, V>(mapCapacity(pairs.size)).apply { putAll(pairs)
}\n\n/**\n * Returns an empty new [HashMap].\n * \n * @sample
samples.collections.Maps.Instantiation.emptyHashMap\n */\n\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\n\npublic inline fun <K, V> hashMapOf(): HashMap<K, V>
= HashMap<K, V>()\n\n/**\n * Returns a new [HashMap] with the specified contents, given as a list of pairs\n *
where the first component is the key and the second is the value.\n * \n * @sample
samples.collections.Maps.Instantiation.hashMapFromPairs\n */\n\npublic fun <K, V> hashMapOf(vararg pairs:
Pair<K, V>): HashMap<K, V> = HashMap<K, V>(mapCapacity(pairs.size)).apply { putAll(pairs) }\n\n/**\n *
Returns an empty new [LinkedHashMap].\n */\n\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\n\npublic inline
fun <K, V> linkedMapOf(): LinkedHashMap<K, V> = LinkedHashMap<K, V>()\n\n/**\n * Returns a new
[LinkedHashMap] with the specified contents, given as a list of pairs\n * where the first component is the key and
the second is the value.\n * \n * If multiple pairs have the same key, the resulting map will contain the value from the
last of those pairs.\n * \n * Entries of the map are iterated in the order they were specified.\n * \n * @sample
samples.collections.Maps.Instantiation.linkedMapFromPairs\n */\n\npublic fun <K, V> linkedMapOf(vararg pairs:

```

Pair<K, V>): LinkedHashMap<K, V> = pairs.toMap(LinkedHashMap(mapCapacity(pairs.size)))\n\n**\n * Builds a new read-only [Map] by populating a [MutableMap] using the given [builderAction]\n * and returning a read-only map with the same key-value pairs.\n * The map passed as a receiver to the [builderAction] is valid only inside that function.\n * Using it outside of the function produces an unspecified behavior.\n * Entries of the map are iterated in the order they were added by the [builderAction].\n * The returned map is serializable (JVM).\n * @sample samples.collections.Builders.Maps.buildMapSample\n

```
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> buildMap(@BuilderInference builderAction: MutableMap<K, V>().->Unit): Map<K, V> {\n    contract { callsInPlace(builderAction, InvocationKind.EXACTLY_ONCE) }\n    return\n    buildMapInternal(builderAction)\n}\n\n@PublishedApi\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\ninternal expect inline fun <K, V> buildMapInternal(builderAction: MutableMap<K, V>().->Unit): Map<K, V>\n\n**\n * Builds a new read-only [Map] by populating a [MutableMap] using the given [builderAction]\n * and returning a read-only map with the same key-value pairs.\n * The map passed as a receiver to the [builderAction] is valid only inside that function.\n * Using it outside of the function produces an unspecified behavior.\n * [capacity] is used to hint the expected number of pairs added in the [builderAction].\n * Entries of the map are iterated in the order they were added by the [builderAction].\n * The returned map is serializable (JVM).\n * @throws IllegalArgumentException if the given [capacity] is negative.\n * @sample samples.collections.Builders.Maps.buildMapSample\n
```

```
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> buildMap(capacity: Int, @BuilderInference builderAction: MutableMap<K, V>().->Unit): Map<K, V> {\n    contract { callsInPlace(builderAction, InvocationKind.EXACTLY_ONCE) }\n    return\n    buildMapInternal(capacity,\n    builderAction)\n}\n\n@PublishedApi\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\ninternal expect inline fun <K, V> buildMapInternal(capacity: Int, builderAction: MutableMap<K, V>().->Unit): Map<K, V>\n\n**\n * Calculate the initial capacity of a map.\n\n@PublishedApi\ninternal expect fun mapCapacity(expectedSize: Int): Int\n\n**\n * Returns `true` if this map is not empty.\n * @sample samples.collections.Maps.Usage.mapIsNotEmpty\n
```

```
*\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<out K, V>.isEmpty(): Boolean = !isNotEmpty()\n\n**\n * Returns `true` if this nullable map is either null or empty.\n * @sample samples.collections.Maps.Usage.mapIsNullOrEmpty\n
```

```
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<out K, V>?.isNullOrEmpty(): Boolean {\n    contract {\n        returns(false) implies (this@isNullOrEmpty != null)\n    }\n    return this == null || isEmpty()\n}\n\n**\n * Returns the [Map] if its not `null`, or the empty [Map] otherwise.\n * @sample samples.collections.Maps.Usage.mapOrElse\n
```

```
*\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<K, V>?.orElse(): Map<K, V> = this ?: emptyMap()\n\n**\n * Returns this map if it's not empty\n * or the result of calling [defaultValue] function if the map is empty.\n * @sample samples.collections.Maps.Usage.mapIfEmpty\n
```

```
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline fun <M, R> M.ifEmpty(defaultValue: () -> R): R where M : Map<*, *>, M : R =\n    if (isEmpty())\n        defaultValue()\n    else this\n\n**\n * Checks if the map contains the given key.\n * This method allows to use the `x in map` syntax for checking whether an object is contained in the map.\n * @sample samples.collections.Maps.Usage.containsKey\n
```

```
*\n@kotlin.internal.OnlyInputTypes K, V> Map<out K, V>.contains(key: K): Boolean = containsKey(key)\n\n**\n * Returns the value corresponding to the given [key], or `null` if such a key is not present in the map.\n\n@kotlin.internal.InlineOnly\npublic inline operator fun <@kotlin.internal.OnlyInputTypes K, V> Map<out K, V>.get(key: K): V? =\n    @Suppress("UNCHECKED_CAST") (this as Map<K, V>).get(key)\n\n**\n * Allows to use the index operator for storing values in a mutable map.\n\n@kotlin.internal.InlineOnly\npublic inline operator fun <K, V> MutableMap<K, V>.set(key: K, value: V): Unit {\n    put(key, value)\n}\n\n**\n * Returns `true` if the map contains the specified [key].\n * Allows to overcome type-safety restriction of `containsKey` that requires to pass a key of type `K`.\n\n@kotlin.internal.InlineOnly\npublic inline fun
```

```

<@kotlin.internal.OnlyInputTypes K> Map<out K, *>.containsKey(key: K): Boolean =\n
@Suppress("UNCHECKED_CAST") (this as Map<K, *>).containsKey(key)\n\n/**\n * Returns `true` if the map\n
maps one or more keys to the specified [value].\n\n * Allows to overcome type-safety restriction of\n
`containsValue` that requires to pass a value of type `V`.\n\n * @sample\n
samples.collections.Maps.Usage.containsValue\n\n * \/\n\n@Suppress("EXTENSION_SHADOWED_BY_MEMBER")\n
// false warning, extension takes precedence in some cases\n\n@kotlin.internal.InlineOnly\n\npublic inline fun <K,\n
@kotlin.internal.OnlyInputTypes V> Map<K, V>.containsValue(value: V): Boolean =\n
this.containsValue(value)\n\n\n/**\n * Removes the specified key and its corresponding value from this map.\n\n * \n
* @return the previous value associated with the key, or `null` if the key was not present in the map.\n\n * Allows to\n
overcome type-safety restriction of `remove` that requires to pass a key of type `K`.\n\n
* \/\n\n@kotlin.internal.InlineOnly\n\npublic inline fun <@kotlin.internal.OnlyInputTypes K, V> MutableMap<out K,\n
V>.remove(key: K): V? =\n\n @Suppress("UNCHECKED_CAST") (this as MutableMap<K,\n
V>).remove(key)\n\n\n/**\n * Returns the key component of the map entry.\n\n * This method allows to use\n
destructuring declarations when working with maps, for example:\n\n * ```\n * for ((key, value) in map) {\n * // do\n
something with the key and the value\n * }\n * ```\n * \/\n\n@kotlin.internal.InlineOnly\n\npublic inline operator fun <K,\n
V> Map.Entry<K, V>.component1(): K = key\n\n\n/**\n * Returns the value component of the map entry.\n\n * This method allows to use destructuring declarations when working with maps, for example:\n\n * ```\n * for ((key,\n
value) in map) {\n * // do something with the key and the value\n * }\n * ```\n\n
* \/\n\n@kotlin.internal.InlineOnly\n\npublic inline operator fun <K, V> Map.Entry<K, V>.component2(): V =\n
value\n\n\n/**\n * Converts entry to [Pair] with key being first component and value being second.\n\n
* \/\n\n@kotlin.internal.InlineOnly\n\npublic inline fun <K, V> Map.Entry<K, V>.toPair(): Pair<K, V> = Pair(key,\n
value)\n\n\n/**\n * Returns the value for the given key, or the result of the [defaultValue] function if there was no\n
entry for the given key.\n\n * @sample samples.collections.Maps.Usage.getOrElse\n\n
* \/\n\n@kotlin.internal.InlineOnly\n\npublic inline fun <K, V> Map<K, V>.getOrElse(key: K, defaultValue: () -> V): V\n
= get(key) ?: defaultValue()\n\n\ninternal inline fun <K, V> Map<K, V>.getOrElseNullable(key: K, defaultValue: ()\n
-> V): V {\n\n val value = get(key)\n\n if (value == null && !containsKey(key)) {\n\n return defaultValue()\n\n }\n
else {\n\n @Suppress("UNCHECKED_CAST")\n\n return value as V\n\n }\n\n}\n\n\n/**\n * Returns the value\n
for the given [key] or throws an exception if there is no such key in the map.\n\n * If the map was created by\n
[withDefault], resorts to its `defaultValue` provider function\n\n * instead of throwing an exception.\n\n * \n
* @throws NoSuchElementException when the map doesn't contain a value for the specified key and\n\n * no implicit default\n
value was provided for that map.\n\n * \/\n\n@SinceKotlin("1.1")\n\npublic fun <K, V> Map<K, V>.getValue(key: K): V\n
= getOrImplicitDefault(key)\n\n\n/**\n * Returns the value for the given key. If the key is not found in the map, calls\n
the [defaultValue] function,\n\n * puts its result into the map under the given key and returns it.\n\n * Note that the\n
operation is not guaranteed to be atomic if the map is being modified concurrently.\n\n * @sample\n
samples.collections.Maps.Usage.getOrPut\n\n * \/\n\npublic inline fun <K, V> MutableMap<K, V>.getOrPut(key: K,\n
defaultValue: () -> V): V {\n\n val value = get(key)\n\n return if (value == null) {\n\n val answer =\n
defaultValue()\n\n put(key, answer)\n\n answer\n\n } else {\n\n value\n\n }\n\n}\n\n\n/**\n * Returns an\n
[Iterator] over the entries in the [Map].\n\n * @sample samples.collections.Maps.Usage.forOverEntries\n\n
* \/\n\n@kotlin.internal.InlineOnly\n\npublic inline operator fun <K, V> Map<out K, V>.iterator():\n
Iterator<Map.Entry<K, V>> = entries.iterator()\n\n\n/**\n * Returns a [MutableIterator] over the mutable entries in\n
the [MutableMap].\n\n * \n
* \/\n\n@kotlin.jvm.JvmName("mutableIterator")\n\n@kotlin.internal.InlineOnly\n\npublic\n
inline operator fun <K, V> MutableMap<K, V>.iterator(): MutableIterator<MutableMap.MutableEntry<K, V>> =\n
entries.iterator()\n\n\n/**\n * Populates the given [destination] map with entries having the keys of this map and the\n
values obtained\n\n * by applying the [transform] function to each entry in this [Map].\n\n * \/\n\npublic inline fun <K, V,\n
R, M : MutableMap<in K, in R>> Map<out K, V>.mapValuesTo(destination: M, transform: (Map.Entry<K, V>) ->\n
R): M {\n\n return entries.associateByTo(destination, { it.key }, transform)\n\n}\n\n\n/**\n * Populates the given\n
[destination] map with entries having the keys obtained\n\n * by applying the [transform] function to each entry in this\n
[Map] and the values of this map.\n\n * In case if any two entries are mapped to the equal keys, the value of the

```

latter one will overwrite the value associated with the former one.

```

public inline fun <K, V, R, M :
MutableMap<in R, in V>> Map<out K, V>.mapKeysTo(destination: M, transform: (Map.Entry<K, V>) -> R): M
{
    return entries.associateByTo(destination, transform, { it.value })
}

```

* Puts all the given [pairs] into this [MutableMap] with the first component in the pair being the key and the second the value.

```

public fun <K,
V> MutableMap<in K, in V>.putAll(pairs: Array<out Pair<K, V>>): Unit {
    for ((key, value) in pairs) {
        put(key, value)
    }
}

```

* Puts all the elements of the given collection into this [MutableMap] with the first component in the pair being the key and the second the value.

```

public fun <K, V> MutableMap<in K, in
V>.putAll(pairs: Iterable<Pair<K, V>>): Unit {
    for ((key, value) in pairs) {
        put(key, value)
    }
}

```

* Puts all the elements of the given sequence into this [MutableMap] with the first component in the pair being the key and the second the value.

```

public fun <K, V> MutableMap<in K, in V>.putAll(pairs:
Sequence<Pair<K, V>>): Unit {
    for ((key, value) in pairs) {
        put(key, value)
    }
}

```

* Returns a new map with entries having the keys of this map and the values obtained by applying the [transform] function to each entry in this [Map].

* The returned map preserves the entry iteration order of the original map.

```

@sample samples.collections.Maps.Transformations.mapValues
public inline fun <K, V, R> Map<out K,
V>.mapValues(transform: (Map.Entry<K, V>) -> R): Map<K, R> {
    return mapValuesTo(LinkedHashMap<K,
R>(mapCapacity(size)), transform) // .optimizeReadOnlyMap()
}

```

* Returns a new Map with entries having the keys obtained by applying the [transform] function to each entry in this [Map] and the values of this map.

* In case if any two entries are mapped to the equal keys, the value of the latter one will overwrite the value associated with the former one.

* The returned map preserves the entry iteration order of the original map.

```

@sample samples.collections.Maps.Transformations.mapKeys
public inline fun <K, V, R>
Map<out K, V>.mapKeys(transform: (Map.Entry<K, V>) -> R): Map<R, V> {
    return
mapKeysTo(LinkedHashMap<R, V>(mapCapacity(size)), transform) // .optimizeReadOnlyMap()
}

```

* Returns a map containing all key-value pairs with keys matching the given [predicate].

* The returned map preserves the entry iteration order of the original map.

```

@sample samples.collections.Maps.Filtering.filterKeys
public inline fun <K, V> Map<out K, V>.filterKeys(predicate: (K) -> Boolean): Map<K, V> {
    val result =
LinkedHashMap<K, V>()
    for (entry in this) {
        if (predicate(entry.key)) {
            result.put(entry.key,
entry.value)
        }
    }
    return result
}

```

* Returns a map containing all key-value pairs with values matching the given [predicate].

* The returned map preserves the entry iteration order of the original map.

```

@sample samples.collections.Maps.Filtering.filterValues
public inline fun <K, V> Map<out K,
V>.filterValues(predicate: (V) -> Boolean): Map<K, V> {
    val result = LinkedHashMap<K, V>()
    for (entry
in this) {
        if (predicate(entry.value)) {
            result.put(entry.key, entry.value)
        }
    }
    return
result
}

```

* Appends all entries matching the given [predicate] into the mutable map given as [destination] parameter.

* @return the destination map.

```

@sample samples.collections.Maps.Filtering.filterTo
public inline fun <K, V, M : MutableMap<in K, in V>> Map<out K, V>.filterTo(destination: M, predicate:
(Map.Entry<K, V>) -> Boolean): M {
    for (element in this) {
        if (predicate(element)) {
            destination.put(element.key, element.value)
        }
    }
    return destination
}

```

* Returns a new map containing all key-value pairs matching the given [predicate].

* The returned map preserves the entry iteration order of the original map.

```

@sample samples.collections.Maps.Filtering.filter
public inline fun <K, V>
Map<out K, V>.filter(predicate: (Map.Entry<K, V>) -> Boolean): Map<K, V> {
    return
filterTo(LinkedHashMap<K, V>(), predicate)
}

```

* Appends all entries not matching the given [predicate] into the given [destination].

* @return the destination map.

```

@sample
samples.collections.Maps.Filtering.filterNotTo
public inline fun <K, V, M : MutableMap<in K, in V>>
Map<out K, V>.filterNotTo(destination: M, predicate: (Map.Entry<K, V>) -> Boolean): M {
    for (element in
this) {
        if (!predicate(element)) {
            destination.put(element.key, element.value)
        }
    }
    return
destination
}

```

* Returns a new map containing all key-value pairs not matching the given [predicate].

* The returned map preserves the entry iteration order of the original map.

```

@sample
samples.collections.Maps.Filtering.filterNot
public inline fun <K, V> Map<out K, V>.filterNot(predicate:
(Map.Entry<K, V>) -> Boolean): Map<K, V> {
    return filterNotTo(LinkedHashMap<K, V>(),
}

```

predicate)\n\n/**\n * Returns a new map containing all key-value pairs from the given collection of pairs.\n *\n * The returned map preserves the entry iteration order of the original collection.\n * If any of two pairs would have the same key the last one gets added to the map.\n */\npublic fun <K, V> Iterable<Pair<K, V>>.toMap(): Map<K, V> {\n if (this is Collection) {\n return when (size) {\n 0 -> emptyMap()\n 1 -> mapOf(if (this is List) this[0] else iterator().next())\n else -> toMap(LinkedHashMap<K, V>(mapCapacity(size)))\n }\n }\n return toMap(LinkedHashMap<K, V>()).optimizeReadOnlyMap()\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs from the given collection of pairs.\n */\npublic fun <K, V, M : MutableMap<in K, in V>> Iterable<Pair<K, V>>.toMap(destination: M): M =\n destination.apply { putAll(this@toMap) }\n\n/**\n * Returns a new map containing all key-value pairs from the given array of pairs.\n *\n * The returned map preserves the entry iteration order of the original array.\n * If any of two pairs would have the same key the last one gets added to the map.\n */\npublic fun <K, V> Array<out Pair<K, V>>.toMap(): Map<K, V> = when (size) {\n 0 -> emptyMap()\n 1 -> mapOf(this[0])\n else -> toMap(LinkedHashMap<K, V>(mapCapacity(size)))\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs from the given array of pairs.\n */\npublic fun <K, V, M : MutableMap<in K, in V>> Array<out Pair<K, V>>.toMap(destination: M): M =\n destination.apply { putAll(this@toMap) }\n\n/**\n * Returns a new map containing all key-value pairs from the given sequence of pairs.\n *\n * The returned map preserves the entry iteration order of the original sequence.\n * If any of two pairs would have the same key the last one gets added to the map.\n */\npublic fun <K, V> Sequence<Pair<K, V>>.toMap(): Map<K, V> = toMap(LinkedHashMap<K, V>()).optimizeReadOnlyMap()\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs from the given sequence of pairs.\n */\npublic fun <K, V, M : MutableMap<in K, in V>> Sequence<Pair<K, V>>.toMap(destination: M): M =\n destination.apply { putAll(this@toMap) }\n\n/**\n * Returns a new read-only map containing all key-value pairs from the original map.\n *\n * The returned map preserves the entry iteration order of the original map.\n */\n@SinceKotlin("1.1")\npublic fun <K, V> Map<out K, V>.toMap(): Map<K, V> = when (size) {\n 0 -> emptyMap()\n 1 -> toSingletonMap()\n else -> toMutableMap()\n}\n\n/**\n * Returns a new mutable map containing all key-value pairs from the original map.\n *\n * The returned map preserves the entry iteration order of the original map.\n */\n@SinceKotlin("1.1")\npublic fun <K, V> Map<out K, V>.toMutableMap(): MutableMap<K, V> = LinkedHashMap(this)\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs from the given map.\n */\n@SinceKotlin("1.1")\npublic fun <K, V, M : MutableMap<in K, in V>> Map<out K, V>.toMap(destination: M): M =\n destination.apply { putAll(this@toMap) }\n\n/**\n * Creates a new read-only map by replacing or adding an entry to this map from a given key-value [pair].\n *\n * The returned map preserves the entry iteration order of the original map.\n * The [pair] is iterated in the end if it has a unique key.\n */\npublic operator fun <K, V> Map<out K, V>.plus(pair: Pair<K, V>): Map<K, V> =\n if (this.isEmpty()) mapOf(pair) else LinkedHashMap(this).apply { put(pair.first, pair.second) }\n\n/**\n * Creates a new read-only map by replacing or adding entries to this map from a given collection of key-value [pairs].\n *\n * The returned map preserves the entry iteration order of the original map.\n * Those [pairs] with unique keys are iterated in the end in the order of [pairs] collection.\n */\npublic operator fun <K, V> Map<out K, V>.plus(pairs: Iterable<Pair<K, V>>): Map<K, V> =\n if (this.isEmpty()) pairs.toMap() else LinkedHashMap(this).apply { putAll(pairs) }\n\n/**\n * Creates a new read-only map by replacing or adding entries to this map from a given array of key-value [pairs].\n *\n * The returned map preserves the entry iteration order of the original map.\n * Those [pairs] with unique keys are iterated in the end in the order of [pairs] array.\n */\npublic operator fun <K, V> Map<out K, V>.plus(pairs: Array<out Pair<K, V>>): Map<K, V> =\n if (this.isEmpty()) pairs.toMap() else LinkedHashMap(this).apply { putAll(pairs) }\n\n/**\n * Creates a new read-only map by replacing or adding entries to this map from a given sequence of key-value [pairs].\n *\n * The returned map preserves the entry iteration order of the original map.\n * Those [pairs] with unique keys are iterated in the end in the order of [pairs] sequence.\n */\npublic operator fun <K, V> Map<out K, V>.plus(pairs: Sequence<Pair<K, V>>): Map<K, V> =\n LinkedHashMap(this).apply { putAll(pairs) }.optimizeReadOnlyMap()\n\n/**\n * Creates a new read-only map by replacing or adding entries to this map from another [map].\n *\n * The returned map preserves the entry iteration order of the original map.\n * Those entries of another [map] that are missing in this

```

map are iterated in the end in the order of that [map].\n *\npublic operator fun <K, V> Map<out K, V>.plus(map:
Map<out K, V>): Map<K, V> =\n    LinkedHashMap(this).apply { putAll(map) }\n\n/**\n * Appends or replaces
the given [pair] in this mutable map.\n *\n@kotlin.internal.InlineOnly\npublic inline operator fun <K, V>
MutableMap<in K, in V>.plusAssign(pair: Pair<K, V>) {\n    put(pair.first, pair.second)\n}\n\n/**\n * Appends or
replaces all pairs from the given collection of [pairs] in this mutable map.\n *\n@kotlin.internal.InlineOnly\npublic
inline operator fun <K, V> MutableMap<in K, in V>.plusAssign(pairs: Iterable<Pair<K, V>>) {\n
    putAll(pairs)\n}\n\n/**\n * Appends or replaces all pairs from the given array of [pairs] in this mutable map.\n
*\n@kotlin.internal.InlineOnly\npublic inline operator fun <K, V> MutableMap<in K, in V>.plusAssign(pairs:
Array<out Pair<K, V>>) {\n    putAll(pairs)\n}\n\n/**\n * Appends or replaces all pairs from the given sequence of
[pairs] in this mutable map.\n *\n@kotlin.internal.InlineOnly\npublic inline operator fun <K, V> MutableMap<in
K, in V>.plusAssign(pairs: Sequence<Pair<K, V>>) {\n    putAll(pairs)\n}\n\n/**\n * Appends or replaces all
entries from the given [map] in this mutable map.\n *\n@kotlin.internal.InlineOnly\npublic inline operator fun <K,
V> MutableMap<in K, in V>.plusAssign(map: Map<K, V>) {\n    putAll(map)\n}\n\n/**\n * Returns a map
containing all entries of the original map except the entry with the given [key].\n *\n * The returned map preserves
the entry iteration order of the original map.\n *\n@SinceKotlin("1.1")\npublic operator fun <K, V> Map<out K,
V>.minus(key: K): Map<K, V> =\n    this.toMutableMap().apply { minusAssign(key)
}.optimizeReadOnlyMap()\n\n/**\n * Returns a map containing all entries of the original map except those entries\n
* the keys of which are contained in the given [keys] collection.\n *\n * The returned map preserves the entry
iteration order of the original map.\n *\n@SinceKotlin("1.1")\npublic operator fun <K, V> Map<out K,
V>.minus(keys: Iterable<K>): Map<K, V> =\n    this.toMutableMap().apply { minusAssign(keys)
}.optimizeReadOnlyMap()\n\n/**\n * Returns a map containing all entries of the original map except those entries\n
* the keys of which are contained in the given [keys] array.\n *\n * The returned map preserves the entry iteration
order of the original map.\n *\n@SinceKotlin("1.1")\npublic operator fun <K, V> Map<out K, V>.minus(keys:
Array<out K>): Map<K, V> =\n    this.toMutableMap().apply { minusAssign(keys)
}.optimizeReadOnlyMap()\n\n/**\n * Returns a map containing all entries of the original map except those entries\n
* the keys of which are contained in the given [keys] sequence.\n *\n * The returned map preserves the entry
iteration order of the original map.\n *\n@SinceKotlin("1.1")\npublic operator fun <K, V> Map<out K,
V>.minus(keys: Sequence<K>): Map<K, V> =\n    this.toMutableMap().apply { minusAssign(keys)
}.optimizeReadOnlyMap()\n\n/**\n * Removes the entry with the given [key] from this mutable map.\n
*\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline operator fun <K, V> MutableMap<K,
V>.minusAssign(key: K) {\n    remove(key)\n}\n\n/**\n * Removes all entries the keys of which are contained in
the given [keys] collection from this mutable map.\n
*\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline operator fun <K, V> MutableMap<K,
V>.minusAssign(keys: Iterable<K>) {\n    this.keys.removeAll(keys)\n}\n\n/**\n * Removes all entries the keys of
which are contained in the given [keys] array from this mutable map.\n
*\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline operator fun <K, V> MutableMap<K,
V>.minusAssign(keys: Array<out K>) {\n    this.keys.removeAll(keys)\n}\n\n/**\n * Removes all entries from the
keys of which are contained in the given [keys] sequence from this mutable map.\n
*\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline operator fun <K, V> MutableMap<K,
V>.minusAssign(keys: Sequence<K>) {\n    this.keys.removeAll(keys)\n}\n\n\n// do not expose for now
@PublishedApi\ninternal fun <K, V> Map<K, V>.optimizeReadOnlyMap() = when (size) {\n    0 -> emptyMap()\n    1 -> toSingletonMapOrSelf()\n    else -> this\n}\n\n"/\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n
*\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("SetsKt")\n@file:OptIn(kotlin.experimenta
l.ExperimentalTypeInference::class)\n\npackage kotlin.collections\n\nimport kotlin.contracts.*\n\ninternal object
EmptySet : Set<Nothing>, Serializable {\n    private const val serialVersionUID: Long =
3406603774387020532\n\n    override fun equals(other: Any?): Boolean = other is Set<*> && other.isEmpty()\n

```

```

override fun hashCode(): Int = 0\n  override fun toString(): String = \"[]\"\n  override val size: Int get() = 0\n  override fun isEmpty(): Boolean = true\n  override fun contains(element: Nothing): Boolean = false\n  override fun containsAll(elements: Collection<Nothing>): Boolean = elements.isEmpty()\n  override fun iterator(): Iterator<Nothing> = EmptyIterator\n\n  private fun readResolve(): Any = EmptySet\n\n  * Returns an empty read-only set. The returned set is serializable (JVM).\n  * @sample samples.collections.Collections.Sets.emptyReadOnlySet\n\n  * public fun <T> emptySet(): Set<T> = EmptySet\n\n  * Returns a new read-only set with the given elements.\n  * Elements of the set are iterated in the order they were specified.\n  * The returned set is serializable (JVM).\n  * @sample samples.collections.Collections.Sets.readOnlySet\n\n  * public fun <T> setOf(vararg elements: T): Set<T> = if (elements.size > 0) elements.toSet() else emptySet()\n\n  * Returns an empty read-only set. The returned set is serializable (JVM).\n  * @sample samples.collections.Collections.Sets.emptyReadOnlySet\n\n  * public inline fun <T> setOf(): Set<T> = emptySet()\n\n  * Returns an empty new [MutableSet].\n  * The returned set preserves the element iteration order.\n  * @sample samples.collections.Collections.Sets.emptyMutableSet\n\n  * public inline fun <T> mutableSetOf(): MutableSet<T> = LinkedHashSet()\n\n  * Returns a new [MutableSet] with the given elements.\n  * Elements of the set are iterated in the order they were specified.\n  * @sample samples.collections.Collections.Sets.mutableSet\n\n  * public fun <T> mutableSetOf(vararg elements: T): MutableSet<T> = elements.toCollection(LinkedHashSet(mapCapacity(elements.size)))\n\n  * Returns an empty new [HashSet].\n\n  * public inline fun <T> hashSetOf(): HashSet<T> = HashSet()\n\n  * Returns a new [HashSet] with the given elements.\n\n  * public fun <T> hashSetOf(vararg elements: T): HashSet<T> = elements.toCollection(HashSet(mapCapacity(elements.size)))\n\n  * Returns an empty new [LinkedHashSet].\n  * @sample samples.collections.Collections.Sets.emptyLinkedHashSet\n\n  * public inline fun <T> linkedSetOf(): LinkedHashSet<T> = LinkedHashSet()\n\n  * Returns a new [LinkedHashSet] with the given elements.\n  * Elements of the set are iterated in the order they were specified.\n  * @sample samples.collections.Collections.Sets.linkedHashSet\n\n  * public fun <T> linkedSetOf(vararg elements: T): LinkedHashSet<T> = elements.toCollection(LinkedHashSet(mapCapacity(elements.size)))\n\n  * Returns a new read-only set either with single given element, if it is not null, or empty set if the element is null.\n  * The returned set is serializable (JVM).\n  * @sample samples.collections.Collections.Sets.setOfNotNull\n\n  * public fun <T : Any> setOfNotNull(element: T?): Set<T> = if (element != null) setOf(element) else emptySet()\n\n  * Returns a new read-only set only with those given elements, that are not null.\n  * Elements of the set are iterated in the order they were specified.\n  * The returned set is serializable (JVM).\n  * @sample samples.collections.Collections.Sets.setOfNotNull\n\n  * public fun <T : Any> setOfNotNull(vararg elements: T?): Set<T> {\n    return elements.filterNotNullTo(LinkedHashSet())\n  }\n\n  * Builds a new read-only [Set] by populating a [MutableSet] using the given [builderAction] and returning a read-only set with the same elements.\n  * The set passed as a receiver to the [builderAction] is valid only inside that function.\n  * Using it outside of the function produces an unspecified behavior.\n  * Elements of the set are iterated in the order they were added by the [builderAction].\n  * The returned set is serializable (JVM).\n  * @sample samples.collections.Builders.Sets.buildSetSample\n\n  * public inline fun <E> buildSet(@BuilderInference builderAction: MutableSet<E>.() -> Unit): Set<E> {\n    contract { callsInPlace(builderAction, InvocationKind.EXACTLY_ONCE) }\n    return buildSetInternal(builderAction)\n  }\n\n  * Builds a new read-only [Set] by populating a [MutableSet] using the given [builderAction] and returning a read-only set with the same elements.\n  * The set passed as a receiver to the [builderAction] is valid only inside that function.\n  * Using it outside of the function produces an unspecified behavior.\n  * [capacity] is used to hint the

```

```

expected number of elements added in the [builderAction].\n *\n * Elements of the set are iterated in the order they
were added by the [builderAction].\n *\n * The returned set is serializable (JVM).\n *\n * @throws
IllegalArgumentExcepTion if the given [capacity] is negative.\n *\n * @sample
samples.collections.Builders.Sets.buildSetSample\n
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
inline fun <E> buildSet(capacity: Int, @BuilderInference builderAction: MutableSet<E>.) -> Unit): Set<E> {\n
contract { callsInPlace(builderAction, InvocationKind.EXACTLY_ONCE) }\n return buildSetInternal(capacity,
builderAction)\n}\n\n@PublishedApi\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\ninternal expect inline
fun <E> buildSetInternal(capacity: Int, builderAction: MutableSet<E>.) -> Unit): Set<E>\n\n/** Returns this Set
if it's not `null` and the empty set otherwise. *\n@kotlin.internal.InlineOnly\npublic inline fun <T>
Set<T>?.orEmpty(): Set<T> = this ?: emptySet()\n\ninternal fun <T> Set<T>.optimizeReadOnlySet() = when (size)
{\n 0 -> emptySet()\n 1 -> setOf(iterator().next())\n else -> this\n}\n"/*\n * Copyright 2010-2018 JetBrains
s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0
license that can be found in the license/LICENSE.txt file.\n
*\n@n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("StringsKt")\n@file:Suppress("PLATFOR
M_CLASS_MAPPED_TO_KOTLIN")\n\npackage kotlin.text\n\n/**\n * Parses the string as a signed [Byte]
number and returns the result\n * or `null` if the string is not a valid representation of a number.\n
*\n@SinceKotlin("1.1")\npublic fun String.toByteArrayOrNull(): Byte? = toByteOrNull(radix = 10)\n\n/**\n * Parses
the string as a signed [Byte] number and returns the result\n * or `null` if the string is not a valid representation of a
number.\n *\n * @throws IllegalArgumentExcepTion when [radix] is not a valid radix for string to number
conversion.\n *\n@SinceKotlin("1.1")\npublic fun String.toByteArrayOrNull(radix: Int): Byte? {\n val int =
this.toIntOrNull(radix) ?: return null\n if (int < Byte.MIN_VALUE || int > Byte.MAX_VALUE) return null\n
return int.toByteArray()\n}\n\n/**\n * Parses the string as a [Short] number and returns the result\n * or `null` if the string
is not a valid representation of a number.\n *\n@SinceKotlin("1.1")\npublic fun String.toShortOrNull(): Short? =
toShortOrNull(radix = 10)\n\n/**\n * Parses the string as a [Short] number and returns the result\n * or `null` if the
string is not a valid representation of a number.\n *\n * @throws IllegalArgumentExcepTion when [radix] is not a
valid radix for string to number conversion.\n *\n@SinceKotlin("1.1")\npublic fun String.toShortOrNull(radix:
Int): Short? {\n val int = this.toIntOrNull(radix) ?: return null\n if (int < Short.MIN_VALUE || int >
Short.MAX_VALUE) return null\n return int.toShort()\n}\n\n/**\n * Parses the string as an [Int] number and
returns the result\n * or `null` if the string is not a valid representation of a number.\n
*\n@SinceKotlin("1.1")\npublic fun String.toIntOrNull(): Int? = toIntOrNull(radix = 10)\n\n/**\n * Parses the
string as an [Int] number and returns the result\n * or `null` if the string is not a valid representation of a number.\n
*\n * @throws IllegalArgumentExcepTion when [radix] is not a valid radix for string to number conversion.\n
*\n@SinceKotlin("1.1")\npublic fun String.toIntOrNull(radix: Int): Int? {\n checkRadix(radix)\n\n val length
= this.length\n if (length == 0) return null\n\n val start: Int\n val isNegative: Boolean\n val limit: Int\n val
firstChar = this[0]\n if (firstChar < '0') { // Possible leading sign\n if (length == 1) return null // non-digit
(possible sign) only, no digits after\n\n start = 1\n\n if (firstChar == '-') {\n isNegative = true\n
limit = Int.MIN_VALUE\n } else if (firstChar == '+') {\n isNegative = false\n limit = -
Int.MAX_VALUE\n } else\n return null\n } else {\n start = 0\n isNegative = false\n limit
= -Int.MAX_VALUE\n }\n\n val limitForMaxRadix = (-Int.MAX_VALUE) / 36\n\n var limitBeforeMul =
limitForMaxRadix\n var result = 0\n for (i in start until length) {\n val digit = digitOf(this[i], radix)\n\n
if (digit < 0) return null\n if (result < limitBeforeMul) {\n if (limitBeforeMul == limitForMaxRadix) {\n
limitBeforeMul = limit / radix\n\n if (result < limitBeforeMul) {\n return null\n
}\n } else {\n return null\n }\n }\n\n result *= radix\n\n if (result < limit + digit)
return null\n\n result -= digit\n }\n\n return if (isNegative) result else -result\n}\n\n/**\n * Parses the string
as a [Long] number and returns the result\n * or `null` if the string is not a valid representation of a number.\n
*\n@SinceKotlin("1.1")\npublic fun String.toLongOrNull(): Long? = toLongOrNull(radix = 10)\n\n/**\n * Parses
the string as a [Long] number and returns the result\n * or `null` if the string is not a valid representation of a

```

```

number.\n *\n * @throws IllegalArgumentException when [radix] is not a valid radix for string to number
conversion.\n *\n * @SinceKotlin("1.1")\npublic fun String.toLongOrNull(radix: Int): Long? {\n
checkRadix(radix)\n\n val length = this.length\n if (length == 0) return null\n\n val start: Int\n val isNegative:
Boolean\n val limit: Long\n\n val firstChar = this[0]\n if (firstChar < '0') { // Possible leading sign\n if
(length == 1) return null // non-digit (possible sign) only, no digits after\n\n start = 1\n\n if (firstChar == '-')
{\n isNegative = true\n limit = Long.MIN_VALUE\n } else if (firstChar == '+') {\n
isNegative = false\n limit = -Long.MAX_VALUE\n } else\n return null\n } else {\n start =
0\n isNegative = false\n limit = -Long.MAX_VALUE\n }\n\n\n val limitForMaxRadix = (-
Long.MAX_VALUE) / 36\n\n var limitBeforeMul = limitForMaxRadix\n var result = 0L\n for (i in start until
length) {\n val digit = digitOf(this[i], radix)\n if (digit < 0) return null\n if (result < limitBeforeMul)
{\n if (limitBeforeMul == limitForMaxRadix) {\n limitBeforeMul = limit / radix\n\n if
(result < limitBeforeMul) {\n return null\n }\n } else {\n return null\n
}\n }\n\n result *= radix\n\n if (result < limit + digit) return null\n\n result -= digit\n }\n\n
return if (isNegative) result else -result\n}\n\n\ninternal fun numberFormatError(input: String): Nothing = throw
NumberFormatException("Invalid number format: '$input')\n", "/*\n * Copyright 2010-2021 JetBrains s.r.o. and
Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.time\n\nimport kotlin.contracts.*\nimport
kotlin.jvm.JvmInline\nimport kotlin.math.*\n\n/**\n * Represents the amount of time one instant of time is away
from another instant.\n * A negative duration is possible in a situation when the second instant is earlier than the
first one.\n * The type can store duration values up to \u00b1146 years with nanosecond precision,\n * and up to
\u00b1146 million years with millisecond precision.\n * If a duration-returning operation provided in `kotlin.time`
produces a duration value that doesn't fit into the above range,\n * the returned `Duration` is infinite.\n * An
infinite duration value [Duration.INFINITE] can be used to represent infinite timeouts.\n * To construct a
duration use either the extension function [toDuration],\n * or the extension properties [hours], [minutes], [seconds],
and so on,\n * available on [Int], [Long], and [Double] numeric types.\n * To get the value of this duration
expressed in a particular [duration units][DurationUnit]\n * use the functions [toInt], [toLong], and [toDouble]\n * or
the properties [inWholeHours], [inWholeMinutes], [inWholeSeconds], [inWholeNanoseconds], and so on.\n
*/\n\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalTime::class)\n@JvmInline\npublic value class
Duration internal constructor(private val rawValue: Long) : Comparable<Duration> {\n\n private val value: Long
get() = rawValue shr 1\n\n private inline val unitDiscriminator: Int get() = rawValue.toInt() and 1\n\n private fun
isInNanos() = unitDiscriminator == 0\n\n private fun isInMillis() = unitDiscriminator == 1\n\n private val
storageUnit get() = if (isInNanos()) DurationUnit.NANOSECONDS else DurationUnit.MILLISECONDS\n\n\n init
{\n\n if (durationAssertionsEnabled) {\n\n if (isInNanos()) {\n\n if (value !in -
MAX_NANOS..MAX_NANOS) throw AssertionError("$value ns is out of nanoseconds range")\n\n } else
{\n\n if (value !in -MAX_MILLIS..MAX_MILLIS) throw AssertionError("$value ms is out of milliseconds
range")\n\n if (value in -MAX_NANOS_IN_MILLIS..MAX_NANOS_IN_MILLIS) throw
AssertionError("$value ms is denormalized")\n\n }\n\n }\n\n }\n\n companion object {\n\n /** The
duration equal to exactly 0 seconds. *\n\n public val ZERO: Duration = Duration(0L)\n\n /** The duration
whose value is positive infinity. It is useful for representing timeouts that should never expire. *\n\n public val
INFINITE: Duration = durationOfMillis(MAX_MILLIS)\n\n internal val NEG_INFINITE: Duration =
durationOfMillis(-MAX_MILLIS)\n\n /** Converts the given time duration [value] expressed in the specified
[sourceUnit] into the specified [targetUnit]. *\n\n @ExperimentalTime\n\n public fun convert(value: Double,
sourceUnit: DurationUnit, targetUnit: DurationUnit): Double =\n\n convertDurationUnit(value, sourceUnit,
targetUnit)\n\n // Duration construction extension properties in Duration companion scope\n\n /** Returns a
[Duration] equal to this [Int] number of nanoseconds. *\n\n @kotlin.internal.InlineOnly\n\n public inline val
Int.nanoseconds get() = toDuration(DurationUnit.NANOSECONDS)\n\n /** Returns a [Duration] equal to this
[Long] number of nanoseconds. *\n\n @kotlin.internal.InlineOnly\n\n public inline val Long.nanoseconds
get() = toDuration(DurationUnit.NANOSECONDS)\n\n /**\n * Returns a [Duration] equal to this

```

```

[Double] number of nanoseconds.\n      *\n      * Depending on its magnitude, the value is rounded to an integer
number of nanoseconds or milliseconds.\n      *\n      * @throws IllegalArgumentException if this [Double]
value is `NaN`.\n      *\n      @kotlin.internal.InlineOnly\n      public inline val Double.nanoseconds get() =
toDuration(DurationUnit.NANOSECONDS)\n\n      /** Returns a [Duration] equal to this [Int] number of
microseconds. *\n      @kotlin.internal.InlineOnly\n      public inline val Int.microseconds get() =
toDuration(DurationUnit.MICROSECONDS)\n\n      /** Returns a [Duration] equal to this [Long] number of
microseconds. *\n      @kotlin.internal.InlineOnly\n      public inline val Long.microseconds get() =
toDuration(DurationUnit.MICROSECONDS)\n\n      /**\n      * Returns a [Duration] equal to this [Double]
number of microseconds.\n      *\n      * Depending on its magnitude, the value is rounded to an integer number
of nanoseconds or milliseconds.\n      *\n      * @throws IllegalArgumentException if this [Double] value is
`NaN`.\n      *\n      @kotlin.internal.InlineOnly\n      public inline val Double.microseconds get() =
toDuration(DurationUnit.MICROSECONDS)\n\n      /** Returns a [Duration] equal to this [Int] number of
milliseconds. *\n      @kotlin.internal.InlineOnly\n      public inline val Int.milliseconds get() =
toDuration(DurationUnit.MILLISECONDS)\n\n      /** Returns a [Duration] equal to this [Long] number of
milliseconds. *\n      @kotlin.internal.InlineOnly\n      public inline val Long.milliseconds get() =
toDuration(DurationUnit.MILLISECONDS)\n\n      /**\n      * Returns a [Duration] equal to this [Double]
number of milliseconds.\n      *\n      * Depending on its magnitude, the value is rounded to an integer number of
nanoseconds or milliseconds.\n      *\n      * @throws IllegalArgumentException if this [Double] value is
`NaN`.\n      *\n      @kotlin.internal.InlineOnly\n      public inline val Double.milliseconds get() =
toDuration(DurationUnit.MILLISECONDS)\n\n      /** Returns a [Duration] equal to this [Int] number of
seconds. *\n      @kotlin.internal.InlineOnly\n      public inline val Int.seconds get() =
toDuration(DurationUnit.SECONDS)\n\n      /** Returns a [Duration] equal to this [Long] number of seconds. *\n      @kotlin.internal.InlineOnly\n      public inline val Long.seconds get() =
toDuration(DurationUnit.SECONDS)\n\n      /**\n      * Returns a [Duration] equal to this [Double] number of
seconds.\n      *\n      * Depending on its magnitude, the value is rounded to an integer number of nanoseconds or
milliseconds.\n      *\n      * @throws IllegalArgumentException if this [Double] value is `NaN`.\n      *\n      @kotlin.internal.InlineOnly\n      public inline val Double.seconds get() =
toDuration(DurationUnit.SECONDS)\n\n      /** Returns a [Duration] equal to this [Int] number of minutes. *\n      @kotlin.internal.InlineOnly\n      public inline val Int.minutes get() = toDuration(DurationUnit.MINUTES)\n\n      /** Returns a [Duration] equal to this [Long] number of minutes. *\n      @kotlin.internal.InlineOnly\n      public inline val Long.minutes get() = toDuration(DurationUnit.MINUTES)\n\n      /**\n      * Returns a
[Duration] equal to this [Double] number of minutes.\n      *\n      * Depending on its magnitude, the value is
rounded to an integer number of nanoseconds or milliseconds.\n      *\n      * @throws IllegalArgumentException
if this [Double] value is `NaN`.\n      *\n      @kotlin.internal.InlineOnly\n      public inline val Double.minutes
get() = toDuration(DurationUnit.MINUTES)\n\n      /** Returns a [Duration] equal to this [Int] number of hours.
*\n      @kotlin.internal.InlineOnly\n      public inline val Int.hours get() = toDuration(DurationUnit.HOURS)\n\n      /** Returns a [Duration] equal to this [Long] number of hours. *\n      @kotlin.internal.InlineOnly\n      public
inline val Long.hours get() = toDuration(DurationUnit.HOURS)\n\n      /**\n      * Returns a [Duration] equal to
this [Double] number of hours.\n      *\n      * Depending on its magnitude, the value is rounded to an integer
number of nanoseconds or milliseconds.\n      *\n      * @throws IllegalArgumentException if this [Double]
value is `NaN`.\n      *\n      @kotlin.internal.InlineOnly\n      public inline val Double.hours get() =
toDuration(DurationUnit.HOURS)\n\n      /** Returns a [Duration] equal to this [Int] number of days. *\n      @kotlin.internal.InlineOnly\n      public inline val Int.days get() = toDuration(DurationUnit.DAYS)\n\n      /**
Returns a [Duration] equal to this [Long] number of days. *\n      @kotlin.internal.InlineOnly\n      public inline
val Long.days get() = toDuration(DurationUnit.DAYS)\n\n      /**\n      * Returns a [Duration] equal to this
[Double] number of days.\n      *\n      * Depending on its magnitude, the value is rounded to an integer number
of nanoseconds or milliseconds.\n      *\n      * @throws IllegalArgumentException if this [Double] value is
`NaN`.\n      *\n      @kotlin.internal.InlineOnly\n      public inline val Double.days get() =

```

```

toDuration(DurationUnit.DAYS)\n\n    // deprecated static factory functions\n\n    /** Returns a [Duration]
representing the specified [value] number of nanoseconds. */\n    @SinceKotlin("1.5")\n
@ExperimentalTime\n    @Deprecated("Use 'Int.nanoseconds' extension property from Duration.Companion
instead.", ReplaceWith("value.nanoseconds", "kotlin.time.Duration.Companion.nanoseconds"))\n
@DeprecatedSinceKotlin(warningSince = "1.6")\n    public fun nanoseconds(value: Int): Duration =
value.toDuration(DurationUnit.NANOSECONDS)\n\n    /** Returns a [Duration] representing the specified
[value] number of nanoseconds. */\n    @SinceKotlin("1.5")\n    @ExperimentalTime\n
@Deprecated("Use 'Long.nanoseconds' extension property from Duration.Companion instead.",
ReplaceWith("value.nanoseconds", "kotlin.time.Duration.Companion.nanoseconds"))\n
@DeprecatedSinceKotlin(warningSince = "1.6")\n    public fun nanoseconds(value: Long): Duration =
value.toDuration(DurationUnit.NANOSECONDS)\n\n    /**\n    * Returns a [Duration] representing the
specified [value] number of nanoseconds.\n    *\n    * @throws IllegalArgumentException if the provided
`Double` [value] is `NaN`.\n    */\n    @SinceKotlin("1.5")\n    @ExperimentalTime\n
@Deprecated("Use 'Double.nanoseconds' extension property from Duration.Companion instead.",
ReplaceWith("value.nanoseconds", "kotlin.time.Duration.Companion.nanoseconds"))\n
@DeprecatedSinceKotlin(warningSince = "1.6")\n    public fun nanoseconds(value: Double): Duration =
value.toDuration(DurationUnit.NANOSECONDS)\n\n\n    /** Returns a [Duration] representing the specified
[value] number of microseconds. */\n    @SinceKotlin("1.5")\n    @ExperimentalTime\n
@Deprecated("Use 'Int.microseconds' extension property from Duration.Companion instead.",
ReplaceWith("value.microseconds", "kotlin.time.Duration.Companion.microseconds"))\n
@DeprecatedSinceKotlin(warningSince = "1.6")\n    public fun microseconds(value: Int): Duration =
value.toDuration(DurationUnit.MICROSECONDS)\n\n    /** Returns a [Duration] representing the specified
[value] number of microseconds. */\n    @SinceKotlin("1.5")\n    @ExperimentalTime\n
@Deprecated("Use 'Long.microseconds' extension property from Duration.Companion instead.",
ReplaceWith("value.microseconds", "kotlin.time.Duration.Companion.microseconds"))\n
@DeprecatedSinceKotlin(warningSince = "1.6")\n    public fun microseconds(value: Long): Duration =
value.toDuration(DurationUnit.MICROSECONDS)\n\n    /**\n    * Returns a [Duration] representing the
specified [value] number of microseconds.\n    *\n    * @throws IllegalArgumentException if the provided
`Double` [value] is `NaN`.\n    */\n    @SinceKotlin("1.5")\n    @ExperimentalTime\n
@Deprecated("Use 'Double.microseconds' extension property from Duration.Companion instead.",
ReplaceWith("value.microseconds", "kotlin.time.Duration.Companion.microseconds"))\n
@DeprecatedSinceKotlin(warningSince = "1.6")\n    public fun microseconds(value: Double): Duration =
value.toDuration(DurationUnit.MICROSECONDS)\n\n\n    /** Returns a [Duration] representing the specified
[value] number of milliseconds. */\n    @SinceKotlin("1.5")\n    @ExperimentalTime\n
@Deprecated("Use 'Int.milliseconds' extension property from Duration.Companion instead.",
ReplaceWith("value.milliseconds", "kotlin.time.Duration.Companion.milliseconds"))\n
@DeprecatedSinceKotlin(warningSince = "1.6")\n    public fun milliseconds(value: Int): Duration =
value.toDuration(DurationUnit.MILLISECONDS)\n\n    /** Returns a [Duration] representing the specified
[value] number of milliseconds. */\n    @SinceKotlin("1.5")\n    @ExperimentalTime\n
@Deprecated("Use 'Long.milliseconds' extension property from Duration.Companion instead.",
ReplaceWith("value.milliseconds", "kotlin.time.Duration.Companion.milliseconds"))\n
@DeprecatedSinceKotlin(warningSince = "1.6")\n    public fun milliseconds(value: Long): Duration =
value.toDuration(DurationUnit.MILLISECONDS)\n\n    /**\n    * Returns a [Duration] representing the
specified [value] number of milliseconds.\n    *\n    * @throws IllegalArgumentException if the provided
`Double` [value] is `NaN`.\n    */\n    @SinceKotlin("1.5")\n    @ExperimentalTime\n
@Deprecated("Use 'Double.milliseconds' extension property from Duration.Companion instead.",
ReplaceWith("value.milliseconds", "kotlin.time.Duration.Companion.milliseconds"))\n
@DeprecatedSinceKotlin(warningSince = "1.6")\n    public fun milliseconds(value: Double): Duration =

```

```

value.toDuration(DurationUnit.MILLISECONDS)\n\n    /** Returns a [Duration] representing the specified
[value] number of seconds. */\n    @SinceKotlin("1.5")\n    @ExperimentalTime\n    @Deprecated("Use
'Int.seconds' extension property from Duration.Companion instead.", ReplaceWith("value.seconds",
"kotlin.time.Duration.Companion.seconds"))\n    @DeprecatedSinceKotlin(warningSince = "1.6")\n    public
fun seconds(value: Int): Duration = value.toDuration(DurationUnit.SECONDS)\n\n    /** Returns a [Duration]
representing the specified [value] number of seconds. */\n    @SinceKotlin("1.5")\n    @ExperimentalTime\n
    @Deprecated("Use 'Long.seconds' extension property from Duration.Companion instead.",
ReplaceWith("value.seconds", "kotlin.time.Duration.Companion.seconds"))\n
    @DeprecatedSinceKotlin(warningSince = "1.6")\n    public fun seconds(value: Long): Duration =
value.toDuration(DurationUnit.SECONDS)\n\n    /**\n    * Returns a [Duration] representing the specified
[value] number of seconds.\n    *\n    * @throws IllegalArgumentException if the provided `Double` [value] is
`NaN`.\n    */\n    @SinceKotlin("1.5")\n    @ExperimentalTime\n    @Deprecated("Use
'Double.seconds' extension property from Duration.Companion instead.", ReplaceWith("value.seconds",
"kotlin.time.Duration.Companion.seconds"))\n    @DeprecatedSinceKotlin(warningSince = "1.6")\n    public
fun seconds(value: Double): Duration = value.toDuration(DurationUnit.SECONDS)\n\n    /** Returns a
[Duration] representing the specified [value] number of minutes. */\n    @SinceKotlin("1.5")\n
    @ExperimentalTime\n    @Deprecated("Use 'Int.minutes' extension property from Duration.Companion
instead.", ReplaceWith("value.minutes", "kotlin.time.Duration.Companion.minutes"))\n
    @DeprecatedSinceKotlin(warningSince = "1.6")\n    public fun minutes(value: Int): Duration =
value.toDuration(DurationUnit.MINUTES)\n\n    /** Returns a [Duration] representing the specified [value]
number of minutes. */\n    @SinceKotlin("1.5")\n    @ExperimentalTime\n    @Deprecated("Use
'Long.minutes' extension property from Duration.Companion instead.", ReplaceWith("value.minutes",
"kotlin.time.Duration.Companion.minutes"))\n    @DeprecatedSinceKotlin(warningSince = "1.6")\n    public
fun minutes(value: Long): Duration = value.toDuration(DurationUnit.MINUTES)\n\n    /**\n    * Returns a
[Duration] representing the specified [value] number of minutes.\n    *\n    * @throws
IllegalArgumentException if the provided `Double` [value] is `NaN`.\n    */\n    @SinceKotlin("1.5")\n
    @ExperimentalTime\n    @Deprecated("Use 'Double.minutes' extension property from Duration.Companion
instead.", ReplaceWith("value.minutes", "kotlin.time.Duration.Companion.minutes"))\n
    @DeprecatedSinceKotlin(warningSince = "1.6")\n    public fun minutes(value: Double): Duration =
value.toDuration(DurationUnit.MINUTES)\n\n    /** Returns a [Duration] representing the specified [value]
number of hours. */\n    @SinceKotlin("1.5")\n    @ExperimentalTime\n    @Deprecated("Use 'Int.hours'
extension property from Duration.Companion instead.", ReplaceWith("value.hours",
"kotlin.time.Duration.Companion.hours"))\n    @DeprecatedSinceKotlin(warningSince = "1.6")\n    public
fun hours(value: Int): Duration = value.toDuration(DurationUnit.HOURS)\n\n    /** Returns a [Duration]
representing the specified [value] number of hours. */\n    @SinceKotlin("1.5")\n    @ExperimentalTime\n
    @Deprecated("Use 'Long.hours' extension property from Duration.Companion instead.",
ReplaceWith("value.hours", "kotlin.time.Duration.Companion.hours"))\n
    @DeprecatedSinceKotlin(warningSince = "1.6")\n    public fun hours(value: Long): Duration =
value.toDuration(DurationUnit.HOURS)\n\n    /**\n    * Returns a [Duration] representing the specified
[value] number of hours.\n    *\n    * @throws IllegalArgumentException if the provided `Double` [value] is
`NaN`.\n    */\n    @SinceKotlin("1.5")\n    @ExperimentalTime\n    @Deprecated("Use 'Double.hours'
extension property from Duration.Companion instead.", ReplaceWith("value.hours",
"kotlin.time.Duration.Companion.hours"))\n    @DeprecatedSinceKotlin(warningSince = "1.6")\n    public
fun hours(value: Double): Duration = value.toDuration(DurationUnit.HOURS)\n\n    /** Returns a [Duration]
representing the specified [value] number of days. */\n    @SinceKotlin("1.5")\n    @ExperimentalTime\n
    @Deprecated("Use 'Int.days' extension property from Duration.Companion instead.", ReplaceWith("value.days",
"kotlin.time.Duration.Companion.days"))\n    @DeprecatedSinceKotlin(warningSince = "1.6")\n    public
fun days(value: Int): Duration = value.toDuration(DurationUnit.DAYS)\n\n    /** Returns a [Duration]

```

```

representing the specified [value] number of days. */
@SinceKotlin("1.5")
@ExperimentalTime
@Deprecated("Use 'Long.days' extension property from Duration.Companion instead.",
ReplaceWith("value.days", "kotlin.time.Duration.Companion.days"))
@DeprecatedSinceKotlin(warningSince = "1.6")
public fun days(value: Long): Duration =
value.toDuration(DurationUnit.DAYS)
/**
 * Returns a [Duration] representing the specified [value]
number of days.
 *
 * @throws IllegalArgumentException if the provided `Double` [value] is `NaN`.
 */
@SinceKotlin("1.5")
@ExperimentalTime
@Deprecated("Use 'Double.days' extension
property from Duration.Companion instead.", ReplaceWith("value.days",
"kotlin.time.Duration.Companion.days"))
@DeprecatedSinceKotlin(warningSince = "1.6")
public
fun days(value: Double): Duration = value.toDuration(DurationUnit.DAYS)
/**
 * Parses a string that
represents a duration and returns the parsed [Duration] value.
 *
 * The following formats are
accepted:
 *
 * - ISO-8601 Duration format, e.g. `P1DT2H3M4.058S`, see [toIsoString] and
[parseIsoString].
 *
 * - The format of string returned by the default [Duration.toString] and `toString` in a
specific unit,
 *
 * e.g. `10s`, `1h 30m` or `-(1h 30m)`.
 *
 * @throws IllegalArgumentException if
the string doesn't represent a duration in any of the supported formats.
 *
 * @sample
samples.time.Durations.parse
 */
public fun parse(value: String): Duration = try {
parseDuration(value, strictIso = false)
} catch (e: IllegalArgumentException) {
throw
IllegalArgumentException("Invalid duration string format: '$value'.", e)
}
/**
 * Parses a
string that represents a duration in ISO-8601 format and returns the parsed [Duration] value.
 *
 *
 * @throws IllegalArgumentException if the string doesn't represent a duration in ISO-8601 format.
 *
 * @sample
samples.time.Durations.parseIsoString
 */
public fun parseIsoString(value: String): Duration = try {
parseDuration(value, strictIso = true)
} catch (e: IllegalArgumentException) {
throw
IllegalArgumentException("Invalid ISO duration string format: '$value'.", e)
}
/**
 * Parses a
string that represents a duration and returns the parsed [Duration] value,
 *
 * or `null` if the string doesn't
represent a duration in any of the supported formats.
 *
 * The following formats are accepted:
 *
 *
 * - ISO-8601 Duration format, e.g. `P1DT2H3M4.058S`, see [toIsoString] and [parseIsoString].
 *
 * -
The format of string returned by the default [Duration.toString] and `toString` in a specific unit,
 *
 * e.g. `10s`,
`1h 30m` or `-(1h 30m)`.
 *
 * @sample samples.time.Durations.parse
 */
public fun
parseOrNull(value: String): Duration? = try {
parseDuration(value, strictIso = false)
} catch (e:
IllegalArgumentException) {
null
}
/**
 * Parses a string that represents a duration in
ISO-8601 format and returns the parsed [Duration] value,
 *
 * or `null` if the string doesn't represent a duration
in ISO-8601 format.
 *
 * @sample samples.time.Durations.parseIsoString
 */
public fun
parseIsoStringOrNull(value: String): Duration? = try {
parseDuration(value, strictIso = true)
} catch
(e: IllegalArgumentException) {
null
}
}
}
// arithmetic operators
/** Returns the
negative of this value.
 */
public operator fun unaryMinus(): Duration = durationOf(-value,
unitDiscriminator)
/**
 * Returns a duration whose value is the sum of this and [other] duration values.
 *
 * @throws IllegalArgumentException if the operation results in an undefined value for the given arguments,
 *
 * e.g. when adding infinite durations of different sign.
 */
public operator fun plus(other: Duration):
Duration {
when {
this.isInfinite() -> {
if (other.isFinite() || (this.rawValue xor
other.rawValue >= 0))
return this
else
throw
IllegalArgumentException("Summing infinite durations of different signs yields an undefined result.")
}
other.isInfinite() -> return other
}
return when {
this.unitDiscriminator ==
other.unitDiscriminator -> {
val result = this.value + other.value // never overflows long, but can
overflow long63
when {
isInNanos() ->
durationOfNanosNormalized(result)
else ->
durationOfMillisNormalized(result)
}
}
this.isInMillis() ->
addValuesMixedRanges(this.value, other.value)
else ->
addValuesMixedRanges(other.value, this.value)
}
}
}
private fun
addValuesMixedRanges(thisMillis: Long, otherNanos: Long): Duration {
val otherMillis =

```

```

nanosToMillis(otherNanos)\n    val resultMillis = thisMillis + otherMillis\n    return if (resultMillis in -
MAX_NANOS_IN_MILLIS..MAX_NANOS_IN_MILLIS) {\n        val otherNanoRemainder = otherNanos -
millisToNanos(otherMillis)\n        durationOfNanos(millisToNanos(resultMillis) + otherNanoRemainder)\n    }
else {\n        durationOfMillis(resultMillis.coerceIn(-MAX_MILLIS, MAX_MILLIS))\n    }\n}\n\n/**\n * Returns a duration whose value is the difference between this and [other] duration values.\n * \n * @throws
IllegalArgumentException if the operation results in an undefined value for the given arguments,\n * e.g. when
subtracting infinite durations of the same sign.\n * \n public operator fun minus(other: Duration): Duration =
this + (-other)\n\n /**\n * Returns a duration whose value is this duration value multiplied by the given [scale]
number.\n * \n * @throws IllegalArgumentException if the operation results in an undefined value for the given
arguments,\n * e.g. when multiplying an infinite duration by zero.\n * \n public operator fun times(scale: Int):
Duration {\n    if (isInfinite()) {\n        return when {\n            scale == 0 -> throw
IllegalArgumentException("Multiplying infinite duration by zero yields an undefined result.")\n            scale > 0
-> this\n            else -> -this\n        }\n    }\n    if (scale == 0) return ZERO\n    val value = value\n    val result = value * scale\n    return if (isInNanos()) {\n        if (value in (MAX_NANOS /
Int.MIN_VALUE)..(-MAX_NANOS / Int.MIN_VALUE)) {\n            // can't overflow nanos range for any
scale\n            durationOfNanos(result)\n        } else {\n            if (result / scale == value) {\n
durationOfNanosNormalized(result)\n            } else {\n                val millis = nanosToMillis(value)\n
val remNanos = value - millisToNanos(millis)\n                val resultMillis = millis * scale\n                val
totalMillis = resultMillis + nanosToMillis(remNanos * scale)\n                if (resultMillis / scale == millis &&
totalMillis xor resultMillis >= 0) {\n                    durationOfMillis(totalMillis.coerceIn(-
MAX_MILLIS..MAX_MILLIS))\n                } else {\n                    if (value.sign * scale.sign > 0) INFINITE
else NEG_INFINITE\n                }\n            }\n        } else {\n            if (result / scale == value) {\n
durationOfMillis(result.coerceIn(-MAX_MILLIS..MAX_MILLIS))\n            } else {\n                if (value.sign
* scale.sign > 0) INFINITE else NEG_INFINITE\n            }\n        }\n    }\n}\n\n /**\n * Returns a duration whose
value is this duration value multiplied by the given [scale] number.\n * \n * The operation may involve rounding
when the result cannot be represented exactly with a [Double] number.\n * \n * @throws
IllegalArgumentException if the operation results in an undefined value for the given arguments,\n * e.g. when
multiplying an infinite duration by zero.\n * \n public operator fun times(scale: Double): Duration {\n    val
intScale = scale.roundToInt()\n    if (intScale.toDouble() == scale) {\n        return times(intScale)\n    }\n    val
unit = storageUnit\n    val result = toDouble(unit) * scale\n    return result.toDuration(unit)\n}\n\n /**\n * Returns a duration whose value is this duration value divided by the given [scale] number.\n * \n *
@throws IllegalArgumentException if the operation results in an undefined value for the given arguments,\n *
e.g. when dividing zero duration by zero.\n * \n public operator fun div(scale: Int): Duration {\n    if (scale ==
0) {\n        return when {\n            isPositive() -> INFINITE\n            isNegative() -> NEG_INFINITE\n
            else -> throw IllegalArgumentException("Dividing zero duration by zero yields an undefined result.")\n
        }\n    }\n    if (isInNanos()) {\n        return durationOfNanos(value / scale)\n    } else {\n        if
(isInfinite())\n            return this * scale.sign\n        val result = value / scale\n        if (result in -
MAX_NANOS_IN_MILLIS..MAX_NANOS_IN_MILLIS) {\n            val rem = millisToNanos(value - (result *
scale)) / scale\n            return durationOfNanos(millisToNanos(result) + rem)\n        }\n        return
durationOfMillis(result)\n    }\n}\n\n /**\n * Returns a duration whose value is this duration value divided
by the given [scale] number.\n * \n * @throws IllegalArgumentException if the operation results in an
undefined value for the given arguments,\n * e.g. when dividing an infinite duration by infinity or zero duration
by zero.\n * \n public operator fun div(scale: Double): Duration {\n    val intScale = scale.roundToInt()\n
if (intScale.toDouble() == scale && intScale != 0) {\n        return div(intScale)\n    }\n    val unit =
storageUnit\n    val result = toDouble(unit) / scale\n    return result.toDuration(unit)\n}\n\n /** Returns a
number that is the ratio of this and [other] duration values. \n public operator fun div(other: Duration): Double
{\n    val coarserUnit = maxOf(this.storageUnit, other.storageUnit)\n    return this.toDouble(coarserUnit) /
other.toDouble(coarserUnit)\n}\n\n /** Returns true, if the duration value is less than zero. \n public fun

```

```

isNegative(): Boolean = rawValue < 0\n\n /** Returns true, if the duration value is greater than zero. */\n public
fun isPositive(): Boolean = rawValue > 0\n\n /** Returns true, if the duration value is infinite. */\n public fun
isInfinite(): Boolean = rawValue == INFINITE.rawValue || rawValue == NEG_INFINITE.rawValue\n\n /**
Returns true, if the duration value is finite. */\n public fun isFinite(): Boolean = !isInfinite()\n\n /** Returns the
absolute value of this value. The returned value is always non-negative. */\n public val absoluteValue: Duration
get() = if (isNegative()) -this else this\n\n override fun compareTo(other: Duration): Int {\n    val compareBits =
this.rawValue xor other.rawValue\n    if (compareBits < 0 || compareBits.toInt() and 1 == 0) // different signs or
same sign/same range\n        return this.rawValue.compareTo(other.rawValue)\n    // same sign/different
ranges\n    val r = this.unitDiscriminator - other.unitDiscriminator // compare ranges\n    return if (isNegative())
-r else r\n } \n\n // splitting to components\n\n /**\n * Splits this duration into days, hours, minutes,
seconds, and nanoseconds and executes the given [action] with these components.\n * The result of [action] is
returned as the result of this function.\n * \n * - `nanoseconds` represents the whole number of nanoseconds in
this duration, and its absolute value is less than 1_000_000_000;\n * - `seconds` represents the whole number of
seconds in this duration, and its absolute value is less than 60;\n * - `minutes` represents the whole number of
minutes in this duration, and its absolute value is less than 60;\n * - `hours` represents the whole number of hours
in this duration, and its absolute value is less than 24;\n * - `days` represents the whole number of days in this
duration.\n * \n * Infinite durations are represented as either [Long.MAX_VALUE] days, or
[Long.MIN_VALUE] days (depending on the sign of infinity),\n * and zeroes in the lower components.\n */\n
public inline fun <T> toComponents(action: (days: Long, hours: Int, minutes: Int, seconds: Int, nanoseconds: Int) -
> T): T {\n    contract { callsInPlace(action, InvocationKind.EXACTLY_ONCE) }\n    return
action(inWholeDays, hoursComponent, minutesComponent, secondsComponent, nanosecondsComponent)\n } \n\n
/**\n * Splits this duration into hours, minutes, seconds, and nanoseconds and executes the given [action] with
these components.\n * The result of [action] is returned as the result of this function.\n * \n * - `nanoseconds`
represents the whole number of nanoseconds in this duration, and its absolute value is less than 1_000_000_000;\n
* - `seconds` represents the whole number of seconds in this duration, and its absolute value is less than 60;\n
* - `minutes` represents the whole number of minutes in this duration, and its absolute value is less than 60;\n
* - `hours` represents the whole number of hours in this duration.\n * \n * Infinite durations are represented as
either [Long.MAX_VALUE] hours, or [Long.MIN_VALUE] hours (depending on the sign of infinity),\n * and
zeroes in the lower components.\n */\n public inline fun <T> toComponents(action: (hours: Long, minutes: Int,
seconds: Int, nanoseconds: Int) -> T): T {\n    contract { callsInPlace(action, InvocationKind.EXACTLY_ONCE)
}\n    return action(inWholeHours, minutesComponent, secondsComponent, nanosecondsComponent)\n } \n\n
/**\n * Splits this duration into minutes, seconds, and nanoseconds and executes the given [action] with these
components.\n * The result of [action] is returned as the result of this function.\n * \n * - `nanoseconds`
represents the whole number of nanoseconds in this duration, and its absolute value is less than 1_000_000_000;\n
* - `seconds` represents the whole number of seconds in this duration, and its absolute value is less than 60;\n
* - `minutes` represents the whole number of minutes in this duration.\n * \n * Infinite durations are represented
as either [Long.MAX_VALUE] minutes, or [Long.MIN_VALUE] minutes (depending on the sign of infinity),\n *
and zeroes in the lower components.\n */\n public inline fun <T> toComponents(action: (minutes: Long,
seconds: Int, nanoseconds: Int) -> T): T {\n    contract { callsInPlace(action, InvocationKind.EXACTLY_ONCE)
}\n    return action(inWholeMinutes, secondsComponent, nanosecondsComponent)\n } \n\n /**\n * Splits
this duration into seconds, and nanoseconds and executes the given [action] with these components.\n * The result
of [action] is returned as the result of this function.\n * \n * - `nanoseconds` represents the whole number of
nanoseconds in this duration, and its absolute value is less than 1_000_000_000;\n * - `seconds` represents the
whole number of seconds in this duration.\n * \n * Infinite durations are represented as either
[Long.MAX_VALUE] seconds, or [Long.MIN_VALUE] seconds (depending on the sign of infinity),\n * and
zero nanoseconds.\n */\n public inline fun <T> toComponents(action: (seconds: Long, nanoseconds: Int) -> T):
T {\n    contract { callsInPlace(action, InvocationKind.EXACTLY_ONCE) }\n    return
action(inWholeSeconds, nanosecondsComponent)\n } \n\n @PublishedApi\n internal val hoursComponent:

```

```

Int get() = if (isInfinite()) 0 else (inWholeHours % 24).toInt()
@PublishedApi internal val minutesComponent: Int get() = if (isInfinite()) 0 else (inWholeMinutes % 60).toInt()
@PublishedApi internal val secondsComponent: Int get() = if (isInfinite()) 0 else (inWholeSeconds % 60).toInt()
@PublishedApi internal val nanosecondsComponent: Int get() = when {
    isInfinite() -> 0
    isInMillis() -> millisToNanos(value % 1_000).toInt()
    else -> (value % 1_000_000_000).toInt()
}
// conversion to units
/** Returns the value of this duration expressed as a [Double] number of
the specified [unit].
* The operation may involve rounding when the result cannot be represented exactly
with a [Double] number.
* An infinite duration value is converted either to
[Double.POSITIVE_INFINITY] or [Double.NEGATIVE_INFINITY] depending on its sign.
*/
public fun toDouble(unit: DurationUnit): Double {
    return when (rawValue) {
        INFINITE.rawValue -> Double.POSITIVE_INFINITY
        NEG_INFINITE.rawValue -> Double.NEGATIVE_INFINITY
        else -> {
            // TODO: whether it's ok to convert to Double before scaling
            convertDurationUnit(value.toDouble(), storageUnit, unit)
        }
    }
}
/** Returns the value
of this duration expressed as a [Long] number of the specified [unit].
* If the result doesn't fit in the range
of [Long] type, it is coerced into that range:
* - [Long.MIN_VALUE] is returned if it's less than
`Long.MIN_VALUE`,
* - [Long.MAX_VALUE] is returned if it's greater than `Long.MAX_VALUE`.
* An infinite duration value is converted either to [Long.MAX_VALUE] or [Long.MIN_VALUE] depending on
its sign.
*/
public fun toLong(unit: DurationUnit): Long {
    return when (rawValue) {
        INFINITE.rawValue -> Long.MAX_VALUE
        NEG_INFINITE.rawValue -> Long.MIN_VALUE
        else -> convertDurationUnit(value, storageUnit, unit)
    }
}
/** Returns the value of this
duration expressed as an [Int] number of the specified [unit].
* If the result doesn't fit in the range of [Int]
type, it is coerced into that range:
* - [Int.MIN_VALUE] is returned if it's less than `Int.MIN_VALUE`,
* - [Int.MAX_VALUE] is returned if it's greater than `Int.MAX_VALUE`.
* An infinite duration value is
converted either to [Int.MAX_VALUE] or [Int.MIN_VALUE] depending on its sign.
*/
public fun toInt(unit: DurationUnit): Int =
    toLong(unit).coerceIn(Int.MIN_VALUE.toLong(),
Int.MAX_VALUE.toLong()).toInt()
/** The value of this duration expressed as a [Double] number of days.
*/
@ExperimentalTime @Deprecated("Use inWholeDays property instead or convert toDouble(DAYS) if a
double value is required.", ReplaceWith("toDouble(DurationUnit.DAYS)"))
public val inDays: Double get() =
toDouble(DurationUnit.DAYS)
/** The value of this duration expressed as a [Double] number of hours.
*/
@ExperimentalTime @Deprecated("Use inWholeHours property instead or convert toDouble(HOURS) if a
double value is required.", ReplaceWith("toDouble(DurationUnit.HOURS)"))
public val inHours: Double
get() = toDouble(DurationUnit.HOURS)
/** The value of this duration expressed as a [Double] number of
minutes.
*/
@ExperimentalTime @Deprecated("Use inWholeMinutes property instead or convert
toDouble(MINUTES) if a double value is required.", ReplaceWith("toDouble(DurationUnit.MINUTES)"))
public val inMinutes: Double get() = toDouble(DurationUnit.MINUTES)
/** The value of this duration
expressed as a [Double] number of seconds.
*/
@ExperimentalTime @Deprecated("Use inWholeSeconds
property instead or convert toDouble(SECONDS) if a double value is required.",
ReplaceWith("toDouble(DurationUnit.SECONDS)"))
public val inSeconds: Double get() =
toDouble(DurationUnit.SECONDS)
/** The value of this duration expressed as a [Double] number of
milliseconds.
*/
@ExperimentalTime @Deprecated("Use inWholeMilliseconds property instead or convert
toDouble(MILLISECONDS) if a double value is required.",
ReplaceWith("toDouble(DurationUnit.MILLISECONDS)"))
public val inMilliseconds: Double get() =
toDouble(DurationUnit.MILLISECONDS)
/** The value of this duration expressed as a [Double] number of
microseconds.
*/
@ExperimentalTime @Deprecated("Use inWholeMicroseconds property instead or
convert toDouble(MICROSECONDS) if a double value is required.",
ReplaceWith("toDouble(DurationUnit.MICROSECONDS)"))
public val inMicroseconds: Double get() =
toDouble(DurationUnit.MICROSECONDS)
/** The value of this duration expressed as a [Double] number of
nanoseconds.
*/
@ExperimentalTime @Deprecated("Use inWholeNanoseconds property instead or convert

```

```

toDouble(NANOSECONDS) if a double value is required.",
ReplaceWith("\toDouble(DurationUnit.NANOSECONDS)")\n public val inNanoseconds: Double get() =
toDouble(DurationUnit.NANOSECONDS)\n\n /**\n * The value of this duration expressed as a [Long]
number of days.\n *\n * An infinite duration value is converted either to [Long.MAX_VALUE] or
[Long.MIN_VALUE] depending on its sign.\n */\n public val inWholeDays: Long\n get() =
toLong(DurationUnit.DAYS)\n\n /**\n * The value of this duration expressed as a [Long] number of hours.\n
*\n * An infinite duration value is converted either to [Long.MAX_VALUE] or [Long.MIN_VALUE] depending
on its sign.\n */\n public val inWholeHours: Long\n get() = toLong(DurationUnit.HOURS)\n\n /**\n *
The value of this duration expressed as a [Long] number of minutes.\n *\n * An infinite duration value is
converted either to [Long.MAX_VALUE] or [Long.MIN_VALUE] depending on its sign.\n */\n public val
inWholeMinutes: Long\n get() = toLong(DurationUnit.MINUTES)\n\n /**\n * The value of this duration
expressed as a [Long] number of seconds.\n *\n * An infinite duration value is converted either to
[Long.MAX_VALUE] or [Long.MIN_VALUE] depending on its sign.\n */\n public val inWholeSeconds:
Long\n get() = toLong(DurationUnit.SECONDS)\n\n /**\n * The value of this duration expressed as a
[Long] number of milliseconds.\n *\n * An infinite duration value is converted either to [Long.MAX_VALUE]
or [Long.MIN_VALUE] depending on its sign.\n */\n public val inWholeMilliseconds: Long\n get() {\n
return if (isInMillis() && isFinite()) value else toLong(DurationUnit.MILLISECONDS)\n }\n\n /**\n *
The value of this duration expressed as a [Long] number of microseconds.\n *\n * If the result doesn't fit in the
range of [Long] type, it is coerced into that range:\n * - [Long.MIN_VALUE] is returned if it's less than
`Long.MIN_VALUE`,\n * - [Long.MAX_VALUE] is returned if it's greater than `Long.MAX_VALUE`.\n *\n
* An infinite duration value is converted either to [Long.MAX_VALUE] or [Long.MIN_VALUE] depending on
its sign.\n */\n public val inWholeMicroseconds: Long\n get() =
toLong(DurationUnit.MICROSECONDS)\n\n /**\n * The value of this duration expressed as a [Long] number
of nanoseconds.\n *\n * If the result doesn't fit in the range of [Long] type, it is coerced into that range:\n * -
[Long.MIN_VALUE] is returned if it's less than `Long.MIN_VALUE`,\n * - [Long.MAX_VALUE] is returned if
it's greater than `Long.MAX_VALUE`.\n *\n * An infinite duration value is converted either to
[Long.MAX_VALUE] or [Long.MIN_VALUE] depending on its sign.\n */\n public val inWholeNanoseconds:
Long\n get() {\n val value = value\n return when {\n isInNanos() -> value\n
value > Long.MAX_VALUE / NANOS_IN_MILLIS -> Long.MAX_VALUE\n value <
Long.MIN_VALUE / NANOS_IN_MILLIS -> Long.MIN_VALUE\n else -> millisToNanos(value)\n
}\n }\n\n // shortcuts\n\n /**\n * Returns the value of this duration expressed as a [Long] number of
nanoseconds.\n *\n * If the value doesn't fit in the range of [Long] type, it is coerced into that range, see the
conversion [Double.toLong] for details.\n *\n * The range of durations that can be expressed as a `Long`
number of nanoseconds is approximately \u00b1292 years.\n */\n @ExperimentalTime\n @Deprecated("Use
inWholeNanoseconds property instead.", ReplaceWith("this.inWholeNanoseconds"))\n public fun
toLongNanoseconds(): Long = inWholeNanoseconds\n\n /**\n * Returns the value of this duration expressed as
a [Long] number of milliseconds.\n *\n * The value is coerced to the range of [Long] type, if it doesn't fit in
that range, see the conversion [Double.toLong] for details.\n *\n * The range of durations that can be expressed
as a `Long` number of milliseconds is approximately \u00b1292 million years.\n */\n @ExperimentalTime\n
@Deprecated("Use inWholeMilliseconds property instead.", ReplaceWith("this.inWholeMilliseconds"))\n
public fun toLongMilliseconds(): Long = inWholeMilliseconds\n\n /**\n * Returns a string representation of
this duration value\n * expressed as a combination of numeric components, each in its own unit.\n *\n * Each
component is a number followed by the unit abbreviated name: `d`, `h`, `m`, `s`:\n * `5h`, `1d 12h`, `1h 0m
30.340s`.\n * The last component, usually seconds, can be a number with a fractional part.\n *\n * If the
duration is less than a second, it is represented as a single number\n * with one of sub-second units: `ms`
(milliseconds), `us` (microseconds), or `ns` (nanoseconds):\n * `140.884ms`, `500us`, `24ns`.\n *\n * A
negative duration is prefixed with `-` sign and, if it consists of multiple components, surrounded with parentheses:\n
* `-12m` and `-(1h 30m)`.\n *\n * Special cases:\n * - an infinite duration is formatted as `\"Infinity\"` or

```

```

`"-Infinity"` without a unit.\n * It's recommended to use [toISOString] that uses more strict ISO-8601
format instead of this `toString`\n * when you want to convert a duration to a string in cases of serialization,
interchange, etc.\n * @sample samples.time.Durations.toStringDefault\n * override fun toString():
String = when (rawValue) {\n 0L -> "0s"\n INFINITE.rawValue -> "Infinity"\n
NEG_INFINITE.rawValue -> "-Infinity"\n else -> {\n val isNegative = isNegative()\n
buildString {\n if (isNegative) append('-')\n absoluteValue.toComponents { days, hours, minutes,
seconds, nanoseconds ->\n val hasDays = days != 0\n val hasHours = hours != 0\n
val hasMinutes = minutes != 0\n val hasSeconds = seconds != 0 || nanoseconds != 0\n var
components = 0\n if (hasDays) {\n append(days).append('d')\n
components++\n }\n if (hasHours || (hasDays && (hasMinutes || hasSeconds))) {\n
if (components++ > 0) append(' ')\n append(hours).append('h')\n }\n if
(hasMinutes || (hasSeconds && (hasHours || hasDays))) {\n if (components++ > 0) append(' ')\n
append(minutes).append('m')\n }\n if (hasSeconds) {\n if
(components++ > 0) append(' ')\n when {\n seconds != 0 || hasDays || hasHours ||
hasMinutes ->\n appendFractional(seconds, nanoseconds, 9, "s", isoZeroes = false)\n
nanoseconds >= 1_000_000 ->\n appendFractional(nanoseconds / 1_000_000, nanoseconds
% 1_000_000, 6, "ms", isoZeroes = false)\n nanoseconds >= 1_000 ->\n
appendFractional(nanoseconds / 1_000, nanoseconds % 1_000, 3, "us", isoZeroes = false)\n else -
>\n append(nanoseconds).append("ns")\n }\n }\n if
(isNegative && components > 1) insert(1, ('').append(''))\n }\n }\n }\n private fun
StringBuilder.appendFractional(whole: Int, fractional: Int, fractionalSize: Int, unit: String, isoZeroes: Boolean) {\n
append(whole)\n if (fractional != 0) {\n append('.')\n val fracString =
fractional.toString().padStart(fractionalSize, '0')\n val nonZeroDigits = fracString.indexOfLast { it != '0' } +
1\n when {\n !isoZeroes && nonZeroDigits < 3 -> appendRange(fracString, 0, nonZeroDigits)\n
else -> appendRange(fracString, 0, ((nonZeroDigits + 2) / 3) * 3)\n }\n }\n append(unit)\n
}\n\n /**\n * Returns a string representation of this duration value expressed in the given [unit]\n * and
formatted with the specified [decimals] number of digits after decimal point.\n * Special cases:\n * - an
infinite duration is formatted as `"Infinity"` or `"-Infinity"` without a unit.\n * @param decimals the
number of digits after decimal point to show. The value must be non-negative.\n * No more than 12 decimals will
be shown, even if a larger number is requested.\n * @return the value of duration in the specified [unit]
followed by that unit abbreviated name: `d`, `h`, `m`, `s`, `ms`, `us`, or `ns`.\n * @throws
IllegalArgumentException if [decimals] is less than zero.\n * @sample
samples.time.Durations.toStringDecimals\n * public fun toString(unit: DurationUnit, decimals: Int = 0):
String {\n require(decimals >= 0) { "decimals must be not negative, but was $decimals" }\n val number =
toDouble(unit)\n if (number.isInfinite()) return number.toString()\n return formatToExactDecimals(number,
decimals.coerceAtMost(12)) + unit.shortName()\n }\n\n /**\n * Returns an ISO-8601 based string
representation of this duration.\n * The returned value is presented in the format `PThHmMs.fS`, where `h`,
`m`, `s` are the integer components of this duration (see [toComponents])\n * and `f` is a fractional part of second.
Depending on the roundness of the value the fractional part can be formatted with either\n * 0, 3, 6, or 9 decimal
digits.\n * The infinite duration is represented as `"PT999999999999H"` which is larger than any
possible finite duration in Kotlin.\n * Negative durations are indicated with the sign `-` in the beginning of
the returned string, for example, `"-PT5M30S"`.\n * @sample samples.time.Durations.toISOString\n
*\n * public fun toISOString(): String = buildString {\n if (isNegative()) append('-')\n append("PT")\n
this@Duration.absoluteValue.toComponents { hours, minutes, seconds, nanoseconds ->\n
@Suppress("NAME_SHADOWING")\n var hours = hours\n if (isInfinite()) {\n // use large
enough value instead of Long.MAX_VALUE\n hours = 9_999_999_999_999\n }\n val
hasHours = hours != 0\n val hasSeconds = seconds != 0 || nanoseconds != 0\n val hasMinutes =
minutes != 0 || (hasSeconds && hasHours)\n if (hasHours) {\n append(hours).append('H')\n

```



```

Double.microseconds get() = toDuration(DurationUnit.MICROSECONDS)\n\n/** Returns a [Duration] equal to
this [Int] number of milliseconds. *\n@SinceKotlin("1.3")\n@ExperimentalTime\n@Deprecated("Use
'Int.milliseconds' extension property from Duration.Companion instead.", ReplaceWith("this.milliseconds",
"kotlin.time.Duration.Companion.milliseconds"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic val
Int.milliseconds get() = toDuration(DurationUnit.MILLISECONDS)\n\n/** Returns a [Duration] equal to this
[Long] number of milliseconds. *\n@SinceKotlin("1.3")\n@ExperimentalTime\n@Deprecated("Use
'Long.milliseconds' extension property from Duration.Companion instead.", ReplaceWith("this.milliseconds",
"kotlin.time.Duration.Companion.milliseconds"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic val
Long.milliseconds get() = toDuration(DurationUnit.MILLISECONDS)\n\n/**\n * Returns a [Duration] equal to this
[Double] number of milliseconds.\n * *\n * @throws IllegalArgumentException if this [Double] value is `NaN`.\n
*\n@SinceKotlin("1.3")\n@ExperimentalTime\n@Deprecated("Use 'Double.milliseconds' extension property
from Duration.Companion instead.", ReplaceWith("this.milliseconds",
"kotlin.time.Duration.Companion.milliseconds"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic val
Double.milliseconds get() = toDuration(DurationUnit.MILLISECONDS)\n\n/** Returns a [Duration] equal to this
[Int] number of seconds. *\n@SinceKotlin("1.3")\n@ExperimentalTime\n@Deprecated("Use 'Int.seconds'
extension property from Duration.Companion instead.", ReplaceWith("this.seconds",
"kotlin.time.Duration.Companion.seconds"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic val
Int.seconds get() = toDuration(DurationUnit.SECONDS)\n\n/** Returns a [Duration] equal to this [Long] number of
seconds. *\n@SinceKotlin("1.3")\n@ExperimentalTime\n@Deprecated("Use 'Long.seconds' extension property
from Duration.Companion instead.", ReplaceWith("this.seconds",
"kotlin.time.Duration.Companion.seconds"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic val
Long.seconds get() = toDuration(DurationUnit.SECONDS)\n\n/**\n * Returns a [Duration] equal to this [Double]
number of seconds.\n * *\n * @throws IllegalArgumentException if this [Double] value is `NaN`.\n
*\n@SinceKotlin("1.3")\n@ExperimentalTime\n@Deprecated("Use 'Double.seconds' extension property from
Duration.Companion instead.", ReplaceWith("this.seconds",
"kotlin.time.Duration.Companion.seconds"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic val
Double.seconds get() = toDuration(DurationUnit.SECONDS)\n\n/** Returns a [Duration] equal to this [Int]
number of minutes. *\n@SinceKotlin("1.3")\n@ExperimentalTime\n@Deprecated("Use 'Int.minutes' extension
property from Duration.Companion instead.", ReplaceWith("this.minutes",
"kotlin.time.Duration.Companion.minutes"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic val
Int.minutes get() = toDuration(DurationUnit.MINUTES)\n\n/** Returns a [Duration] equal to this [Long] number of
minutes. *\n@SinceKotlin("1.3")\n@ExperimentalTime\n@Deprecated("Use 'Long.minutes' extension property
from Duration.Companion instead.", ReplaceWith("this.minutes",
"kotlin.time.Duration.Companion.minutes"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic val
Long.minutes get() = toDuration(DurationUnit.MINUTES)\n\n/**\n * Returns a [Duration] equal to this [Double]
number of minutes.\n * *\n * @throws IllegalArgumentException if this [Double] value is `NaN`.\n
*\n@SinceKotlin("1.3")\n@ExperimentalTime\n@Deprecated("Use 'Double.minutes' extension property from
Duration.Companion instead.", ReplaceWith("this.minutes",
"kotlin.time.Duration.Companion.minutes"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic val
Double.minutes get() = toDuration(DurationUnit.MINUTES)\n\n/** Returns a [Duration] equal to this [Int]
number of hours. *\n@SinceKotlin("1.3")\n@ExperimentalTime\n@Deprecated("Use 'Int.hours' extension
property from Duration.Companion instead.", ReplaceWith("this.hours",
"kotlin.time.Duration.Companion.hours"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic val
Int.hours get() = toDuration(DurationUnit.HOURS)\n\n/** Returns a [Duration] equal to this [Long] number of
hours. *\n@SinceKotlin("1.3")\n@ExperimentalTime\n@Deprecated("Use 'Long.hours' extension property from
Duration.Companion instead.", ReplaceWith("this.hours",
"kotlin.time.Duration.Companion.hours"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic val
Long.hours get() = toDuration(DurationUnit.HOURS)\n\n/**\n * Returns a [Duration] equal to this [Double]

```

```

number of hours.\n *\n * @throws IllegalArgumentException if this [Double] value is `NaN`.\n
*\n@SinceKotlin("1.3")\n@ExperimentalTime\n@Deprecated("Use 'Double.hours' extension property from
Duration.Companion instead.", ReplaceWith("this.hours"),
"kotlin.time.Duration.Companion.hours")\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic val
Double.hours get() = toDuration(DurationUnit.HOURS)\n\n/** Returns a [Duration] equal to this [Int] number of
days. *\n@SinceKotlin("1.3")\n@ExperimentalTime\n@Deprecated("Use 'Int.days' extension property from
Duration.Companion instead.", ReplaceWith("this.days"),
"kotlin.time.Duration.Companion.days")\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic val Int.days
get() = toDuration(DurationUnit.DAYS)\n\n/** Returns a [Duration] equal to this [Long] number of days.
*\n@SinceKotlin("1.3")\n@ExperimentalTime\n@Deprecated("Use 'Long.days' extension property from
Duration.Companion instead.", ReplaceWith("this.days"),
"kotlin.time.Duration.Companion.days")\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic val
Long.days get() = toDuration(DurationUnit.DAYS)\n\n/** Returns a [Duration] equal to this [Double] number
of days.\n *\n * @throws IllegalArgumentException if this [Double] value is `NaN`.\n
*\n@SinceKotlin("1.3")\n@ExperimentalTime\n@Deprecated("Use 'Double.days' extension property from
Duration.Companion instead.", ReplaceWith("this.days"),
"kotlin.time.Duration.Companion.days")\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic val
Double.days get() = toDuration(DurationUnit.DAYS)\n\n/** Returns a duration whose value is the specified
[duration] value multiplied by this number.
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalTime::class)\n@kotlin.internal.InlineOnly\npublic
inline operator fun Int.times(duration: Duration): Duration = duration * this\n\n/** Returns a duration whose
value is the specified [duration] value multiplied by this number.\n *\n * The operation may involve rounding when
the result cannot be represented exactly with a [Double] number.\n *\n * @throws IllegalArgumentException if the
operation results in a `NaN` value.\n
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalTime::class)\n@kotlin.internal.InlineOnly\npublic
inline operator fun Double.times(duration: Duration): Duration = duration * this\n\n\nprivate fun
parseDuration(value: String, strictIso: Boolean): Duration {\n    var length = value.length\n    if (length == 0) throw
IllegalArgumentException("The string is empty")\n    var index = 0\n    var result = Duration.ZERO\n    val
infinityString = "Infinity"\n    when (value[index]) {\n        '+', '-' -> index++\n    }\n    val hasSign = index > 0\n
val isNegative = hasSign && value.startsWith('-')\n    when {\n        length <= index ->\n            throw
IllegalArgumentException("No components")\n        value[index] == 'P' -> {\n            if (++index == length) throw
IllegalArgumentException()\n            val nonDigitSymbols = "+. "\n            var isTimeComponent = false\n
var prevUnit: DurationUnit? = null\n            while (index < length) {\n                if (value[index] == 'T') {\n
                    if (isTimeComponent || ++index == length) throw IllegalArgumentException()\n                    isTimeComponent =
true\n                    continue\n                }\n                val component = value.substringWhile(index) { it in '0'..'9' || it in
nonDigitSymbols }\n                if (component.isEmpty()) throw IllegalArgumentException()\n                index +=
component.length\n                val unitChar = value.getOrElse(index) { throw IllegalArgumentException("Missing
unit for value $component") }\n                index++\n                val unit = durationUnitByIsoChar(unitChar,
isTimeComponent)\n                if (prevUnit != null && prevUnit <= unit) throw
IllegalArgumentException("Unexpected order of duration components")\n                prevUnit = unit\n                val
dotIndex = component.indexOf('.')\n                if (unit == DurationUnit.SECONDS && dotIndex > 0) {\n
                    val whole = component.substring(0, dotIndex)\n                    result +=
parseOverLongIsoComponent(whole).toDuration(unit)\n                    result +=
component.substring(dotIndex).toDouble().toDuration(unit)\n                } else {\n                    result +=
parseOverLongIsoComponent(component).toDuration(unit)\n                }\n            }\n            strictIso ->\n                throw
IllegalArgumentException("value.regionMatches(index, infinityString, 0, length = maxOf(length -
index, infinityString.length), ignoreCase = true) -> {\n                    result = Duration.INFINITE\n                }\n            else -> {\n
                // parse default string format\n                var prevUnit: DurationUnit? = null\n                var afterFirst = false\n

```

```

var allowSpaces = !hasSign\n        if (hasSign && value[index] == '(' && value.last() == ')') {\n
allowSpaces = true\n        if (++index == --length) throw IllegalArgumentException("No components\").\n
    }\n        while (index < length) {\n            if (afterFirst && allowSpaces) {\n                index =
value.skipWhile(index) { it == ' ' }\n            }\n            afterFirst = true\n            val component =
value.substringWhile(index) { it in '0'..'9' || it == '.' }\n            if (component.isEmpty()) throw
IllegalArgumentException()\n            index += component.length\n            val unitName =
value.substringWhile(index) { it in 'a'..'z' }\n            index += unitName.length\n            val unit =
durationUnitByShortName(unitName)\n            if (prevUnit != null && prevUnit <= unit) throw
IllegalArgumentException("Unexpected order of duration components")\n            prevUnit = unit\n            val
dotIndex = component.indexOf('.')\n            if (dotIndex > 0) {\n                val whole = component.substring(0,
dotIndex)\n                result += whole.toLong().toDuration(unit)\n                result +=
component.substring(dotIndex).toDouble().toDuration(unit)\n                if (index < length) throw
IllegalArgumentException("Fractional component must be last")\n            } else {\n                result +=
component.toLong().toDuration(unit)\n            }\n        }\n    }\n    return if (isNegative) -result else
result\n}\n\nprivate fun parseOverLongIsoComponent(value: String): Long {\n    val length = value.length\n    var
startIndex = 0\n    if (length > 0 && value[0] in "+-") startIndex++\n    if ((length - startIndex) > 16 &&
(startIndex..value.lastIndex).all { value[it] in '0'..'9' }) {\n        // all chars are digits, but more than
ceiling(log10(MAX_MILLIS / 1000)) of them\n        return if (value[0] == '-') Long.MIN_VALUE else
Long.MAX_VALUE\n    }\n    // TODO: replace with just toLong after min JDK becomes 8\n    return if
(value.startsWith("+")) value.drop(1).toLong() else value.toLong()\n}\n\nprivate inline fun
String.substringWhile(startIndex: Int, predicate: (Char) -> Boolean): String =\n    substring(startIndex,
skipWhile(startIndex, predicate))\n\nprivate inline fun String.skipWhile(startIndex: Int, predicate: (Char) ->
Boolean): Int {\n    var i = startIndex\n    while (i < length && predicate(this[i])) i++\n    return i\n}\n\nThe ranges are chosen so that they are:\n// - symmetric relative to zero: this greatly simplifies operations with sign,
e.g. unaryMinus and minus.\n// - non-overlapping, but adjacent: the first value that doesn't fit in nanos range, can be
exactly represented in millis.\n\ninternal const val NANOS_IN_MILLIS = 1_000_000\n// maximum number
duration can store in nanosecond range\ninternal const val MAX_NANOS = Long.MAX_VALUE / 2 /
NANOS_IN_MILLIS * NANOS_IN_MILLIS - 1 // ends in ...999_999\n// maximum number duration can store in
millisecond range, also encodes an infinite value\ninternal const val MAX_MILLIS = Long.MAX_VALUE / 2\n//
MAX_NANOS expressed in milliseconds\nprivate const val MAX_NANOS_IN_MILLIS = MAX_NANOS /
NANOS_IN_MILLIS\n\nprivate fun nanosToMillis(nanos: Long): Long = nanos / NANOS_IN_MILLIS\nprivate
fun millisToNanos(millis: Long): Long = millis * NANOS_IN_MILLIS\n\nprivate fun
durationOfNanos(normalNanos: Long) = Duration(normalNanos shl 1)\nprivate fun durationOfMillis(normalMillis:
Long) = Duration((normalMillis shl 1) + 1)\nprivate fun durationOf(normalValue: Long, unitDiscriminator: Int) =
Duration((normalValue shl 1) + unitDiscriminator)\nprivate fun durationOfNanosNormalized(nanos: Long) =\n    if
(nanos in -MAX_NANOS..MAX_NANOS) {\n        durationOfNanos(nanos)\n    } else {\n
durationOfMillis(nanosToMillis(nanos))\n    }\n\nprivate fun durationOfMillisNormalized(millis: Long) =\n    if
(millis in -MAX_NANOS_IN_MILLIS..MAX_NANOS_IN_MILLIS) {\n
durationOfNanos(millisToNanos(millis))\n    } else {\n        durationOfMillis(millis.coerceIn(-MAX_MILLIS,
MAX_MILLIS))\n    }\n\ninternal expect val durationAssertionsEnabled: Boolean\n\ninternal expect fun
formatToExactDecimals(value: Double, decimals: Int): String\n\ninternal expect fun formatUpToDecimals(value:
Double, decimals: Int): String", "/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\n@file:kotlin.jvm.JvmName("UnsignedKt")\n\npackage
kotlin\n\n@PublishedApi\n\ninternal fun uintCompare(v1: Int, v2: Int): Int = (v1 xor
Int.MIN_VALUE).compareTo(v2 xor Int.MIN_VALUE)\n\n@PublishedApi\n\ninternal fun ulongCompare(v1: Long,
v2: Long): Int = (v1 xor Long.MIN_VALUE).compareTo(v2 xor Long.MIN_VALUE)\n\n@PublishedApi\n\ninternal
fun uintDivide(v1: UInt, v2: UInt): UInt = (v1.toLong() / v2.toLong()).toUInt()\n\n@PublishedApi\n\ninternal fun

```

```

uintRemainder(v1: UInt, v2: UInt): UInt = (v1.toLong() % v2.toLong()).toUInt()\n\n// Division and remainder are
based on Guava's UnsignedLongs implementation\n\n// Copyright 2011 The Guava
Authors\n\n@PublishedApi\ninternal fun ulongDivide(v1: ULong, v2: ULong): ULong {\n    val dividend =
v1.toLong()\n    val divisor = v2.toLong()\n    if (divisor < 0) { // i.e., divisor >= 2^63:\n        return if (v1 < v2)
ULong(0) else ULong(1)\n    }\n    // Optimization - use signed division if both dividend and divisor < 2^63\n    if
(dividend >= 0) {\n        return ULong(dividend / divisor)\n    }\n    // Otherwise, approximate the quotient, check,
and correct if necessary.\n    val quotient = ((dividend ushr 1) / divisor) shl 1\n    val rem = dividend - quotient *
divisor\n    return ULong(quotient + if (ULong(rem) >= ULong(divisor)) 1 else 0)\n}\n\n@PublishedApi\ninternal
fun ulongRemainder(v1: ULong, v2: ULong): ULong {\n    val dividend = v1.toLong()\n    val divisor =
v2.toLong()\n    if (divisor < 0) { // i.e., divisor >= 2^63:\n        return if (v1 < v2) {\n            v1 // dividend <
divisor\n        } else {\n            v1 - v2 // dividend >= divisor\n        }\n    }\n    // Optimization - use signed
modulus if both dividend and divisor < 2^63\n    if (dividend >= 0) {\n        return ULong(dividend % divisor)\n
    }\n    // Otherwise, approximate the quotient, check, and correct if necessary.\n    val quotient = ((dividend ushr 1)
/ divisor) shl 1\n    val rem = dividend - quotient * divisor\n    return ULong(rem - if (ULong(rem) >=
ULong(divisor)) divisor else 0)\n}\n\n@PublishedApi\ninternal fun doubleToUInt(v: Double): UInt = when {\n
v.isNaN() -> 0u\n    v <= UInt.MIN_VALUE.toDouble() -> UInt.MIN_VALUE\n    v >=
UInt.MAX_VALUE.toDouble() -> UInt.MAX_VALUE\n    v <= Int.MAX_VALUE -> v.toInt().toUInt()\n    else -
> (v - Int.MAX_VALUE).toInt().toUInt() + Int.MAX_VALUE.toUInt() // Int.MAX_VALUE < v <
UInt.MAX_VALUE\n}\n\n@PublishedApi\ninternal fun doubleToULong(v: Double): ULong = when {\n
v.isNaN() -> 0u\n    v <= ULong.MIN_VALUE.toDouble() -> ULong.MIN_VALUE\n    v >=
ULong.MAX_VALUE.toDouble() -> ULong.MAX_VALUE\n    v < Long.MAX_VALUE ->
v.toLong().toULong()\n    // Real values from Long.MAX_VALUE to (Long.MAX_VALUE + 1) are not
representable in Double, so don't handle them.\n    else -> (v - 9223372036854775808.0).toLong().toULong() +
9223372036854775808uL // Long.MAX_VALUE + 1 < v <
ULong.MAX_VALUE\n}\n\n@PublishedApi\ninternal fun uintToDouble(v: Int): Double = (v and
Int.MAX_VALUE).toDouble() + (v ushr 31 shl 30).toDouble() * 2\n\n@PublishedApi\ninternal fun
ulongToDouble(v: Long): Double = (v ushr 11).toDouble() * 2048 + (v and 2047)\n\n@PublishedApi\ninternal fun
ulongToString(v: Long): String = ulongToString(v, 10)\n\n@PublishedApi\ninternal fun ulongToString(v: Long, base: Int): String {\n
if (v >= 0) return v.toString(base)\n    var quotient = ((v ushr 1) / base) shl 1\n    var rem = v - quotient * base\n
if (rem >= base) {\n        rem -= base\n        quotient += 1\n    }\n    return quotient.toString(base) +
rem.toString(base)\n}\n\n", /*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n
*\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("CollectionsKt")\n\npackage
kotlin.collections\n\n/**\n * Given an [iterator] function constructs an [Iterable] instance that returns values through
the [Iterator]\n * provided by that function.\n * @sample samples.collections.Iterables.Building.iterable\n
*\n\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable(crossinline iterator: () -> Iterator<T>): Iterable<T>
= object : Iterable<T> {\n    override fun iterator(): Iterator<T> = iterator()\n}\n\n/**\n * A wrapper over another
[Iterable] (or any other object that can produce an [Iterator]) that returns\n * an indexing iterator.\n *\n\n@kotlin.internal class
IndexingIterable<out T>(private val iteratorFactory: () -> Iterator<T>) : Iterable<IndexedValue<T>> {\n    override
fun iterator(): Iterator<IndexedValue<T>> = IndexingIterator(iteratorFactory())\n}\n\n/**\n * Returns the size of
this iterable if it is known, or `null` otherwise.\n *\n\n@PublishedApi\ninternal fun <T>
Iterable<T>.collectionSizeOrNull(): Int? = if (this is Collection<*>) this.size else null\n\n/**\n * Returns the size of
this iterable if it is known, or the specified [default] value otherwise.\n *\n\n@PublishedApi\ninternal fun <T>
Iterable<T>.collectionSizeOrDefault(default: Int): Int = if (this is Collection<*>) this.size else default\n\n/**\n *
Returns a single list of all elements from all collections in the given collection.\n * @sample
samples.collections.Iterables.Operations.flattenIterable\n *\n\n@kotlin.internal public fun <T> Iterable<Iterable<T>>.flatten():
List<T> {\n    val result = ArrayList<T>()\n    for (element in this) {\n        result.addAll(element)\n    }\n    return

```

```

result\n}\n\n/**\n * Returns a pair of lists, where\n * *first* list is built from the first values of each pair from this
collection,\n * *second* list is built from the second values of each pair from this collection.\n * @sample
samples.collections.Iterables.Operations.unzipIterable\n * \npublic fun <T, R> Iterable<Pair<T, R>>.unzip():
Pair<List<T>, List<R>> {\n    val expectedSize = collectionSizeOrDefault(10)\n    val listT =
ArrayList<T>(expectedSize)\n    val listR = ArrayList<R>(expectedSize)\n    for (pair in this) {\n
listT.add(pair.first)\n    listR.add(pair.second)\n    }\n    return listT to listR\n}\n\n"/*\n * Copyright 2010-2020
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("\SequencesKt")\n\npackage
kotlin.sequences\n\nimport kotlin.random.Random\n\n/**\n * Given an [iterator] function constructs a [Sequence]
that returns values through the [Iterator]\n * provided by that function.\n * The values are evaluated lazily, and the
sequence is potentially infinite.\n * \n * @sample samples.collections.Sequences.Building.sequenceFromIterator\n
*\n\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence(crossinline iterator: () -> Iterator<T>):
Sequence<T> = object : Sequence<T> {\n    override fun iterator(): Iterator<T> = iterator()\n}\n\n"/*\n * Creates a
sequence that returns all elements from this iterator. The sequence is constrained to be iterated only once.\n * \n *
@sample samples.collections.Sequences.Building.sequenceFromIterator\n * \n\npublic fun <T>
Iterator<T>.asSequence(): Sequence<T> = Sequence { this }.constrainOnce()\n\n"/*\n * Creates a sequence that
returns the specified values.\n * \n * @sample samples.collections.Sequences.Building.sequenceOfValues\n
*\n\npublic fun <T> sequenceOf(vararg elements: T): Sequence<T> = if (elements.isEmpty()) emptySequence() else
elements.asSequence()\n\n"/*\n * Returns an empty sequence.\n * \n\npublic fun <T> emptySequence():
Sequence<T> = EmptySequence\n\nprivate object EmptySequence : Sequence<Nothing>,
DropTakeSequence<Nothing> {\n    override fun iterator(): Iterator<Nothing> = EmptyIterator\n    override fun
drop(n: Int) = EmptySequence\n    override fun take(n: Int) = EmptySequence\n}\n\n"/*\n * Returns this sequence if
it's not `null` and the empty sequence otherwise.\n * @sample
samples.collections.Sequences.Usage.sequenceOrEmpty\n
*\n\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>?.orEmpty():
Sequence<T> = this ?: emptySequence()\n\n"/*\n * Returns a sequence that iterates through the elements either of
this sequence\n * or, if this sequence turns out to be empty, of the sequence returned by [defaultValue] function.\n
*\n * @sample samples.collections.Sequences.Usage.sequenceIfEmpty\n *\n\n@SinceKotlin("1.3")\npublic fun
<T> Sequence<T>.ifEmpty(defaultValue: () -> Sequence<T>): Sequence<T> = sequence {\n    val iterator =
this@ifEmpty.iterator()\n    if (iterator.hasNext()) {\n        yieldAll(iterator)\n    } else {\n
yieldAll(defaultValue())\n    }\n}\n\n"/*\n * Returns a sequence of all elements from all sequences in this
sequence.\n * \n * The operation is _intermediate_ and _stateless_.\n * \n * @sample
samples.collections.Sequences.Transformations.flattenSequenceOfSequences\n * \n\npublic fun <T>
Sequence<Sequence<T>>.flatten(): Sequence<T> = flatten { it.iterator() }\n\n"/*\n * Returns a sequence of all
elements from all iterables in this sequence.\n * \n * The operation is _intermediate_ and _stateless_.\n * \n *
@sample samples.collections.Sequences.Transformations.flattenSequenceOfLists\n
*\n\n@kotlin.jvm.JvmName("\flattenSequenceOfIterable")\n\npublic fun <T> Sequence<Iterable<T>>.flatten():
Sequence<T> = flatten { it.iterator() }\n\nprivate fun <T, R> Sequence<T>.flatten(iterator: (T) -> Iterator<R>):
Sequence<R> {\n    if (this is TransformingSequence<*, *>) {\n        return (this as TransformingSequence<*,
T>).flatten(iterator)\n    }\n    return FlatteningSequence(this, { it }, iterator)\n}\n\n"/*\n * Returns a pair of lists,
where\n * *first* list is built from the first values of each pair from this sequence,\n * *second* list is built from the
second values of each pair from this sequence.\n * \n * The operation is _terminal_.\n * \n * @sample
samples.collections.Sequences.Transformations.unzip\n * \n\npublic fun <T, R> Sequence<Pair<T, R>>.unzip():
Pair<List<T>, List<R>> {\n    val listT = ArrayList<T>()\n    val listR = ArrayList<R>()\n    for (pair in this) {\n
listT.add(pair.first)\n    listR.add(pair.second)\n    }\n    return listT to listR\n}\n\n"/*\n * Returns a sequence that
yields elements of this sequence randomly shuffled.\n * \n * Note that every iteration of the sequence returns
elements in a different order.\n * \n * The operation is _intermediate_ and _stateful_.\n

```

```

*\/n@SinceKotlin("1.4")\npublic fun <T> Sequence<T>.shuffled(): Sequence<T> = shuffled(Random)\n\n/**\n * Returns a sequence that yields elements of this sequence randomly shuffled\n * using the specified [random] instance as the source of randomness.\n * Note that every iteration of the sequence returns elements in a different order.\n * The operation is _intermediate_ and _stateful_.\n */\n@SinceKotlin("1.4")\npublic fun <T> Sequence<T>.shuffled(random: Random): Sequence<T> = sequence<T> {\n    val buffer = toMutableList()\n    while (buffer.isNotEmpty()) {\n        val j = random.nextInt(buffer.size)\n        val last = buffer.removeLast()\n        val value = if (j < buffer.size) buffer.set(j, last) else last\n        yield(value)\n    }\n}\n\n/**\n * A sequence that returns the values from the underlying [sequence] that either match or do not match\n * the specified [predicate].\n * @param sendWhen If `true`, values for which the predicate returns `true` are returned. Otherwise,\n * values for which the predicate returns `false` are returned\n */\ninternal class FilteringSequence<T>(\n    private val sequence: Sequence<T>,\n    private val sendWhen: Boolean = true,\n    private val predicate: (T) -> Boolean\n) : Sequence<T> {\n    override fun iterator(): Iterator<T> = object : Iterator<T> {\n        val iterator = sequence.iterator()\n        var nextState: Int = -1 // -1 for unknown, 0 for done, 1 for continue\n        var nextItem: T? = null\n        private fun calcNext() {\n            while (iterator.hasNext()) {\n                val item = iterator.next()\n                if (predicate(item) == sendWhen) {\n                    nextItem = item\n                    nextState = 1\n                }\n            }\n            nextState = 0\n        }\n        override fun next(): T {\n            if (nextState == -1)\n                calcNext()\n            if (nextState == 0)\n                throw NoSuchElementException()\n            val result = nextItem\n            nextItem = null\n            nextState = -1\n            @Suppress("UNCHECKED_CAST")\n            return result as T\n        }\n        override fun hasNext(): Boolean {\n            if (nextState == -1)\n                calcNext()\n            return nextState == 1\n        }\n    }\n}\n\n/**\n * A sequence which returns the results of applying the given [transformer] function to the values\n * in the underlying [sequence].\n */\ninternal class TransformingSequence<T, R>\nconstructor(private val sequence: Sequence<T>, private val transformer: (T) -> R) : Sequence<R> {\n    override fun iterator(): Iterator<R> = object : Iterator<R> {\n        val iterator = sequence.iterator()\n        override fun next(): R {\n            return transformer(iterator.next())\n        }\n        override fun hasNext(): Boolean {\n            return iterator.hasNext()\n        }\n    }\n\n    internal fun <E> flatten(iterator: (R) -> Iterator<E>): Sequence<E> {\n        return FlatteningSequence<T, R, E>(sequence, transformer, iterator)\n    }\n}\n\n/**\n * A sequence which returns the results of applying the given [transformer] function to the values\n * in the underlying [sequence], where the transformer function takes the index of the value in the underlying\n * sequence along with the value itself.\n */\ninternal class TransformingIndexedSequence<T, R>\nconstructor(private val sequence: Sequence<T>, private val transformer: (Int, T) -> R) : Sequence<R> {\n    override fun iterator(): Iterator<R> = object : Iterator<R> {\n        val iterator = sequence.iterator()\n        var index = 0\n        override fun next(): R {\n            return transformer(checkIndexOverflow(index++), iterator.next())\n        }\n        override fun hasNext(): Boolean {\n            return iterator.hasNext()\n        }\n    }\n}\n\n/**\n * A sequence which combines values from the underlying [sequence] with their indices and returns them as\n * [IndexedValue] objects.\n */\ninternal class IndexingSequence<T>\nconstructor(private val sequence: Sequence<T>) : Sequence<IndexedValue<T>> {\n    override fun iterator(): Iterator<IndexedValue<T>> = object : Iterator<IndexedValue<T>> {\n        val iterator = sequence.iterator()\n        var index = 0\n        override fun next(): IndexedValue<T> {\n            return IndexedValue(checkIndexOverflow(index++), iterator.next())\n        }\n        override fun hasNext(): Boolean {\n            return iterator.hasNext()\n        }\n    }\n}\n\n/**\n * A sequence which takes the values from two parallel underlying sequences, passes them to the given\n * [transform] function and returns the values returned by that function. The sequence stops returning\n * values as soon as one of the underlying sequences stops returning values.\n */\ninternal class MergingSequence<T1, T2, V>\nconstructor(\n    private val sequence1: Sequence<T1>,\n    private val sequence2: Sequence<T2>,\n    private val transform: (T1, T2) -> V\n) : Sequence<V> {\n    override fun iterator(): Iterator<V> = object : Iterator<V> {\n        val iterator1 = sequence1.iterator()\n        val iterator2 = sequence2.iterator()\n        override fun next(): V {\n            return transform(iterator1.next(), iterator2.next())\n        }\n        override fun hasNext(): Boolean {\n            return iterator1.hasNext() && iterator2.hasNext()\n        }\n    }\n}\n\ninternal class FlatteningSequence<T, R, E>\nconstructor(\n    private val sequence: Sequence<T>,\n    private val transformer: (T) -> R,\n    private val

```

```

iterator: (R) -> Iterator<E>\n) : Sequence<E> {\n  override fun iterator(): Iterator<E> = object : Iterator<E> {\n
val iterator = sequence.iterator()\n    var itemIterator: Iterator<E>? = null\n\n    override fun next(): E {\n
if (!ensureItemIterator())\n        throw NoSuchElementException()\n        return itemIterator!!.next()\n
}\n\n    override fun hasNext(): Boolean {\n        return ensureItemIterator()\n    }\n\n    private fun
ensureItemIterator(): Boolean {\n        if (itemIterator?.hasNext() == false)\n            itemIterator = null\n\n        while (itemIterator == null) {\n            if (!iterator.hasNext()) {\n                return false\n            } else {\n
                val element = iterator.next()\n                val nextItemIterator = iterator(transformer(element))\n
            }\n            itemIterator = nextItemIterator\n            return true\n        }\n    }\n\n    return true\n    }\n}\n\ninternal fun <T, C, R> flatMapIndexed(source:
Sequence<T>, transform: (Int, T) -> C, iterator: (C) -> Iterator<R>): Sequence<R> =\n    sequence {\n        var
index = 0\n        for (element in source) {\n            val result = transform(checkIndexOverflow(index++), element)\n
            yieldAll(iterator(result))\n        }\n    }\n\n/**\n * A sequence that supports drop(n) and take(n) operations\n
*\n\ninternal interface DropTakeSequence<T> : Sequence<T> {\n    fun drop(n: Int): Sequence<T>\n    fun take(n:
Int): Sequence<T>\n}\n\n/**\n * A sequence that skips [startIndex] values from the underlying [sequence]\n * and
stops returning values right before [endIndex], i.e. stops at `endIndex - 1`\n *\n\ninternal class SubSequence<T>(\n
private val sequence: Sequence<T>,\n    private val startIndex: Int,\n    private val endIndex: Int\n) : Sequence<T>,\n
DropTakeSequence<T> {\n\n    init {\n        require(startIndex >= 0) { \"startIndex should be non-negative, but is
$startIndex\" }\n        require(endIndex >= 0) { \"endIndex should be non-negative, but is $endIndex\" }\n
        require(endIndex >= startIndex) { \"endIndex should be not less than startIndex, but was $endIndex < $startIndex\"
}\n    }\n\n    private val count: Int get() = endIndex - startIndex\n\n    override fun drop(n: Int): Sequence<T> = if (n
>= count) emptySequence() else SubSequence(sequence, startIndex + n, endIndex)\n    override fun take(n: Int):
Sequence<T> = if (n >= count) this else SubSequence(sequence, startIndex, startIndex + n)\n\n    override fun
iterator() = object : Iterator<T> {\n        val iterator = sequence.iterator()\n        var position = 0\n        //
Shouldn't be called from constructor to avoid premature iteration\n        private fun drop() {\n            while (position
< startIndex && iterator.hasNext()) {\n                iterator.next()\n                position++\n            }\n        }\n
        override fun hasNext(): Boolean {\n            drop()\n            return (position < endIndex) && iterator.hasNext()\n
        }\n\n        override fun next(): T {\n            drop()\n            if (position >= endIndex)\n                throw
NoSuchElementException()\n            position++\n            return iterator.next()\n        }\n    }\n}\n\n/**\n * A
sequence that returns at most [count] values from the underlying [sequence], and stops returning values\n * as soon
as that count is reached.\n *\n\ninternal class TakeSequence<T>(\n    private val sequence: Sequence<T>,\n    private
val count: Int\n) : Sequence<T>, DropTakeSequence<T> {\n\n    init {\n        require(count >= 0) { \"count must be
non-negative, but was $count.\" }\n    }\n\n    override fun drop(n: Int): Sequence<T> = if (n >= count)\n
emptySequence() else SubSequence(sequence, n, count)\n    override fun take(n: Int): Sequence<T> = if (n >=
count) this else TakeSequence(sequence, n)\n\n    override fun iterator(): Iterator<T> = object : Iterator<T> {\n
var left = count\n        val iterator = sequence.iterator()\n\n        override fun next(): T {\n            if (left == 0)\n
                throw NoSuchElementException()\n            left--\n            return iterator.next()\n        }\n\n        override fun
hasNext(): Boolean {\n            return left > 0 && iterator.hasNext()\n        }\n    }\n}\n\n/**\n * A sequence that
returns values from the underlying [sequence] while the [predicate] function returns\n * `true`, and stops returning
values once the function returns `false` for the next element.\n *\n\ninternal class
TakeWhileSequence<T>(\n    constructor(\n        private val sequence: Sequence<T>,\n        private val predicate: (T) ->
Boolean\n) : Sequence<T> {\n    override fun iterator(): Iterator<T> = object : Iterator<T> {\n        val iterator =
sequence.iterator()\n        var nextState: Int = -1 // -1 for unknown, 0 for done, 1 for continue\n        var nextItem: T?
= null\n\n        private fun calcNext() {\n            if (iterator.hasNext()) {\n                val item = iterator.next()\n
                if (predicate(item)) {\n                    nextState = 1\n                    nextItem = item\n                    return\n                }\n
                nextState = 0\n            }\n        }\n\n        override fun next(): T {\n            if (nextState == -1)\n
                calcNext() // will change nextState\n            if (nextState == 0)\n                throw NoSuchElementException()\n
                @Suppress(\"UNCHECKED_CAST\")\n                val result = nextItem as T\n                // Clean next to avoid keeping
reference on yielded instance\n                nextItem = null\n                nextState = -1\n                return result\n            }\n
}\n

```



```

fun <T : Any> generateSequence(nextFunction: () -> T?): Sequence<T> {
    return
    GeneratorSequence(nextFunction, { nextFunction() }).constrainOnce()
}
/**
 * Returns a sequence defined by the starting value [seed] and the function [nextFunction],
 * which is invoked to calculate the next value based on the previous one on each iteration.
 * The sequence produces values until it encounters first `null` value.
 * If [seed] is `null`, an empty sequence is produced.
 * The sequence can be iterated multiple times, each time starting with [seed].
 * @see kotlin.sequences.sequence
 * @sample
    samples.collections.Sequences.Building.generateSequenceWithSeed
*/
@kotlin.internal.LowPriorityInOverloadResolution
public fun <T : Any> generateSequence(seed: T?, nextFunction: (T) -> T?): Sequence<T> =
    if (seed == null) EmptySequence
    else
    GeneratorSequence({ seed }, nextFunction)
/**
 * Returns a sequence defined by the function [seedFunction], which is invoked to produce the starting value,
 * and the [nextFunction], which is invoked to calculate the next value based on the previous one on each iteration.
 * The sequence produces values until it encounters first `null` value.
 * If [seedFunction] returns `null`, an empty sequence is produced.
 * The sequence can be iterated multiple times.
 * @see kotlin.sequences.sequence
 * @sample
    samples.collections.Sequences.Building.generateSequenceWithLazySeed
*/
public fun <T : Any> generateSequence(seedFunction: () -> T?, nextFunction: (T) -> T?): Sequence<T> =
    GeneratorSequence(seedFunction, nextFunction)
}
/**
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
*/
@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("PreconditionsKt")
package kotlin
import kotlin.contracts.contract
/**
 * Throws an [IllegalArgumentException] if the [value] is false.
 * @sample samples.misc.Preconditions.failRequireWithLazyMessage
*/
@kotlin.internal.InlineOnly
public inline fun require(value: Boolean): Unit {
    contract {
        returns() implies value
    }
    require(value) {
        "Failed requirement."
    }
}
/**
 * Throws an [IllegalArgumentException] with the result of calling [lazyMessage] if the [value] is false.
 * @sample samples.misc.Preconditions.failRequireWithLazyMessage
*/
@kotlin.internal.InlineOnly
public inline fun require(value: Boolean, lazyMessage: () -> Any): Unit {
    contract {
        returns() implies value
    }
    if (!value) {
        val message = lazyMessage()
        throw IllegalArgumentException(message.toString())
    }
}
/**
 * Throws an [IllegalArgumentException] if the [value] is null. Otherwise returns the not null value.
*/
@kotlin.internal.InlineOnly
public inline fun <T : Any> requireNotNull(value: T?): T {
    contract {
        returns() implies (value != null)
    }
    return requireNotNull(value) { "Required value was null." }
}
/**
 * Throws an [IllegalArgumentException] with the result of calling [lazyMessage] if the [value] is null. Otherwise
 * returns the not null value.
 * @sample samples.misc.Preconditions.failRequireNotNullWithLazyMessage
*/
@kotlin.internal.InlineOnly
public inline fun <T : Any> requireNotNull(value: T?, lazyMessage: () -> Any): T {
    contract {
        returns() implies (value != null)
    }
    if (value == null) {
        val message = lazyMessage()
        throw IllegalArgumentException(message.toString())
    } else {
        return value
    }
}
/**
 * Throws an [IllegalStateException] if the [value] is false.
 * @sample
    samples.misc.Preconditions.failCheckWithLazyMessage
*/
@kotlin.internal.InlineOnly
public inline fun check(value: Boolean): Unit {
    contract {
        returns() implies value
    }
    check(value) { "Check failed." }
}
/**
 * Throws an [IllegalStateException] with the result of calling [lazyMessage] if the [value] is false.
 * @sample samples.misc.Preconditions.failCheckWithLazyMessage
*/
@kotlin.internal.InlineOnly
public inline fun check(value: Boolean, lazyMessage: () -> Any): Unit {
    contract {
        returns() implies value
    }
    if (!value) {
        val message = lazyMessage()
        throw IllegalStateException(message.toString())
    }
}
/**
 * Throws an [IllegalStateException] if the [value] is null. Otherwise
 * returns the not null value.
 * @sample samples.misc.Preconditions.failCheckWithLazyMessage
*/
@kotlin.internal.InlineOnly
public inline fun <T : Any> checkNotNull(value: T?): T {
    contract {
        returns() implies (value != null)
    }
    return checkNotNull(value) { "Required value was null." }
}
/**
 * Throws an [IllegalStateException] with the result of calling [lazyMessage] if the [value] is null. Otherwise
 * returns the not null value.
 * @sample

```

```

samples.misc.Preconditions.failCheckWithLazyMessage\n *^\n@kotlin.internal.InlineOnly\npublic inline fun <T :
Any> checkNotNull(value: T?, lazyMessage: () -> Any): T {\n  contract {\n    returns() implies (value != null)\n  }\n  if (value == null) {\n    val message = lazyMessage()\n    throw
IllegalStateException(message.toString())\n  } else {\n    return value\n  }\n}\n\n/**\n * Throws an
[IllegalStateException] with the given [message].\n *\n * @sample samples.misc.Preconditions.failWithError\n
*^\n@kotlin.internal.InlineOnly\npublic inline fun error(message: Any): Nothing = throw
IllegalStateException(message.toString())\n", /*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n *\n\npackage kotlin.collections\n\n//\n\n// NOTE: THIS FILE IS AUTO-GENERATED
by the GenerateStandardLib.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\n\nimport
kotlin.js.*\nimport primitiveArrayConcat\nimport withType\nimport kotlin.ranges.contains\nimport
kotlin.ranges.reversed\n\n/**\n * Returns an element at the given [index] or throws an
[IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n *\n *\n * @sample
samples.collections.Collections.Elements.elementAt\n *\n\npublic actual fun <T> Array<out T>.elementAt(index:
Int): T {\n  return elementAtOrElse(index) { throw IndexOutOfBoundsException("\nindex: $index, size: $size\n")
}\n}\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is
out of bounds of this array.\n *\n *\n * @sample samples.collections.Collections.Elements.elementAt\n *\n\npublic
actual fun ByteArray.elementAt(index: Int): Byte {\n  return elementAtOrElse(index) { throw
IndexOutOfBoundsException("\nindex: $index, size: $size\n") }\n}\n\n/**\n * Returns an element at the given
[index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n *\n *\n * @sample
samples.collections.Collections.Elements.elementAt\n *\n\npublic actual fun ShortArray.elementAt(index: Int): Short
{\n  return elementAtOrElse(index) { throw IndexOutOfBoundsException("\nindex: $index, size: $size\n")
}\n}\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is
out of bounds of this array.\n *\n *\n * @sample samples.collections.Collections.Elements.elementAt\n *\n\npublic
actual fun IntArray.elementAt(index: Int): Int {\n  return elementAtOrElse(index) { throw
IndexOutOfBoundsException("\nindex: $index, size: $size\n") }\n}\n\n/**\n * Returns an element at the given
[index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n *\n *\n * @sample
samples.collections.Collections.Elements.elementAt\n *\n\npublic actual fun LongArray.elementAt(index: Int): Long
{\n  return elementAtOrElse(index) { throw IndexOutOfBoundsException("\nindex: $index, size: $size\n")
}\n}\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is
out of bounds of this array.\n *\n *\n * @sample samples.collections.Collections.Elements.elementAt\n *\n\npublic
actual fun FloatArray.elementAt(index: Int): Float {\n  return elementAtOrElse(index) { throw
IndexOutOfBoundsException("\nindex: $index, size: $size\n") }\n}\n\n/**\n * Returns an element at the given
[index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n *\n *\n * @sample
samples.collections.Collections.Elements.elementAt\n *\n\npublic actual fun DoubleArray.elementAt(index: Int):
Double {\n  return elementAtOrElse(index) { throw IndexOutOfBoundsException("\nindex: $index, size: $size\n")
}\n}\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is
out of bounds of this array.\n *\n *\n * @sample samples.collections.Collections.Elements.elementAt\n *\n\npublic
actual fun BooleanArray.elementAt(index: Int): Boolean {\n  return elementAtOrElse(index) { throw
IndexOutOfBoundsException("\nindex: $index, size: $size\n") }\n}\n\n/**\n * Returns an element at the given
[index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n *\n *\n * @sample
samples.collections.Collections.Elements.elementAt\n *\n\npublic actual fun CharArray.elementAt(index: Int): Char
{\n  return elementAtOrElse(index) { throw IndexOutOfBoundsException("\nindex: $index, size: $size\n")
}\n}\n\n/**\n * Returns a [List] that wraps the original array.\n *\n\npublic actual fun <T> Array<out T>.asList():
List<T> {\n  return ArrayList<T>(this.unsafeCast<Array<Any?>>())\n}\n\n/**\n * Returns a [List] that wraps the
original array.\n *\n\n@kotlin.internal.InlineOnly\npublic actual inline fun ByteArray.asList(): List<Byte> {\n
return this.unsafeCast<Array<Byte>>().asList()\n}\n\n/**\n * Returns a [List] that wraps the original array.\n
*^\n@kotlin.internal.InlineOnly\npublic actual inline fun ShortArray.asList(): List<Short> {\n  return

```

```

this.unsafeCast<Array<Short>>().asList()\n\n/**\n * Returns a [List] that wraps the original array.\n */\n@kotlin.internal.InlineOnly\npublic actual inline fun IntArray.asList(): List<Int> {\n    return\n    this.unsafeCast<Array<Int>>().asList()\n}\n\n/**\n * Returns a [List] that wraps the original array.\n */\n@kotlin.internal.InlineOnly\npublic actual inline fun LongArray.asList(): List<Long> {\n    return\n    this.unsafeCast<Array<Long>>().asList()\n}\n\n/**\n * Returns a [List] that wraps the original array.\n */\n@kotlin.internal.InlineOnly\npublic actual inline fun FloatArray.asList(): List<Float> {\n    return\n    this.unsafeCast<Array<Float>>().asList()\n}\n\n/**\n * Returns a [List] that wraps the original array.\n */\n@kotlin.internal.InlineOnly\npublic actual inline fun DoubleArray.asList(): List<Double> {\n    return\n    this.unsafeCast<Array<Double>>().asList()\n}\n\n/**\n * Returns a [List] that wraps the original array.\n */\n@kotlin.internal.InlineOnly\npublic actual inline fun BooleanArray.asList(): List<Boolean> {\n    return\n    this.unsafeCast<Array<Boolean>>().asList()\n}\n\n/**\n * Returns a [List] that wraps the original array.\n */\npublic actual fun CharArray.asList(): List<Char> {\n    return object : AbstractList<Char>(), RandomAccess {\n        override val size: Int get() = this@asList.size\n        override fun isEmpty(): Boolean = this@asList.isEmpty()\n        override fun contains(element: Char): Boolean = this@asList.contains(element)\n        override fun get(index: Int): Char {\n            AbstractList.checkElementIndex(index, size)\n            return this@asList[index]\n        }\n        override fun indexOf(element: Char): Int {\n            @Suppress("USELESS_CAST")\n            if ((element as Any?) !is Char) return -1\n            return this@asList.indexOf(element)\n        }\n        override fun\n        lastIndexOf(element: Char): Int {\n            @Suppress("USELESS_CAST")\n            if ((element as Any?) !is Char) return -1\n            return this@asList.lastIndexOf(element)\n        }\n    }\n}\n\n/**\n * Returns `true` if the\n * two specified arrays are *deeply* equal to one another,\n * i.e. contain the same number of the same elements in the\n * same order.\n * If two corresponding elements are nested arrays, they are also compared deeply.\n * If any of\n * arrays contains itself on any nesting level the behavior is undefined.\n * The elements of other types are\n * compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is\n * equal to itself and `-0.0` is not equal to `0.0`.\n */\n@SinceKotlin("1.1")\n@kotlin.internal.LowPriorityInOverloadResolution\npublic actual infix fun <T>\nArray<out T>.contentDeepEquals(other: Array<out T>): Boolean {\n    return\n    this.contentDeepEquals(other)\n}\n\n/**\n * Returns `true` if the two specified arrays are *deeply* equal to one\n * another,\n * i.e. contain the same number of the same elements in the same order.\n * The specified arrays are\n * also considered deeply equal if both are `null`.\n * If two corresponding elements are nested arrays, they are\n * also compared deeply.\n * If any of arrays contains itself on any nesting level the behavior is undefined.\n * The\n * elements of other types are compared for equality with the [equals][Any.equals] function.\n * For floating point\n * numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n */\n@SinceKotlin("1.4")\n@library("arrayDeepEquals")\npublic actual infix fun <T> Array<out\nT>?.contentDeepEquals(other: Array<out T>?): Boolean {\n    definedExternally\n}\n\n/**\n * Returns a hash code\n * based on the contents of this array as if it is [List].\n * Nested arrays are treated as lists too.\n * If any of arrays\n * contains itself on any nesting level the behavior is undefined.\n */\n@SinceKotlin("1.1")\n@kotlin.internal.LowPriorityInOverloadResolution\npublic actual fun <T> Array<out\nT>.contentDeepHashCode(): Int {\n    return this.contentDeepHashCode()\n}\n\n/**\n * Returns a hash code based\n * on the contents of this array as if it is [List].\n * Nested arrays are treated as lists too.\n * If any of arrays\n * contains itself on any nesting level the behavior is undefined.\n */\n@SinceKotlin("1.4")\n@library("arrayDeepHashCode")\npublic actual fun <T> Array<out\nT>?.contentDeepHashCode(): Int {\n    definedExternally\n}\n\n/**\n * Returns a string representation of the\n * contents of this array as if it is a [List].\n * Nested arrays are treated as lists too.\n * If any of arrays contains\n * itself on any nesting level that reference\n * is rendered as `[...]` to prevent recursion.\n * @sample\n * samples.collections.Arrays.ContentOperations.contentDeepToString\n */\n@SinceKotlin("1.1")\n@kotlin.internal.LowPriorityInOverloadResolution\npublic actual fun <T> Array<out\nT>.contentDeepToString(): String {\n    return this.contentDeepToString()\n}\n\n/**\n * Returns a string\n * representation of the contents of this array as if it is a [List].\n * Nested arrays are treated as lists too.\n * If any

```



```

*\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.\n")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic actual infix fun
BooleanArray.contentEquals(other: BooleanArray): Boolean {\n    return this.contentEquals(other)\n}\n\n**\n *
Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of
the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals]
function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n
*\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.\n")\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic actual infix fun
CharArray.contentEquals(other: CharArray): Boolean {\n    return this.contentEquals(other)\n}\n\n**\n * Returns
`true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the
same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals]
function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n
*\n@SinceKotlin("1.4")\n@library("arrayEquals")\npublic actual infix fun <T> Array<out
T>?.contentEquals(other: Array<out T>?): Boolean {\n    definedExternally\n}\n\n**\n * Returns `true` if the two
specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the
same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating
point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n
*\n@SinceKotlin("1.4")\n@library("arrayEquals")\npublic actual infix fun ByteArray?.contentEquals(other:
ByteArray?): Boolean {\n    definedExternally\n}\n\n**\n * Returns `true` if the two specified arrays are
*structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n
* The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it
means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n
*\n@SinceKotlin("1.4")\n@library("arrayEquals")\npublic actual infix fun ShortArray?.contentEquals(other:
ShortArray?): Boolean {\n    definedExternally\n}\n\n**\n * Returns `true` if the two specified arrays are
*structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n
* The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it
means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n
*\n@SinceKotlin("1.4")\n@library("arrayEquals")\npublic actual infix fun IntArray?.contentEquals(other:
IntArray?): Boolean {\n    definedExternally\n}\n\n**\n * Returns `true` if the two specified arrays are
*structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n
* The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it
means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n
*\n@SinceKotlin("1.4")\n@library("arrayEquals")\npublic actual infix fun LongArray?.contentEquals(other:
LongArray?): Boolean {\n    definedExternally\n}\n\n**\n * Returns `true` if the two specified arrays are
*structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n
* The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it
means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n
*\n@SinceKotlin("1.4")\n@library("arrayEquals")\npublic actual infix fun FloatArray?.contentEquals(other:
FloatArray?): Boolean {\n    definedExternally\n}\n\n**\n * Returns `true` if the two specified arrays are
*structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n
* The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it
means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n
*\n@SinceKotlin("1.4")\n@library("arrayEquals")\npublic actual infix fun DoubleArray?.contentEquals(other:
DoubleArray?): Boolean {\n    definedExternally\n}\n\n**\n * Returns `true` if the two specified arrays are
*structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n
* The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it
means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n
*\n@SinceKotlin("1.4")\n@library("arrayEquals")\npublic actual infix fun BooleanArray?.contentEquals(other:

```

BooleanArray?): Boolean {\n definedExternally\n}\n\n/**\n * Returns `true` if the two specified arrays are
structurally equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n
* The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it
means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n
*\n\n@SinceKotlin("1.4")\n\n@library("arrayEquals")\n\npublic actual infix fun CharArray?.contentEquals(other:
CharArray?): Boolean {\n definedExternally\n}\n\n/**\n * Returns a hash code based on the contents of this array
as if it is [List].\n * \n\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")\n\n@SinceKotlin("1.1")\n\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\n\npublic actual fun <T>
Array<out T>.contentHashCode(): Int {\n return this.contentHashCode()\n}\n\n/**\n * Returns a hash code based
on the contents of this array as if it is [List].\n * \n\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")\n\n@SinceKotlin("1.1")\n\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\n\npublic actual fun
ByteArray.contentHashCode(): Int {\n return this.contentHashCode()\n}\n\n/**\n * Returns a hash code based on
the contents of this array as if it is [List].\n * \n\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")\n\n@SinceKotlin("1.1")\n\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\n\npublic actual fun
ShortArray.contentHashCode(): Int {\n return this.contentHashCode()\n}\n\n/**\n * Returns a hash code based on
the contents of this array as if it is [List].\n * \n\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")\n\n@SinceKotlin("1.1")\n\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\n\npublic actual fun
IntArray.contentHashCode(): Int {\n return this.contentHashCode()\n}\n\n/**\n * Returns a hash code based on
the contents of this array as if it is [List].\n * \n\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")\n\n@SinceKotlin("1.1")\n\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\n\npublic actual fun
LongArray.contentHashCode(): Int {\n return this.contentHashCode()\n}\n\n/**\n * Returns a hash code based on
the contents of this array as if it is [List].\n * \n\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")\n\n@SinceKotlin("1.1")\n\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\n\npublic actual fun
FloatArray.contentHashCode(): Int {\n return this.contentHashCode()\n}\n\n/**\n * Returns a hash code based on
the contents of this array as if it is [List].\n * \n\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")\n\n@SinceKotlin("1.1")\n\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\n\npublic actual fun
DoubleArray.contentHashCode(): Int {\n return this.contentHashCode()\n}\n\n/**\n * Returns a hash code based
on the contents of this array as if it is [List].\n * \n\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")\n\n@SinceKotlin("1.1")\n\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\n\npublic actual fun
BooleanArray.contentHashCode(): Int {\n return this.contentHashCode()\n}\n\n/**\n * Returns a hash code based
on the contents of this array as if it is [List].\n * \n\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")\n\n@SinceKotlin("1.1")\n\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\n\npublic actual fun
CharArray.contentHashCode(): Int {\n return this.contentHashCode()\n}\n\n/**\n * Returns a hash code based on
the contents of this array as if it is [List].\n * \n\n@SinceKotlin("1.4")\n\n@library("arrayHashCode")\n\npublic actual
fun <T> Array<out T>?.contentHashCode(): Int {\n definedExternally\n}\n\n/**\n * Returns a hash code based on
the contents of this array as if it is [List].\n * \n\n@SinceKotlin("1.4")\n\n@library("arrayHashCode")\n\npublic actual
fun ByteArray?.contentHashCode(): Int {\n definedExternally\n}\n\n/**\n * Returns a hash code based on the
contents of this array as if it is [List].\n * \n\n@SinceKotlin("1.4")\n\n@library("arrayHashCode")\n\npublic actual fun
ShortArray?.contentHashCode(): Int {\n definedExternally\n}\n\n/**\n * Returns a hash code based on the
contents of this array as if it is [List].\n * \n\n@SinceKotlin("1.4")\n\n@library("arrayHashCode")\n\npublic actual fun
IntArray?.contentHashCode(): Int {\n definedExternally\n}\n\n/**\n * Returns a hash code based on the contents
of this array as if it is [List].\n * \n\n@SinceKotlin("1.4")\n\n@library("arrayHashCode")\n\npublic actual fun
LongArray?.contentHashCode(): Int {\n definedExternally\n}\n\n/**\n * Returns a hash code based on the
contents of this array as if it is [List].\n * \n\n@SinceKotlin("1.4")\n\n@library("arrayHashCode")\n\npublic actual fun
FloatArray?.contentHashCode(): Int {\n definedExternally\n}\n\n/**\n * Returns a hash code based on the
contents of this array as if it is [List].\n * \n\n@SinceKotlin("1.4")\n\n@library("arrayHashCode")\n\npublic actual fun
DoubleArray?.contentHashCode(): Int {\n definedExternally\n}\n\n/**\n * Returns a hash code based on the
contents of this array as if it is [List].\n * \n\n@SinceKotlin("1.4")\n\n@library("arrayHashCode")\n\npublic actual fun


```

{\n  definedExternally\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is
[List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n
*\n@SinceKotlin("1.4")\n@library("arrayToString")\npublic actual fun IntArray?.contentToString(): String {\n
definedExternally\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is
[List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n
*\n@SinceKotlin("1.4")\n@library("arrayToString")\npublic actual fun LongArray?.contentToString(): String
{\n  definedExternally\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is
[List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n
*\n@SinceKotlin("1.4")\n@library("arrayToString")\npublic actual fun FloatArray?.contentToString(): String
{\n  definedExternally\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is
[List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n
*\n@SinceKotlin("1.4")\n@library("arrayToString")\npublic actual fun DoubleArray?.contentToString(): String
{\n  definedExternally\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is
[List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n
*\n@SinceKotlin("1.4")\n@library("arrayToString")\npublic actual fun BooleanArray?.contentToString():
String {\n  definedExternally\n}\n\n/**\n * Returns a string representation of the contents of the specified array as
if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n
*\n@SinceKotlin("1.4")\n@library("arrayToString")\npublic actual fun CharArray?.contentToString(): String
{\n  definedExternally\n}\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that
array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it
overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset
the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of
the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this
array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex]
or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws
IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified
[destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the
[destination] array.\n
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT
_ARGUMENTS")\npublic actual inline fun <T> Array<out T>.copyInto(destination: Array<T>, destinationOffset:
Int = 0, startIndex: Int = 0, endIndex: Int = size): Array<T> {\n  arrayCopy(this, destination, destinationOffset,
startIndex, endIndex)\n  return destination\n}\n\n/**\n * Copies this array or its subrange into the [destination]
array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the
subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n *
@param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the
beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the
subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or
[IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex
> endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array
starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices
range.\n * \n * @return the [destination] array.\n
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT
_ARGUMENTS")\npublic actual inline fun ByteArray.copyInto(destination: ByteArray, destinationOffset: Int = 0,
startIndex: Int = 0, endIndex: Int = size): ByteArray {\n  arrayCopy(this.unsafeCast<Array<Byte>>(),
destination.unsafeCast<Array<Byte>>(), destinationOffset, startIndex, endIndex)\n  return destination\n}\n\n/**\n
* Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to
pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination
range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the
[destination] array to

```

copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n

```
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual inline fun ShortArray.copyInto(destination: ShortArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): ShortArray {\n    arrayCopy(this.unsafeCast<Array<Short>>(), destination.unsafeCast<Array<Short>>(), destinationOffset, startIndex, endIndex)\n    return destination\n}\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n
```

```
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual inline fun IntArray.copyInto(destination: IntArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): IntArray {\n    arrayCopy(this.unsafeCast<Array<Int>>(), destination.unsafeCast<Array<Int>>(), destinationOffset, startIndex, endIndex)\n    return destination\n}\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n
```

```
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual inline fun LongArray.copyInto(destination: LongArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): LongArray {\n    arrayCopy(this.unsafeCast<Array<Long>>(), destination.unsafeCast<Array<Long>>(), destinationOffset, startIndex, endIndex)\n    return destination\n}\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n
```

```
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual inline fun FloatArray.copyInto(destination: FloatArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): FloatArray {\n    arrayCopy(this.unsafeCast<Array<Float>>(), destination.unsafeCast<Array<Float>>(), destinationOffset, startIndex, endIndex)\n    return destination\n}\n
```

```

destination.unsafeCast<Array<Float>>(), destinationOffset, startIndex, endIndex)\n  return destination\n}\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n\n*/\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual inline fun DoubleArray.copyInto(destination: DoubleArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): DoubleArray {\n  arrayCopy(this.unsafeCast<Array<Double>>(), destination.unsafeCast<Array<Double>>(), destinationOffset, startIndex, endIndex)\n  return destination\n}\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n\n*/\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual inline fun BooleanArray.copyInto(destination: BooleanArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): BooleanArray {\n  arrayCopy(this.unsafeCast<Array<Boolean>>(), destination.unsafeCast<Array<Boolean>>(), destinationOffset, startIndex, endIndex)\n  return destination\n}\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n\n*/\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual inline fun CharArray.copyInto(destination: CharArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): CharArray {\n  arrayCopy(this.unsafeCast<Array<Char>>(), destination.unsafeCast<Array<Char>>(), destinationOffset, startIndex, endIndex)\n  return destination\n}\n\n/**\n * Returns new array which is a copy of the original array.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.copyOf\n\n*/\n@Suppress("ACTUAL_WITHOUT_EXPECT", "NOTHING_TO_INLINE")\npublic actual inline fun <T> Array<out T>.copyOf(): Array<T> {\n  return this.asDynamic().slice()\n}\n\n/**\n * Returns new array which is a copy of the original array.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.copyOf\n\n*/\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline fun ByteArray.copyOf(): ByteArray {\n  return this.asDynamic().slice()\n}\n\n/**\n * Returns new array which is a copy of the original array.\n * \n * @sample

```

```

samples.collections.Arrays.CopyOfOperations.copyOfOf\n *\n@Suppress(\\"NOTHING_TO_INLINE\\")\npublic
actual inline fun ShortArray.copyOfOf(): ShortArray {\n  return this.asDynamic().slice()\n}\n\n/**\n * Returns new
array which is a copy of the original array.\n * \n * @sample
samples.collections.Arrays.CopyOfOperations.copyOfOf\n *\n@Suppress(\\"NOTHING_TO_INLINE\\")\npublic
actual inline fun IntArray.copyOfOf(): IntArray {\n  return this.asDynamic().slice()\n}\n\n/**\n * Returns new array
which is a copy of the original array.\n * \n * @sample
samples.collections.Arrays.CopyOfOperations.copyOfOf\n *\n@Suppress(\\"NOTHING_TO_INLINE\\")\npublic
actual inline fun LongArray.copyOfOf(): LongArray {\n  return withType(\\"LongArray\\",
this.asDynamic().slice())\n}\n\n/**\n * Returns new array which is a copy of the original array.\n * \n * @sample
samples.collections.Arrays.CopyOfOperations.copyOfOf\n *\n@Suppress(\\"NOTHING_TO_INLINE\\")\npublic
actual inline fun FloatArray.copyOfOf(): FloatArray {\n  return this.asDynamic().slice()\n}\n\n/**\n * Returns new
array which is a copy of the original array.\n * \n * @sample
samples.collections.Arrays.CopyOfOperations.copyOfOf\n *\n@Suppress(\\"NOTHING_TO_INLINE\\")\npublic
actual inline fun DoubleArray.copyOfOf(): DoubleArray {\n  return this.asDynamic().slice()\n}\n\n/**\n * Returns
new array which is a copy of the original array.\n * \n * @sample
samples.collections.Arrays.CopyOfOperations.copyOfOf\n *\npublic actual fun BooleanArray.copyOfOf():
BooleanArray {\n  return withType(\\"BooleanArray\\", this.asDynamic().slice())\n}\n\n/**\n * Returns new array
which is a copy of the original array.\n * \n * @sample
samples.collections.Arrays.CopyOfOperations.copyOfOf\n *\npublic actual fun CharArray.copyOfOf(): CharArray {\n
return withType(\\"CharArray\\",
this.asDynamic().slice())\n}\n\n/**\n * Returns new array which is a copy of the original array, resized to the given
[newSize].\n * The copy is either truncated or padded at the end with zero values if necessary.\n * \n * - If [newSize]
is less than the size of the original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater
than the size of the original array, the extra elements in the copy array are filled with zero values.\n * \n * @sample
samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n *\npublic actual fun
ByteArray.copyOfOf(newSize: Int): ByteArray {\n  require(newSize >= 0) { \\"Invalid new array size: $newSize.\\\"
}\n  return fillFrom(this, ByteArray(newSize))\n}\n\n/**\n * Returns new array which is a copy of the original
array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with zero values if
necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array is truncated to the
[newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are
filled with zero values.\n * \n * @sample
samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n *\npublic actual fun
ShortArray.copyOfOf(newSize: Int): ShortArray {\n  require(newSize >= 0) { \\"Invalid new
array size: $newSize.\\\" }\n  return fillFrom(this, ShortArray(newSize))\n}\n\n/**\n * Returns new array which is a
copy of the original array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with
zero values if necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array is truncated
to the [newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the copy array
are filled with zero values.\n * \n * @sample
samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n *\npublic actual fun
IntArray.copyOfOf(newSize: Int): IntArray {\n  require(newSize >= 0) { \\"Invalid new array size: $newSize.\\\" }\n
return fillFrom(this, IntArray(newSize))\n}\n\n/**\n * Returns new array which is a copy of the original array,
resized to the given [newSize].\n * The copy is either truncated or padded at the end with zero values if necessary.\n
*\n * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n * - If
[newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero
values.\n * \n * @sample
samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n *\npublic actual
fun LongArray.copyOfOf(newSize: Int): LongArray {\n  require(newSize >= 0) { \\"Invalid new array size:
$newSize.\\\" }\n  return withType(\\"LongArray\\", arrayCopyResize(this, newSize, 0L))\n}\n\n/**\n * Returns new
array which is a copy of the original array, resized to the given [newSize].\n * The copy is either truncated or padded
at the end with zero values if necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy
array is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original array, the extra
elements in the copy array are filled with zero values.\n * \n * @sample

```

```

samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n * \npublic actual fun
FloatArray.copyOf(newSize: Int): FloatArray {\n  require(newSize >= 0) { \"Invalid new array size: $newSize.\"
}\n  return fillFrom(this, FloatArray(newSize))\n}\n\n/**\n * Returns new array which is a copy of the original
array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with zero values if
necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array is truncated to the
[newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are
filled with zero values.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n
*\npublic actual fun DoubleArray.copyOf(newSize: Int): DoubleArray {\n  require(newSize >= 0) { \"Invalid new
array size: $newSize.\" }\n  return fillFrom(this, DoubleArray(newSize))\n}\n\n/**\n * Returns new array which is
a copy of the original array, resized to the given [newSize].\n * The copy is either truncated or padded at the end
with `false` values if necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array is
truncated to the [newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the
copy array are filled with `false` values.\n * \n * @sample
samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n * \npublic actual fun
BooleanArray.copyOf(newSize: Int): BooleanArray {\n  require(newSize >= 0) { \"Invalid new array size:
$newSize.\" }\n  return withType(\"BooleanArray\", arrayCopyResize(this, newSize, false))\n}\n\n/**\n * Returns
new array which is a copy of the original array, resized to the given [newSize].\n * The copy is either truncated or
padded at the end with null char (`\u0000`) values if necessary.\n * \n * - If [newSize] is less than the size of the
original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original
array, the extra elements in the copy array are filled with null char (`\u0000`) values.\n * \n * @sample
samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n * \npublic actual fun
CharArray.copyOf(newSize: Int): CharArray {\n  require(newSize >= 0) { \"Invalid new array size: $newSize.\"
}\n  return withType(\"CharArray\", fillFrom(this, CharArray(newSize)))\n}\n\n/**\n * Returns new array which is
a copy of the original array, resized to the given [newSize].\n * The copy is either truncated or padded at the end
with `null` values if necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array is
truncated to the [newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the
copy array are filled with `null` values.\n * \n * @sample
samples.collections.Arrays.CopyOfOperations.resizingCopyOf\n
*\n@Suppress(\"ACTUAL_WITHOUT_EXPECT\")\npublic actual fun <T> Array<out T>.copyOf(newSize: Int):
Array<T?> {\n  require(newSize >= 0) { \"Invalid new array size: $newSize.\" }\n  return arrayCopyResize(this,
newSize, null)\n}\n\n/**\n * Returns a new array which is a copy of the specified range of the original array.\n * \n
*\n * @param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive)
to copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than
the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*\n@Suppress(\"ACTUAL_WITHOUT_EXPECT\")\npublic actual fun <T> Array<out
T>.copyOfRange(fromIndex: Int, toIndex: Int): Array<T> {\n  AbstractList.checkRangeIndexes(fromIndex,
toIndex, size)\n  return this.asDynamic().slice(fromIndex, toIndex)\n}\n\n/**\n * Returns a new array which is a
copy of the specified range of the original array.\n * \n * \n * @param fromIndex the start of the range (inclusive) to
copy.\n * @param toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException
if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws
IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \npublic actual fun
ByteArray.copyOfRange(fromIndex: Int, toIndex: Int): ByteArray {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n  return this.asDynamic().slice(fromIndex,
toIndex)\n}\n\n/**\n * Returns a new array which is a copy of the specified range of the original array.\n * \n
*\n * @param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to
copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the
size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \npublic
actual fun ShortArray.copyOfRange(fromIndex: Int, toIndex: Int): ShortArray {\n

```

```

AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    return this.asDynamic().slice(fromIndex,
toIndex)\n}\n\n/**\n * Returns a new array which is a copy of the specified range of the original array.\n * \n *
@param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to
copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the
size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n\npublic
actual fun IntArray.copyOfRange(fromIndex: Int, toIndex: Int): IntArray {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    return this.asDynamic().slice(fromIndex,
toIndex)\n}\n\n/**\n * Returns a new array which is a copy of the specified range of the original array.\n * \n *
@param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to
copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the
size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n\npublic
actual fun LongArray.copyOfRange(fromIndex: Int, toIndex: Int): LongArray {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    return withType("<code>"LongArray"</code>",
this.asDynamic().slice(fromIndex, toIndex))\n}\n\n/**\n * Returns a new array which is a copy of the specified
range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n * @param
toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is
less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if
[fromIndex] is greater than [toIndex].\n */\n\npublic actual fun FloatArray.copyOfRange(fromIndex: Int, toIndex: Int):
FloatArray {\n    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    return
this.asDynamic().slice(fromIndex, toIndex)\n}\n\n/**\n * Returns a new array which is a copy of the specified range
of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the
end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero
or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater
than [toIndex].\n */\n\npublic actual fun DoubleArray.copyOfRange(fromIndex: Int, toIndex: Int): DoubleArray {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    return this.asDynamic().slice(fromIndex,
toIndex)\n}\n\n/**\n * Returns a new array which is a copy of the specified range of the original array.\n * \n *
@param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to
copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the
size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n\npublic
actual fun BooleanArray.copyOfRange(fromIndex: Int, toIndex: Int): BooleanArray {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    return withType("<code>"BooleanArray"</code>",
this.asDynamic().slice(fromIndex, toIndex))\n}\n\n/**\n * Returns a new array which is a copy of the specified
range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n * @param
toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is
less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if
[fromIndex] is greater than [toIndex].\n */\n\npublic actual fun CharArray.copyOfRange(fromIndex: Int, toIndex: Int):
CharArray {\n    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    return withType("<code>"CharArray"</code>",
this.asDynamic().slice(fromIndex, toIndex))\n}\n\n/**\n * Fills this array or its subrange with the specified
[element] value.\n * \n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param
toIndex the end of the range (exclusive) to fill, size of this array by default.\n * \n * @throws
IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n *
@throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n\npublic
actual fun <T> Array<T>.fill(element: T, fromIndex: Int = 0, toIndex: Int = size): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    this.asDynamic().fill(element, fromIndex,
toIndex);\n}\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param
fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive)
to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero

```

or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n

```

*\n@SinceKotlin("1.3")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic
actual fun ByteArray.fill(element: Byte, fromIndex: Int = 0, toIndex: Int = size): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    this.asDynamic().fill(element, fromIndex,
toIndex);\n}\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param
fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive)
to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero
or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater
than [toIndex].\n
*\n@SinceKotlin("1.3")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic
actual fun ShortArray.fill(element: Short, fromIndex: Int = 0, toIndex: Int = size): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    this.asDynamic().fill(element, fromIndex,
toIndex);\n}\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param
fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive)
to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero
or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater
than [toIndex].\n
*\n@SinceKotlin("1.3")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic
actual fun IntArray.fill(element: Int, fromIndex: Int = 0, toIndex: Int = size): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    this.asDynamic().fill(element, fromIndex,
toIndex);\n}\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param
fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive)
to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero
or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater
than [toIndex].\n
*\n@SinceKotlin("1.3")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic
actual fun LongArray.fill(element: Long, fromIndex: Int = 0, toIndex: Int = size): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    this.asDynamic().fill(element, fromIndex,
toIndex);\n}\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param
fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive)
to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero
or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater
than [toIndex].\n
*\n@SinceKotlin("1.3")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic
actual fun FloatArray.fill(element: Float, fromIndex: Int = 0, toIndex: Int = size): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    this.asDynamic().fill(element, fromIndex,
toIndex);\n}\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param
fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive)
to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero
or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater
than [toIndex].\n
*\n@SinceKotlin("1.3")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic
actual fun DoubleArray.fill(element: Double, fromIndex: Int = 0, toIndex: Int = size): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    this.asDynamic().fill(element, fromIndex,
toIndex);\n}\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param
fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive)
to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero
or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater
than [toIndex].\n

```

or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n

```

*\n@SinceKotlin("1.3")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic
actual fun BooleanArray.fill(element: Boolean, fromIndex: Int = 0, toIndex: Int = size): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    this.asDynamic().fill(element, fromIndex,
toIndex);\n}\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param
fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive)
to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero
or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater
than [toIndex].\n
*\n@SinceKotlin("1.3")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic
actual fun CharArray.fill(element: Char, fromIndex: Int = 0, toIndex: Int = size): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    this.asDynamic().fill(element, fromIndex,
toIndex);\n}\n\n/**\n * Returns an array containing all elements of the original array and then the given [element].\n
*\n@Suppress("ACTUAL_WITHOUT_EXPECT", "NOTHING_TO_INLINE")\npublic actual inline operator
fun <T> Array<out T>.plus(element: T): Array<T> {\n    return
this.asDynamic().concat(arrayOf(element))\n}\n\n/**\n * Returns an array containing all elements of the original
array and then the given [element].\n *\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator
fun ByteArray.plus(element: Byte): ByteArray {\n    return plus(byteArrayOf(element))\n}\n\n/**\n * Returns an
array containing all elements of the original array and then the given [element].\n
*\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun ShortArray.plus(element: Short):
ShortArray {\n    return plus(shortArrayOf(element))\n}\n\n/**\n * Returns an array containing all elements of the
original array and then the given [element].\n *\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline
operator fun IntArray.plus(element: Int): IntArray {\n    return plus(intArrayOf(element))\n}\n\n/**\n * Returns an
array containing all elements of the original array and then the given [element].\n
*\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun LongArray.plus(element: Long):
LongArray {\n    return plus(longArrayOf(element))\n}\n\n/**\n * Returns an array containing all elements of the
original array and then the given [element].\n *\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline
operator fun FloatArray.plus(element: Float): FloatArray {\n    return plus(floatArrayOf(element))\n}\n\n/**\n *
Returns an array containing all elements of the original array and then the given [element].\n
*\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun DoubleArray.plus(element:
Double): DoubleArray {\n    return plus(doubleArrayOf(element))\n}\n\n/**\n * Returns an array containing all
elements of the original array and then the given [element].\n *\n@Suppress("NOTHING_TO_INLINE")\npublic
actual inline operator fun BooleanArray.plus(element: Boolean): BooleanArray {\n    return
plus(booleanArrayOf(element))\n}\n\n/**\n * Returns an array containing all elements of the original array and then
the given [element].\n *\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun
CharArray.plus(element: Char): CharArray {\n    return plus(charArrayOf(element))\n}\n\n/**\n * Returns an array
containing all elements of the original array and then all elements of the given [elements] collection.\n
*\n@Suppress("ACTUAL_WITHOUT_EXPECT")\npublic actual operator fun <T> Array<out T>.plus(elements:
Collection<T>): Array<T> {\n    return arrayPlusCollection(this, elements)\n}\n\n/**\n * Returns an array
containing all elements of the original array and then all elements of the given [elements] collection.\n *\n@public
actual operator fun ByteArray.plus(elements: Collection<Byte>): ByteArray {\n    return
fillFromCollection(this.copyOf(size + elements.size), this.size, elements)\n}\n\n/**\n * Returns an array containing
all elements of the original array and then all elements of the given [elements] collection.\n *\n@public actual
operator fun ShortArray.plus(elements: Collection<Short>): ShortArray {\n    return
fillFromCollection(this.copyOf(size + elements.size), this.size, elements)\n}\n\n/**\n * Returns an array containing
all elements of the original array and then all elements of the given [elements] collection.\n *\n@public actual
operator fun IntArray.plus(elements: Collection<Int>): IntArray {\n    return fillFromCollection(this.copyOf(size +

```

elements.size), this.size, elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n */\npublic actual operator fun LongArray.plus(elements: Collection<Long>): LongArray {\n return arrayPlusCollection(this, elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n */\npublic actual operator fun FloatArray.plus(elements: Collection<Float>): FloatArray {\n return fillFromCollection(this.copyOf(size + elements.size), this.size, elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n */\npublic actual operator fun DoubleArray.plus(elements: Collection<Double>): DoubleArray {\n return fillFromCollection(this.copyOf(size + elements.size), this.size, elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n */\npublic actual operator fun BooleanArray.plus(elements: Collection<Boolean>): BooleanArray {\n return arrayPlusCollection(this, elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n */\npublic actual operator fun CharArray.plus(elements: Collection<Char>): CharArray {\n return fillFromCollection(this.copyOf(size + elements.size), this.size, elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\n@Suppress("ACTUAL_WITHOUT_EXPECT", "NOTHING_TO_INLINE")\npublic actual inline operator fun <T> Array<out T>.plus(elements: Array<out T>): Array<T> {\n return this.asDynamic().concat(elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun ByteArray.plus(elements: ByteArray): ByteArray {\n return primitiveArrayConcat(this, elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun ShortArray.plus(elements: ShortArray): ShortArray {\n return primitiveArrayConcat(this, elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun IntArray.plus(elements: IntArray): IntArray {\n return primitiveArrayConcat(this, elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun LongArray.plus(elements: LongArray): LongArray {\n return primitiveArrayConcat(this, elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun FloatArray.plus(elements: FloatArray): FloatArray {\n return primitiveArrayConcat(this, elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun DoubleArray.plus(elements: DoubleArray): DoubleArray {\n return primitiveArrayConcat(this, elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun BooleanArray.plus(elements: BooleanArray): BooleanArray {\n return primitiveArrayConcat(this, elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then the given [element].\n */\n@Suppress("ACTUAL_WITHOUT_EXPECT", "NOTHING_TO_INLINE")\npublic actual inline fun <T> Array<out T>.addElement(element: T): Array<T> {\n return this.asDynamic().concat(arrayOf(element))\n}\n\n/**\n * Sorts the array in-place.\n */\n@sample\nsamples.collections.Arrays.Sorting.sortArray\n\n/**\n * @library("primitiveArraySort")\n */\npublic actual fun IntArray.sort(): Unit {\n definedExternally\n}\n\n/**\n * Sorts the array in-place.\n */\n@sample

```

samples.collections.Arrays.Sorting.sortArray\n *\\npublic actual fun LongArray.sort(): Unit {\n
@Suppress("\\DEPRECATION\\")\n if (size > 1) sort { a: Long, b: Long -> a.compareTo(b) }\n}\n\n/**\n * Sorts
the array in-place.\n * \n * @sample samples.collections.Arrays.Sorting.sortArray\n
*\n@library("\\primitiveArraySort\\")\npublic actual fun ByteArray.sort(): Unit {\n definedExternally\n}\n\n/**\n
* Sorts the array in-place.\n * \n * @sample samples.collections.Arrays.Sorting.sortArray\n
*\n@library("\\primitiveArraySort\\")\npublic actual fun ShortArray.sort(): Unit {\n definedExternally\n}\n\n/**\n
* Sorts the array in-place.\n * \n * @sample samples.collections.Arrays.Sorting.sortArray\n
*\n@library("\\primitiveArraySort\\")\npublic actual fun DoubleArray.sort(): Unit {\n
definedExternally\n}\n\n/**\n * Sorts the array in-place.\n * \n * @sample
samples.collections.Arrays.Sorting.sortArray\n *\n@library("\\primitiveArraySort\\")\npublic actual fun
FloatArray.sort(): Unit {\n definedExternally\n}\n\n/**\n * Sorts the array in-place.\n * \n * @sample
samples.collections.Arrays.Sorting.sortArray\n *\n@library("\\primitiveArraySort\\")\npublic actual fun
CharArray.sort(): Unit {\n definedExternally\n}\n\n/**\n * Sorts the array in-place according to the natural order
of its elements.\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other
after sorting.\n * \n * @sample samples.collections.Arrays.Sorting.sortArrayOfComparable\n *\npublic actual fun
<T : Comparable<T>> Array<out T>.sort(): Unit {\n if (size > 1) sortArray(this)\n}\n\n/**\n * Sorts the array in-
place according to the order specified by the given [comparison] function.\n * \n * The sort is _stable_. It means that
equal elements preserve their order relative to each other after sorting.\n * \n@Deprecated("\\Use sortWith instead",
ReplaceWith("\\this.sortWith(Comparator(comparison))"))\n@DeprecatedSinceKotlin(warningSince =
"1.6")\npublic fun <T> Array<out T>.sort(comparison: (a: T, b: T) -> Int): Unit {\n if (size > 1)
sortArrayWith(this, comparison)\n}\n\n/**\n * Sorts a range in the array in-place.\n * \n * The sort is _stable_. It
means that equal elements preserve their order relative to each other after sorting.\n * \n * @param fromIndex the
start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size
of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex]
is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than
[toIndex].\n * \n * @sample samples.collections.Arrays.Sorting.sortRangeOfArrayOfComparable\n
*\n@SinceKotlin("1.4")\n@Suppress("\\ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\\")\npublic
actual fun <T : Comparable<T>> Array<out T>.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n sortArrayWith(this, fromIndex, toIndex,
naturalOrder())\n}\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex the start of the range
(inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by
default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than
the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n *
@sample samples.collections.Arrays.Sorting.sortRangeOfArray\n
*\n@SinceKotlin("1.4")\n@Suppress("\\ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\\")\npublic
actual fun ByteArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n val subarray =
this.asDynamic().subarray(fromIndex, toIndex).unsafeCast<ByteArray>()\n subarray.sort()\n}\n\n/**\n * Sorts a
range in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n *
@param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * \n * @throws
IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n *
@throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample
samples.collections.Arrays.Sorting.sortRangeOfArray\n
*\n@SinceKotlin("1.4")\n@Suppress("\\ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\\")\npublic
actual fun ShortArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n val subarray =
this.asDynamic().subarray(fromIndex, toIndex).unsafeCast<ShortArray>()\n subarray.sort()\n}\n\n/**\n * Sorts a
range in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n *

```



```

nullsFirst(comparator: Comparator<in T>): Comparator<T?> =\n    Comparator { a, b ->\n        when {\n            a\n            === b -> 0\n            a == null -> -1\n            b == null -> 1\n            else -> comparator.compare(a, b)\n        }\n    }\n\n/**\n * Provides a comparator of nullable [Comparable] values\n * considering `null` value less than any other\n * value.\n */\n * @sample samples.comparisons.Comparisons.nullsFirstLastComparator\n\n*\n@kotlin.internal.InlineOnly\npublic inline fun <T : Comparable<T>> nullsFirst(): Comparator<T?> =\n    nullsFirst(naturalOrder())\n\n/**\n * Extends the given [comparator] of non-nullable values to a comparator of\n * nullable values\n * considering `null` value greater than any other value.\n */\n * @sample\n * samples.comparisons.Comparisons.nullsFirstLastWithComparator\n\n*\npublic fun <T : Any>\nnullsLast(comparator: Comparator<in T>): Comparator<T?> =\n    Comparator { a, b ->\n        when {\n            a\n            === b -> 0\n            a == null -> 1\n            b == null -> -1\n            else -> comparator.compare(a, b)\n        }\n    }\n\n/**\n * Provides a comparator of nullable [Comparable] values\n * considering `null` value greater than any\n * other value.\n */\n * @sample samples.comparisons.Comparisons.nullsFirstLastComparator\n\n*\n@kotlin.internal.InlineOnly\npublic inline fun <T : Comparable<T>> nullsLast(): Comparator<T?> =\n    nullsLast(naturalOrder())\n\n/**\n * Returns a comparator that compares [Comparable] objects in natural order.\n */\n * @sample samples.comparisons.Comparisons.naturalOrderComparator\n\n*\npublic fun <T : Comparable<T>>\nnaturalOrder(): Comparator<T> = @Suppress("UNCHECKED_CAST") (NaturalOrderComparator as\n    Comparator<T>)\n\n/**\n * Returns a comparator that compares [Comparable] objects in reversed natural order.\n */\n * @sample samples.comparisons.Comparisons.nullsFirstLastWithComparator\n\n*\npublic fun <T :\n    Comparable<T>> reverseOrder(): Comparator<T> = @Suppress("UNCHECKED_CAST")\n    (ReverseOrderComparator as Comparator<T>)\n\n/**\n * Returns a comparator that imposes the reverse ordering\n * of this comparator.\n */\n * @sample samples.comparisons.Comparisons.reversed\n\n*\n@Suppress("EXTENSION_SHADOWED_BY_MEMBER")\npublic fun <T> Comparator<T>.reversed():\n    Comparator<T> = when (this) {\n        is ReversedComparator -> this.comparator\n        NaturalOrderComparator ->\n        @Suppress("UNCHECKED_CAST") (ReverseOrderComparator as Comparator<T>)\n        ReverseOrderComparator -> @Suppress("UNCHECKED_CAST") (NaturalOrderComparator as\n            Comparator<T>)\n        else -> ReversedComparator(this)\n    }\n\nprivate class ReversedComparator<T> (public val\n    comparator: Comparator<T>) : Comparator<T> {\n    override fun compare(a: T, b: T): Int = comparator.compare(b,\n        a)\n    @Suppress("VIRTUAL_MEMBER_HIDDEN")\n    fun reversed(): Comparator<T> =\n        comparator\n    private object NaturalOrderComparator : Comparator<Comparable<Any>> {\n        override fun\n            compare(a: Comparable<Any>, b: Comparable<Any>): Int = a.compareTo(b)\n    }\n    @Suppress("VIRTUAL_MEMBER_HIDDEN")\n    fun reversed(): Comparator<Comparable<Any>> =\n        ReverseOrderComparator\n    private object ReverseOrderComparator : Comparator<Comparable<Any>> {\n        override fun\n            compare(a: Comparable<Any>, b: Comparable<Any>): Int = b.compareTo(a)\n    }\n    @Suppress("VIRTUAL_MEMBER_HIDDEN")\n    fun reversed(): Comparator<Comparable<Any>> =\n        NaturalOrderComparator\n    }\n\n"/>\n\n/**\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language\n * contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the\n * license/LICENSE.txt file.\n\n*\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("StandardKt")\npackage kotlin\nimport\n    kotlin.contracts.*\n\n/**\n * An exception is thrown to indicate that a method body remains to be implemented.\n */\n\npublic class NotImplementedError(message: String = "An operation is not implemented.") :\n    Error(message)\n\n/**\n * Always throws [NotImplementedError] stating that operation is not implemented.\n */\n\n@kotlin.internal.InlineOnly\npublic inline fun TODO(): Nothing = throw NotImplementedError()\n\n/**\n * Always throws [NotImplementedError] stating that operation is not implemented.\n */\n * @param reason a string\n * explaining why the implementation is missing.\n */\n@kotlin.internal.InlineOnly\npublic inline fun TODO(reason:\n    String): Nothing = throw NotImplementedError("An operation is not implemented: $reason")\n\n/**\n * Calls\n * the specified function [block] and returns its result.\n */\n * For detailed usage information see the documentation for\n * [scope functions](https://kotlinlang.org/docs/reference/scope-functions.html#run).\n\n*\n@kotlin.internal.InlineOnly\npublic inline fun <R> run(block: () -> R): R {\n    contract {

```

```

callsInPlace(block, InvocationKind.EXACTLY_ONCE)\n } \n return block()\n}\n\n/**\n * Calls the specified
function [block] with `this` value as its receiver and returns its result.\n *\n * For detailed usage information see the
documentation for [scope functions](https://kotlinlang.org/docs/reference/scope-functions.html#run).\n
*\n @kotlin.internal.InlineOnly\n public inline fun <T, R> T.run(block: T.() -> R): R {\n contract {\n
callsInPlace(block, InvocationKind.EXACTLY_ONCE)\n } \n return block()\n}\n\n/**\n * Calls the specified
function [block] with the given [receiver] as its receiver and returns its result.\n *\n * For detailed usage information
see the documentation for [scope functions](https://kotlinlang.org/docs/reference/scope-functions.html#with).\n
*\n @kotlin.internal.InlineOnly\n public inline fun <T, R> with(receiver: T, block: T.() -> R): R {\n contract {\n
callsInPlace(block, InvocationKind.EXACTLY_ONCE)\n } \n return receiver.block()\n}\n\n/**\n * Calls the
specified function [block] with `this` value as its receiver and returns `this` value.\n *\n * For detailed usage
information see the documentation for [scope functions](https://kotlinlang.org/docs/reference/scope-
functions.html#apply).\n *\n @kotlin.internal.InlineOnly\n public inline fun <T> T.apply(block: T.() -> Unit): T {\n
contract {\n callsInPlace(block, InvocationKind.EXACTLY_ONCE)\n } \n block()\n return
this}\n}\n\n/**\n * Calls the specified function [block] with `this` value as its argument and returns `this` value.\n
*\n * For detailed usage information see the documentation for [scope
functions](https://kotlinlang.org/docs/reference/scope-functions.html#also).\n
*\n @kotlin.internal.InlineOnly\n @SinceKotlin("1.1")\n public inline fun <T> T.also(block: (T) -> Unit): T {\n
contract {\n callsInPlace(block, InvocationKind.EXACTLY_ONCE)\n } \n block(this)\n return
this}\n}\n\n/**\n * Calls the specified function [block] with `this` value as its argument and returns its result.\n
*\n * For detailed usage information see the documentation for [scope
functions](https://kotlinlang.org/docs/reference/scope-functions.html#let).\n
*\n @kotlin.internal.InlineOnly\n public inline fun <T, R> T.let(block: (T) -> R): R {\n contract {\n
callsInPlace(block,
InvocationKind.EXACTLY_ONCE)\n } \n return block(this)\n}\n\n/**\n * Returns `this` value if it satisfies the
given [predicate] or `null`, if it doesn't.\n *\n * For detailed usage information see the documentation for [scope
functions](https://kotlinlang.org/docs/reference/scope-functions.html#takeif-and-takeunless).\n
*\n @kotlin.internal.InlineOnly\n @SinceKotlin("1.1")\n public inline fun <T> T.takeIf(predicate: (T) -> Boolean):
T? {\n contract {\n callsInPlace(predicate, InvocationKind.EXACTLY_ONCE)\n } \n return if
(predicate(this)) this else null\n}\n\n/**\n * Returns `this` value if it _does not_ satisfy the given [predicate] or
`null`, if it does.\n *\n * For detailed usage information see the documentation for [scope
functions](https://kotlinlang.org/docs/reference/scope-functions.html#takeif-and-takeunless).\n
*\n @kotlin.internal.InlineOnly\n @SinceKotlin("1.1")\n public inline fun <T> T.takeUnless(predicate: (T) ->
Boolean): T? {\n contract {\n callsInPlace(predicate, InvocationKind.EXACTLY_ONCE)\n } \n return if
(!predicate(this)) this else null\n}\n\n/**\n * Executes the given function [action] specified number of [times].\n
*\n * A zero-based index of current iteration is passed as a parameter to [action].\n *\n * @sample
samples.misc.ControlFlow.repeat\n *\n @kotlin.internal.InlineOnly\n public inline fun repeat(times: Int, action: (Int)
-> Unit) {\n contract { callsInPlace(action) }\n for (index in 0 until times) {\n action(index)\n
}\n}\n\n", /*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of
this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*\n @npackage kotlin.comparisons\n\n/\n NOTE: THIS FILE IS AUTO-GENERATED by the
GenerateStandardLib.kt\n See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\nimport
kotlin.js.\n\n/**\n * Returns the greater of two values.\n *\n * If values are equal, returns the first one.\n
*\n @SinceKotlin("1.1")\n public actual fun <T : Comparable<T>> maxOf(a: T, b: T): T {\n return if (a >= b) a
else b}\n}\n\n/**\n * Returns the greater of two values.\n
*\n @SinceKotlin("1.1")\n @kotlin.internal.InlineOnly\n public actual inline fun maxOf(a: Byte, b: Byte): Byte {\n
return maxOf(a.toInt(), b.toInt()).unsafeCast<Byte>()\n}\n\n/**\n * Returns the greater of two values.\n
*\n @SinceKotlin("1.1")\n @kotlin.internal.InlineOnly\n public actual inline fun maxOf(a: Short, b: Short): Short
{\n return maxOf(a.toInt(), b.toInt()).unsafeCast<Short>()\n}\n\n/**\n * Returns the greater of two values.\n
*\n @SinceKotlin("1.1")\n @kotlin.internal.InlineOnly\n public actual inline fun maxOf(a: Int, b: Int): Int {\n

```

```

return JsMath.max(a, b)\n}\n\n/**\n * Returns the greater of two values.\n
*\n@SinceKotlin("1.1")\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline fun maxOf(a: Long, b:
Long): Long {\n    return if (a >= b) a else b\n}\n\n/**\n * Returns the greater of two values.\n * \n * If either value
is `NaN`, returns `NaN`.\n *\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic actual inline fun
maxOf(a: Float, b: Float): Float {\n    return JsMath.max(a, b)\n}\n\n/**\n * Returns the greater of two values.\n * \n
* If either value is `NaN`, returns `NaN`.\n *\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic actual
inline fun maxOf(a: Double, b: Double): Double {\n    return JsMath.max(a, b)\n}\n\n/**\n * Returns the greater of
three values.\n * \n * If there are multiple equal maximal values, returns the first of them.\n
*\n@SinceKotlin("1.1")\npublic actual fun <T : Comparable<T>> maxOf(a: T, b: T, c: T): T {\n    return
maxOf(a, maxOf(b, c))\n}\n\n/**\n * Returns the greater of three values.\n
*\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic actual inline fun maxOf(a: Byte, b: Byte, c: Byte):
Byte {\n    return JsMath.max(a.toInt(), b.toInt(), c.toInt()).unsafeCast<Byte>()\n}\n\n/**\n * Returns the greater of
three values.\n *\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic actual inline fun maxOf(a: Short, b:
Short, c: Short): Short {\n    return JsMath.max(a.toInt(), b.toInt(), c.toInt()).unsafeCast<Short>()\n}\n\n/**\n *
Returns the greater of three values.\n *\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic actual inline
fun maxOf(a: Int, b: Int, c: Int): Int {\n    return JsMath.max(a, b, c)\n}\n\n/**\n * Returns the greater of three
values.\n *\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic actual inline fun maxOf(a: Long, b: Long,
c: Long): Long {\n    return maxOf(a, maxOf(b, c))\n}\n\n/**\n * Returns the greater of three values.\n * \n * If any
value is `NaN`, returns `NaN`.\n *\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic actual inline fun
maxOf(a: Float, b: Float, c: Float): Float {\n    return JsMath.max(a, b, c)\n}\n\n/**\n * Returns the greater of three
values.\n * \n * If any value is `NaN`, returns `NaN`.\n
*\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic actual inline fun maxOf(a: Double, b: Double, c:
Double): Double {\n    return JsMath.max(a, b, c)\n}\n\n/**\n * Returns the greater of the given values.\n * \n * If
there are multiple equal maximal values, returns the first of them.\n *\n@SinceKotlin("1.4")\npublic actual fun <T
: Comparable<T>> maxOf(a: T, vararg other: T): T {\n    var max = a\n    for (e in other) max = maxOf(max, e)\n    return
max\n}\n\n/**\n * Returns the greater of the given values.\n *\n@SinceKotlin("1.4")\npublic actual fun
maxOf(a: Byte, vararg other: Byte): Byte {\n    var max = a\n    for (e in other) max = maxOf(max, e)\n    return
max\n}\n\n/**\n * Returns the greater of the given values.\n *\n@SinceKotlin("1.4")\npublic actual fun maxOf(a:
Short, vararg other: Short): Short {\n    var max = a\n    for (e in other) max = maxOf(max, e)\n    return
max\n}\n\n/**\n * Returns the greater of the given values.\n *\n@SinceKotlin("1.4")\npublic actual fun maxOf(a:
Int, vararg other: Int): Int {\n    var max = a\n    for (e in other) max = maxOf(max, e)\n    return max\n}\n\n/**\n
* Returns the greater of the given values.\n *\n@SinceKotlin("1.4")\npublic actual fun maxOf(a: Long, vararg
other: Long): Long {\n    var max = a\n    for (e in other) max = maxOf(max, e)\n    return max\n}\n\n/**\n *
Returns the greater of the given values.\n * \n * If any value is `NaN`, returns `NaN`.\n
*\n@SinceKotlin("1.4")\npublic actual fun maxOf(a: Float, vararg other: Float): Float {\n    var max = a\n    for (e
in other) max = maxOf(max, e)\n    return max\n}\n\n/**\n * Returns the greater of the given values.\n * \n * If any
value is `NaN`, returns `NaN`.\n *\n@SinceKotlin("1.4")\npublic actual fun maxOf(a: Double, vararg other:
Double): Double {\n    var max = a\n    for (e in other) max = maxOf(max, e)\n    return max\n}\n\n/**\n * Returns
the smaller of two values.\n * \n * If values are equal, returns the first one.\n *\n@SinceKotlin("1.1")\npublic
actual fun <T : Comparable<T>> minOf(a: T, b: T): T {\n    return if (a <= b) a else b\n}\n\n/**\n * Returns the
smaller of two values.\n *\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic actual inline fun minOf(a:
Byte, b: Byte): Byte {\n    return minOf(a.toInt(), b.toInt()).unsafeCast<Byte>()\n}\n\n/**\n * Returns the smaller of
two values.\n *\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic actual inline fun minOf(a: Short, b:
Short): Short {\n    return minOf(a.toInt(), b.toInt()).unsafeCast<Short>()\n}\n\n/**\n * Returns the smaller of two
values.\n *\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic actual inline fun minOf(a: Int, b: Int): Int
{\n    return JsMath.min(a, b)\n}\n\n/**\n * Returns the smaller of two values.\n
*\n@SinceKotlin("1.1")\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline fun minOf(a: Long, b:
Long): Long {\n    return if (a <= b) a else b\n}\n\n/**\n * Returns the smaller of two values.\n * \n * If either value

```

```

is `NaN`, returns `NaN`.n *^@SinceKotlin("1.1")n@kotlin.internal.InlineOnlynpublic actual inline fun
minOf(a: Float, b: Float): Float {n    return JsMath.min(a, b)n}n/n/**n * Returns the smaller of two values.n *
* If either value is `NaN`, returns `NaN`.n *^@SinceKotlin("1.1")n@kotlin.internal.InlineOnlynpublic actual
inline fun minOf(a: Double, b: Double): Double {n    return JsMath.min(a, b)n}n/n/**n * Returns the smaller of
three values.n * n * If there are multiple equal minimal values, returns the first of them.n
*^@SinceKotlin("1.1")npublic actual fun <T : Comparable<T>> minOf(a: T, b: T, c: T): T {n    return minOf(a,
minOf(b, c))n}n/n/**n * Returns the smaller of three values.n
*^@SinceKotlin("1.1")n@kotlin.internal.InlineOnlynpublic actual inline fun minOf(a: Byte, b: Byte, c: Byte):
Byte {n    return JsMath.min(a.toInt(), b.toInt(), c.toInt()).unsafeCast<Byte>()n}n/n/**n * Returns the smaller of
three values.n *^@SinceKotlin("1.1")n@kotlin.internal.InlineOnlynpublic actual inline fun minOf(a: Short, b:
Short, c: Short): Short {n    return JsMath.min(a.toInt(), b.toInt(), c.toInt()).unsafeCast<Short>()n}n/n/**n *
Returns the smaller of three values.n *^@SinceKotlin("1.1")n@kotlin.internal.InlineOnlynpublic actual inline
fun minOf(a: Int, b: Int, c: Int): Int {n    return JsMath.min(a, b, c)n}n/n/**n * Returns the smaller of three
values.n *^@SinceKotlin("1.1")n@kotlin.internal.InlineOnlynpublic actual inline fun minOf(a: Long, b: Long,
c: Long): Long {n    return minOf(a, minOf(b, c))n}n/n/**n * Returns the smaller of three values.n * n * If any
value is `NaN`, returns `NaN`.n *^@SinceKotlin("1.1")n@kotlin.internal.InlineOnlynpublic actual inline fun
minOf(a: Float, b: Float, c: Float): Float {n    return JsMath.min(a, b, c)n}n/n/**n * Returns the smaller of three
values.n * n * If any value is `NaN`, returns `NaN`.n
*^@SinceKotlin("1.1")n@kotlin.internal.InlineOnlynpublic actual inline fun minOf(a: Double, b: Double, c:
Double): Double {n    return JsMath.min(a, b, c)n}n/n/**n * Returns the smaller of the given values.n * n * If
there are multiple equal minimal values, returns the first of them.n *^@SinceKotlin("1.4")npublic actual fun <T
: Comparable<T>> minOf(a: T, vararg other: T): T {n    var min = a\n    for (e in other) min = minOf(min, e)n
return min\n}n/n/**n * Returns the smaller of the given values.n *^@SinceKotlin("1.4")npublic actual fun
minOf(a: Byte, vararg other: Byte): Byte {n    var min = a\n    for (e in other) min = minOf(min, e)n    return
min\n}n/n/**n * Returns the smaller of the given values.n *^@SinceKotlin("1.4")npublic actual fun minOf(a:
Short, vararg other: Short): Short {n    var min = a\n    for (e in other) min = minOf(min, e)n    return
min\n}n/n/**n * Returns the smaller of the given values.n *^@SinceKotlin("1.4")npublic actual fun minOf(a:
Int, vararg other: Int): Int {n    var min = a\n    for (e in other) min = minOf(min, e)n    return min\n}n/n/**n *
Returns the smaller of the given values.n *^@SinceKotlin("1.4")npublic actual fun minOf(a: Long, vararg
other: Long): Long {n    var min = a\n    for (e in other) min = minOf(min, e)n    return min\n}n/n/**n * Returns
the smaller of the given values.n * n * If any value is `NaN`, returns `NaN`.n *^@SinceKotlin("1.4")npublic
actual fun minOf(a: Float, vararg other: Float): Float {n    var min = a\n    for (e in other) min = minOf(min, e)n
return min\n}n/n/**n * Returns the smaller of the given values.n * n * If any value is `NaN`, returns `NaN`.n
*^@SinceKotlin("1.4")npublic actual fun minOf(a: Double, vararg other: Double): Double {n    var min = a\n
for (e in other) min = minOf(min, e)n    return min\n}n/n"/**n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin
Programming Language contributors.n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.n *^@n// Auto-generated file. DO NOT EDIT!npackage kotlin\nimport
kotlin.experimental.*\nimport
kotlin.jvm.*n@SinceKotlin("1.5")n@WasExperimental(ExperimentalUnsignedTypes::class)n@JvmInlinenpu
blic value class ULong @PublishedApi internal constructor(@PublishedApi internal val data: Long) :
Comparable<ULong> {n    companion object {n        /**n        * A constant holding the minimum value an
instance of ULong can have.n        *^@    public const val MIN_VALUE: ULong = ULong(0)n        /**n
* A constant holding the maximum value an instance of ULong can have.n        *^@    public const val
MAX_VALUE: ULong = ULong(-1)n        /**n        * The number of bytes used to represent an instance of
ULong in a binary form.n        *^@    public const val SIZE_BYTES: Int = 8\n        /**n        * The number of
bits used to represent an instance of ULong in a binary form.n        *^@    public const val SIZE_BITS: Int = 64\n
    }n    /**n    * Compares this value with the specified value for order.n    * Returns zero if this value is equal to
the specified other value, a negative number if it's less than other.n    * or a positive number if it's greater than

```

```

other.\n    *\n    @kotlin.internal.InlineOnly\n    public inline operator fun compareTo(other: UByte): Int =
this.compareTo(other.toULong())\n\n    /**\n     * Compares this value with the specified value for order.\n     * Returns zero if this value is equal to the specified other value, a negative number if it's less than other,\n     * or a positive number if it's greater than other.\n     *\n    @kotlin.internal.InlineOnly\n    public inline operator fun
compareTo(other: UShort): Int = this.compareTo(other.toULong())\n\n    /**\n     * Compares this value with the
specified value for order.\n     * Returns zero if this value is equal to the specified other value, a negative number if
it's less than other,\n     * or a positive number if it's greater than other.\n     *\n    @kotlin.internal.InlineOnly\n
public inline operator fun compareTo(other: UInt): Int = this.compareTo(other.toULong())\n\n    /**\n     *
Compares this value with the specified value for order.\n     * Returns zero if this value is equal to the specified other
value, a negative number if it's less than other,\n     * or a positive number if it's greater than other.\n     *\n
    @kotlin.internal.InlineOnly\n    @Suppress("OVERRIDE_BY_INLINE")\n    public override inline operator fun
compareTo(other: ULong): Int = ulongCompare(this.data, other.data)\n\n    /** Adds the other value to this value.
*\n    @kotlin.internal.InlineOnly\n    public inline operator fun plus(other: UByte): ULong =
this.plus(other.toULong())\n\n    /** Adds the other value to this value. *\n    @kotlin.internal.InlineOnly\n
public inline operator fun plus(other: UShort): ULong = this.plus(other.toULong())\n\n    /** Adds the other value to this
value. *\n    @kotlin.internal.InlineOnly\n    public inline operator fun plus(other: UInt): ULong =
this.plus(other.toULong())\n\n    /** Adds the other value to this value. *\n    @kotlin.internal.InlineOnly\n
public inline operator fun plus(other: ULong): ULong = ULong(this.data.plus(other.data))\n\n    /** Subtracts the other
value from this value. *\n    @kotlin.internal.InlineOnly\n    public inline operator fun minus(other: UByte): ULong
= this.minus(other.toULong())\n\n    /** Subtracts the other value from this value. *\n    @kotlin.internal.InlineOnly\n
public inline operator fun minus(other: UShort): ULong = this.minus(other.toULong())\n\n    /** Subtracts the other
value from this value. *\n    @kotlin.internal.InlineOnly\n    public inline operator fun minus(other: UInt): ULong =
this.minus(other.toULong())\n\n    /** Subtracts the other value from this value. *\n    @kotlin.internal.InlineOnly\n
public inline operator fun minus(other: ULong): ULong = ULong(this.data.minus(other.data))\n\n    /** Multiplies
this value by the other value. *\n    @kotlin.internal.InlineOnly\n    public inline operator fun times(other: UByte):
ULong = this.times(other.toULong())\n\n    /** Multiplies this value by the other value. *\n
    @kotlin.internal.InlineOnly\n    public inline operator fun times(other: UShort): ULong =
this.times(other.toULong())\n\n    /** Multiplies this value by the other value. *\n    @kotlin.internal.InlineOnly\n
public inline operator fun times(other: UInt): ULong = this.times(other.toULong())\n\n    /** Multiplies this value by
the other value. *\n    @kotlin.internal.InlineOnly\n    public inline operator fun times(other: ULong): ULong =
ULong(this.data.times(other.data))\n\n    /** Divides this value by the other value, truncating the result to an integer
that is closer to zero. *\n    @kotlin.internal.InlineOnly\n    public inline operator fun div(other: UByte): ULong =
this.div(other.toULong())\n\n    /** Divides this value by the other value, truncating the result to an integer that is
closer to zero. *\n    @kotlin.internal.InlineOnly\n    public inline operator fun div(other: UShort): ULong =
this.div(other.toULong())\n\n    /** Divides this value by the other value, truncating the result to an integer that is
closer to zero. *\n    @kotlin.internal.InlineOnly\n    public inline operator fun div(other: UInt): ULong =
this.div(other.toULong())\n\n    /** Divides this value by the other value, truncating the result to an integer that is
closer to zero. *\n    @kotlin.internal.InlineOnly\n    public inline operator fun div(other: ULong): ULong =
ulongDivide(this, other)\n\n    /**\n     * Calculates the remainder of truncating division of this value by the other
value.\n     * \n     * The result is always less than the divisor.\n     *\n    @kotlin.internal.InlineOnly\n    public
inline operator fun rem(other: UByte): ULong = this.rem(other.toULong())\n\n    /**\n     * Calculates the remainder
of truncating division of this value by the other value.\n     * \n     * The result is always less than the divisor.\n
*\n    @kotlin.internal.InlineOnly\n    public inline operator fun rem(other: UShort): ULong =
this.rem(other.toULong())\n\n    /**\n     * Calculates the remainder of truncating division of this value by the other
value.\n     * \n     * The result is always less than the divisor.\n     *\n    @kotlin.internal.InlineOnly\n    public
inline operator fun rem(other: UInt): ULong = this.rem(other.toULong())\n\n    /**\n     * Calculates the remainder of
truncating division of this value by the other value.\n     * \n     * The result is always less than the divisor.\n    
*\n    @kotlin.internal.InlineOnly\n    public inline operator fun rem(other: ULong): ULong = ulongRemainder(this,

```

```

other)\n\n /**\n * Divides this value by the other value, flooring the result to an integer that is closer to negative
infinity.\n * \n * For unsigned types, the results of flooring division and truncating division are the same.\n
*\n @kotlin.internal.InlineOnly\n public inline fun floorDiv(other: UByte): ULong =
this.floorDiv(other.toULong())\n /**\n * Divides this value by the other value, flooring the result to an integer
that is closer to negative infinity.\n * \n * For unsigned types, the results of flooring division and truncating
division are the same.\n *\n @kotlin.internal.InlineOnly\n public inline fun floorDiv(other: UShort): ULong
= this.floorDiv(other.toULong())\n /**\n * Divides this value by the other value, flooring the result to an integer
that is closer to negative infinity.\n * \n * For unsigned types, the results of flooring division and truncating
division are the same.\n *\n @kotlin.internal.InlineOnly\n public inline fun floorDiv(other: UInt): ULong =
this.floorDiv(other.toULong())\n /**\n * Divides this value by the other value, flooring the result to an integer
that is closer to negative infinity.\n * \n * For unsigned types, the results of flooring division and truncating
division are the same.\n *\n @kotlin.internal.InlineOnly\n public inline fun floorDiv(other: ULong): ULong =
div(other)\n\n /**\n * Calculates the remainder of flooring division of this value by the other value.\n * \n *
The result is always less than the divisor.\n * \n * For unsigned types, the remainders of flooring division and
truncating division are the same.\n *\n @kotlin.internal.InlineOnly\n public inline fun mod(other: UByte):
UByte = this.mod(other.toULong()).toUByte()\n /**\n * Calculates the remainder of flooring division of this
value by the other value.\n * \n * The result is always less than the divisor.\n * \n * For unsigned types, the
remainders of flooring division and truncating division are the same.\n *\n @kotlin.internal.InlineOnly\n
public inline fun mod(other: UShort): UShort = this.mod(other.toULong()).toUShort()\n /**\n * Calculates the
remainder of flooring division of this value by the other value.\n * \n * The result is always less than the
divisor.\n * \n * For unsigned types, the remainders of flooring division and truncating division are the same.\n
*\n @kotlin.internal.InlineOnly\n public inline fun mod(other: UInt): UInt =
this.mod(other.toULong()).toUInt()\n /**\n * Calculates the remainder of flooring division of this value by the
other value.\n * \n * The result is always less than the divisor.\n * \n * For unsigned types, the remainders
of flooring division and truncating division are the same.\n *\n @kotlin.internal.InlineOnly\n public inline
fun mod(other: ULong): ULong = rem(other)\n\n /**\n * Returns this value incremented by one.\n * \n *
@sample samples.misc.Builtins.inc\n *\n @kotlin.internal.InlineOnly\n public inline operator fun inc():
ULong = ULong(data.inc())\n\n /**\n * Returns this value decremented by one.\n * \n * @sample
samples.misc.Builtins.dec\n *\n @kotlin.internal.InlineOnly\n public inline operator fun dec(): ULong =
ULong(data.dec())\n\n /**\n * Creates a range from this value to the specified [other] value. *\n
@\n @kotlin.internal.InlineOnly\n public inline operator fun rangeTo(other: ULong): ULongRange =
ULongRange(this, other)\n\n /**\n * Shifts this value left by the [bitCount] number of bits.\n * \n * Note
that only the six lowest-order bits of the [bitCount] are used as the shift distance.\n * The shift distance actually
used is therefore always in the range `0..63`.\n *\n @kotlin.internal.InlineOnly\n public inline infix fun
shl(bitCount: Int): ULong = ULong(data shl bitCount)\n\n /**\n * Shifts this value right by the [bitCount]
number of bits, filling the leftmost bits with zeros.\n * \n * Note that only the six lowest-order bits of the
[bitCount] are used as the shift distance.\n * The shift distance actually used is therefore always in the range
`0..63`.\n *\n @kotlin.internal.InlineOnly\n public inline infix fun shr(bitCount: Int): ULong = ULong(data
ushr bitCount)\n\n /**\n * Performs a bitwise AND operation between the two values. *\n
@\n @kotlin.internal.InlineOnly\n public inline infix fun and(other: ULong): ULong = ULong(this.data and
other.data)\n\n /**\n * Performs a bitwise OR operation between the two values. *\n
@\n @kotlin.internal.InlineOnly\n public inline infix fun or(other: ULong): ULong = ULong(this.data or other.data)\n
/**\n * Performs a bitwise XOR
operation between the two values. *\n @kotlin.internal.InlineOnly\n public inline infix fun xor(other: ULong):
ULong = ULong(this.data xor other.data)\n\n /**\n * Inverts the bits in this value. *\n
@\n @kotlin.internal.InlineOnly\n public inline fun inv(): ULong = ULong(data.inv())\n\n /**\n * Converts this [ULong] value to [Byte].\n * \n
* If this value is less than or equals to [Byte.MAX_VALUE], the resulting `Byte` value represents\n * the same
numerical value as this `ULong`.\n * \n * The resulting `Byte` value is represented by the least significant 8 bits
of this `ULong` value.\n * Note that the resulting `Byte` value may be negative.\n *\n

```



```

* The resulting `ULong` value has the same binary representation as this `Long` value.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun Long.toULong(): ULong = ULong(this)\n\n/**\n * Converts this [Float] value to [ULong].\n *\n * The fractional part, if any, is rounded down towards zero.\n * Returns zero if this `Float` value is negative or `NaN`,\n * [ULong.MAX_VALUE] if it's bigger than `ULong.MAX_VALUE`.\n\n*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun Float.toULong(): ULong = doubleToULong(this.toDouble())\n\n/**\n * Converts this [Double]\n * value to [ULong].\n *\n * The fractional part, if any, is rounded down towards zero.\n * Returns zero if this\n * `Double` value is negative or `NaN`, [ULong.MAX_VALUE] if it's bigger than `ULong.MAX_VALUE`.\n\n*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun Double.toULong(): ULong = doubleToULong(this)\n", "/*\n * Copyright 2010-2021 JetBrains\n * s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0\n * license that can be found in the license/LICENSE.txt file.\n\n*\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("CollectionsKt")\n\npackage\nkotlin.collections\n\n\n/\n NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n See:\n https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\n\nimport kotlin.random.*\nimport\nkotlin.ranges.contains\nimport kotlin.ranges.reversed\n\n\n/**\n * Returns 1st *element* from the list.\n *\n * Throws an [IndexOutOfBoundsException] if the size of this list is less than 1.\n\n*\n@kotlin.internal.InlineOnly\npublic inline operator fun <T> List<T>.component1(): T {\n    return\n    get(0)\n}\n\n/**\n * Returns 2nd *element* from the list.\n *\n * Throws an [IndexOutOfBoundsException] if the\n * size of this list is less than 2.\n\n*\n@kotlin.internal.InlineOnly\npublic inline operator fun <T>\nList<T>.component2(): T {\n    return get(1)\n}\n\n/**\n * Returns 3rd *element* from the list.\n *\n * Throws an\n * [IndexOutOfBoundsException] if the size of this list is less than 3.\n\n*\n@kotlin.internal.InlineOnly\npublic inline\noperator fun <T> List<T>.component3(): T {\n    return get(2)\n}\n\n/**\n * Returns 4th *element* from the list.\n *\n * Throws an [IndexOutOfBoundsException] if the size of this list is less than 4.\n\n*\n@kotlin.internal.InlineOnly\npublic inline operator fun <T> List<T>.component4(): T {\n    return\n    get(3)\n}\n\n/**\n * Returns 5th *element* from the list.\n *\n * Throws an [IndexOutOfBoundsException] if the\n * size of this list is less than 5.\n\n*\n@kotlin.internal.InlineOnly\npublic inline operator fun <T>\nList<T>.component5(): T {\n    return get(4)\n}\n\n/**\n * Returns `true` if [element] is found in the collection.\n\n*\npublic operator fun <@kotlin.internal.OnlyInputTypes T> Iterable<T>.contains(element: T): Boolean {\n    if\n    (this is Collection)\n        return contains(element)\n        return indexOf(element) >= 0\n}\n\n/**\n * Returns an\n * element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this\n * collection.\n *\n * @sample samples.collections.Collections.Elements.elementAt\n\n*\npublic fun <T>\nIterable<T>.elementAt(index: Int): T {\n    if (this is List)\n        return get(index)\n        return\n        elementAtOrElse(index) {\n            throw IndexOutOfBoundsException("Collection doesn't contain element at index\n            $index.")\n        }\n}\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if\n * the [index] is out of bounds of this list.\n *\n * @sample samples.collections.Collections.Elements.elementAt\n\n*\n@kotlin.internal.InlineOnly\npublic inline fun <T> List<T>.elementAt(index: Int): T {\n    return\n    get(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue] function\n * if the [index] is out of bounds of this collection.\n *\n * @sample\n * samples.collections.Collections.Elements.elementAtOrElse\n\n*\npublic fun <T>\nIterable<T>.elementAtOrElse(index: Int, defaultValue: (Int) -> T): T {\n    if (this is List)\n        return\n        this.getOrElse(index, defaultValue)\n        if (index < 0)\n            return defaultValue(index)\n            val iterator = iterator()\n            var count = 0\n            while (iterator.hasNext()) {\n                val element = iterator.next()\n                if (index == count++)\n                    return element\n            }\n            return defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the\n * result of calling the [defaultValue] function if the [index] is out of bounds of this list.\n *\n * @sample\n * samples.collections.Collections.Elements.elementAtOrElse\n\n*\n@kotlin.internal.InlineOnly\npublic inline fun\n<T> List<T>.elementAtOrElse(index: Int, defaultValue: (Int) -> T): T {\n    return if (index >= 0 && index <=

```

```

lastIndex) get(index) else defaultValue(index)\n\n\n\n * Returns an element at the given [index] or `null` if the
[index] is out of bounds of this collection.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrNull\n * \n\npublic fun <T>
Iterable<T>.elementAtOrNull(index: Int): T? {\n    if (this is List)\n        return this.getOrNull(index)\n    if (index <
0)\n        return null\n    val iterator = iterator()\n    var count = 0\n    while (iterator.hasNext()) {\n        val element =
iterator.next()\n        if (index == count++)\n            return element\n    }\n    return null\n}\n\n\n\n * Returns an
element at the given [index] or `null` if the [index] is out of bounds of this list.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrNull\n * \n\n@kotlin.internal.InlineOnly\n\npublic inline fun
<T> List<T>.elementAtOrNull(index: Int): T? {\n    return this.getOrNull(index)\n}\n\n\n\n * Returns the first
element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n * \n\n@kotlin.internal.InlineOnly\n\npublic inline fun <T>
Iterable<T>.find(predicate: (T) -> Boolean): T? {\n    return firstOrNull(predicate)\n}\n\n\n\n * Returns the last
element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n * \n\n@kotlin.internal.InlineOnly\n\npublic inline fun <T>
Iterable<T>.findLast(predicate: (T) -> Boolean): T? {\n    return lastOrNull(predicate)\n}\n\n\n\n * Returns the last
element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n * \n\n@kotlin.internal.InlineOnly\n\npublic inline fun <T>
List<T>.findLast(predicate: (T) -> Boolean): T? {\n    return lastOrNull(predicate)\n}\n\n\n\n * Returns first
element.\n * @throws [NoSuchElementException] if the collection is empty.\n * \n\npublic fun <T>
Iterable<T>.first(): T {\n    when (this) {\n        is List -> return this.first()\n        else -> {\n            val iterator =
iterator()\n            if (!iterator.hasNext())\n                throw NoSuchElementException("Collection is empty.")\n            return iterator.next()\n        }\n    }\n}\n\n\n\n * Returns first element.\n * @throws [NoSuchElementException]
if the list is empty.\n * \n\npublic fun <T> List<T>.first(): T {\n    if (isEmpty())\n        throw
NoSuchElementException("List is empty.")\n    return this[0]\n}\n\n\n\n * Returns the first element matching the
given [predicate].\n * @throws [NoSuchElementException] if no such element is found.\n * \n\npublic inline fun <T>
Iterable<T>.first(predicate: (T) -> Boolean): T {\n    for (element in this) if (predicate(element)) return element\n
throw NoSuchElementException("Collection contains no element matching the predicate.")\n}\n\n\n\n * Returns
the first non-null value produced by [transform] function being applied to elements of this collection in iteration
order,\n * or throws [NoSuchElementException] if no non-null value was produced.\n * \n * @sample
samples.collections.Collections.Transformations.firstNotNullOf\n
*\n * \n\n@SinceKotlin("1.5")\n\n@kotlin.internal.InlineOnly\n\npublic inline fun <T, R : Any>
Iterable<T>.firstNotNullOf(transform: (T) -> R?): R {\n    return firstNotNullOfOrNull(transform) ?: throw
NoSuchElementException("No element of the collection was transformed to a non-null value.")\n}\n\n\n\n *
Returns the first non-null value produced by [transform] function being applied to elements of this collection in
iteration order,\n * or `null` if no non-null value was produced.\n * \n * @sample
samples.collections.Collections.Transformations.firstNotNullOf\n
*\n * \n\n@SinceKotlin("1.5")\n\n@kotlin.internal.InlineOnly\n\npublic inline fun <T, R : Any>
Iterable<T>.firstNotNullOfOrNull(transform: (T) -> R?): R? {\n    for (element in this) {\n        val result =
transform(element)\n        if (result != null) {\n            return result\n        }\n    }\n    return null\n}\n\n\n\n *
Returns the first element, or `null` if the collection is empty.\n * \n\npublic fun <T> Iterable<T>.firstOrNull(): T? {\n
when (this) {\n    is List -> {\n        if (isEmpty())\n            return null\n        else\n            return this[0]\n    }\n    else -> {\n        val iterator = iterator()\n        if (!iterator.hasNext())\n            return null\n        return iterator.next()\n    }\n}\n}\n\n\n\n * Returns the first element, or `null` if the list is empty.\n * \n\npublic
fun <T> List<T>.firstOrNull(): T? {\n    return if (isEmpty()) null else this[0]\n}\n\n\n\n * Returns the first element
matching the given [predicate], or `null` if element was not found.\n * \n\npublic inline fun <T>
Iterable<T>.firstOrNull(predicate: (T) -> Boolean): T? {\n    for (element in this) if (predicate(element)) return
element\n    return null\n}\n\n\n\n * Returns an element at the given [index] or the result of calling the
[defaultValue] function if the [index] is out of bounds of this list.\n * \n\n@kotlin.internal.InlineOnly\n\npublic inline

```



```

some cases\npublic fun <@kotlin.internal.OnlyInputTypes T> List<T>.lastIndexOf(element: T): Int {\n    return
lastIndexOf(element)\n}\n\n/**\n * Returns the last element, or `null` if the collection is empty.\n * \n * @sample
samples.collections.Collections.Elements.last\n */\npublic fun <T> Iterable<T>.lastOrNull(): T? {\n    when (this)
{\n        is List -> return if (isEmpty()) null else this[size - 1]\n        else -> {\n            val iterator = iterator()\n            if (!iterator.hasNext())\n                return null\n            var last = iterator.next()\n            while (iterator.hasNext())\n                last = iterator.next()\n            return last\n        }\n    }\n}\n\n/**\n * Returns the last element, or `null` if the
list is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic fun <T>
List<T>.lastOrNull(): T? {\n    return if (isEmpty()) null else this[size - 1]\n}\n\n/**\n * Returns the last element
matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.last\n */\npublic inline fun <T> Iterable<T>.lastOrNull(predicate: (T) ->
Boolean): T? {\n    var last: T? = null\n    for (element in this) {\n        if (predicate(element)) {\n            last =
element\n        }\n    }\n    return last\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null`
if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic inline
fun <T> List<T>.lastOrNull(predicate: (T) -> Boolean): T? {\n    val iterator = this.listIterator(size)\n    while
(iterator.hasPrevious()) {\n        val element = iterator.previous()\n        if (predicate(element)) return element\n    }\n    return null\n}\n\n/**\n * Returns a random element from this collection.\n * \n * @throws
NoSuchElementException if this collection is empty.\n */\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline fun <T> Collection<T>.random(): T {\n    return random(Random)\n}\n\n/**\n * Returns a random element from this collection using the specified source of
randomness.\n * \n * @throws NoSuchElementException if this collection is empty.\n */\n@SinceKotlin("1.3")\npublic fun <T> Collection<T>.random(random: Random): T {\n    if (isEmpty())\n        throw NoSuchElementException("Collection is empty.")\n    return elementAt(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this collection, or `null` if this collection is empty.\n */\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
inline fun <T> Collection<T>.randomOrNull(): T? {\n    return randomOrNull(Random)\n}\n\n/**\n * Returns a
random element from this collection using the specified source of randomness, or `null` if this collection is empty.\n */\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun <T>
Collection<T>.randomOrNull(random: Random): T? {\n    if (isEmpty())\n        return null\n    return
elementAt(random.nextInt(size))\n}\n\n/**\n * Returns the single element, or throws an exception if the collection is
empty or has more than one element.\n */\npublic fun <T> Iterable<T>.single(): T {\n    when (this) {\n        is List -
> return this.single()\n        else -> {\n            val iterator = iterator()\n            if (!iterator.hasNext())\n                throw NoSuchElementException("Collection is empty.")\n            val single = iterator.next()\n            if
(iterator.hasNext())\n                throw IllegalArgumentException("Collection has more than one element.")\n            return single\n        }\n    }\n}\n\n/**\n * Returns the single element, or throws an exception if the list is empty or
has more than one element.\n */\npublic fun <T> List<T>.single(): T {\n    return when (size) {\n        0 -> throw
NoSuchElementException("List is empty.")\n        1 -> this[0]\n        else -> throw
IllegalArgumentException("List has more than one element.")\n    }\n}\n\n/**\n * Returns the single element
matching the given [predicate], or throws exception if there is no or more than one matching element.\n */\npublic
inline fun <T> Iterable<T>.single(predicate: (T) -> Boolean): T {\n    var single: T? = null\n    var found = false\n    for (element in this) {\n        if (predicate(element)) {\n            if (found) throw
IllegalArgumentException("Collection contains more than one matching element.")\n            single = element\n            found = true\n        }\n    }\n    if (!found) throw NoSuchElementException("Collection contains no element
matching the predicate.")\n    @Suppress("UNCHECKED_CAST")\n    return single as T\n}\n\n/**\n * Returns
single element, or `null` if the collection is empty or has more than one element.\n */\npublic fun <T>
Iterable<T>.singleOrNull(): T? {\n    when (this) {\n        is List -> return if (size == 1) this[0] else null\n        else ->
{\n            val iterator = iterator()\n            if (!iterator.hasNext())\n                return null\n            val single =
iterator.next()\n            if (iterator.hasNext())\n                return null\n            return single\n        }\n    }\n}\n\n/**\n * Returns single element, or `null` if the list is empty or has more than one element.\n */\npublic fun <T>

```

```

List<T>.singleOrNull(): T? {\n  return if (size == 1) this[0] else null\n}\n\n/**\n * Returns the single element matching the given [predicate], or `null` if element was not found or more than one element was found.\n */\npublic inline fun <T> Iterable<T>.singleOrNull(predicate: (T) -> Boolean): T? {\n  var single: T? = null\n  var found = false\n  for (element in this) {\n    if (predicate(element)) {\n      if (found) return null\n      single = element\n      found = true\n    }\n  }\n  if (!found) return null\n  return single\n}\n\n/**\n * Returns a list containing all elements except first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic fun <T> Iterable<T>.drop(n: Int): List<T> {\n  require(n >= 0) { \"Requested element count $n is less than zero.\" }\n  if (n == 0) return toList()\n  val list: ArrayList<T>\n  if (this is Collection<*>) {\n    val resultSize = size - n\n    if (resultSize <= 0)\n      return emptyList()\n    if (resultSize == 1)\n      return listOf(last())\n    list = ArrayList<T>(resultSize)\n    if (this is List<T>) {\n      if (this is RandomAccess) {\n        for (index in n until size)\n          list.add(this[index])\n      } else {\n        for (item in listIterator(n))\n          list.add(item)\n      }\n      return list\n    }\n  }\n  else {\n    list = ArrayList<T>()\n    var count = 0\n    for (item in this) {\n      if (count >= n) list.add(item) else ++count\n    }\n    return list.optimizeReadOnlyList()\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic fun <T> List<T>.dropLast(n: Int): List<T> {\n  require(n >= 0) { \"Requested element count $n is less than zero.\" }\n  return take((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last elements that satisfy the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic inline fun <T> List<T>.dropLastWhile(predicate: (T) -> Boolean): List<T> {\n  if (!isEmpty()) {\n    val iterator = listIterator(size)\n    while (iterator.hasPrevious()) {\n      if (!predicate(iterator.previous())) {\n        return take(iterator.nextIndex() + 1)\n      }\n    }\n  }\n  return emptyList()\n}\n\n/**\n * Returns a list containing all elements except first elements that satisfy the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic inline fun <T> Iterable<T>.dropWhile(predicate: (T) -> Boolean): List<T> {\n  var yielding = false\n  val list = ArrayList<T>()\n  for (item in this)\n    if (yielding)\n      list.add(item)\n    else if (!predicate(item)) {\n      list.add(item)\n      yielding = true\n    }\n  return list\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n */\npublic inline fun <T> Iterable<T>.filter(predicate: (T) -> Boolean): List<T> {\n  return filterTo(ArrayList<T>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * \n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexed\n */\npublic inline fun <T> Iterable<T>.filterIndexed(predicate: (index: Int, T) -> Boolean): List<T> {\n  return filterIndexedTo(ArrayList<T>(), predicate)\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n * \n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexedTo\n */\npublic inline fun <T, C : MutableCollection<in T>> Iterable<T>.filterIndexedTo(destination: C, predicate: (index: Int, T) -> Boolean): C {\n  forEachIndexed { index, element ->\n    if (predicate(index, element)) destination.add(element)\n  }\n  return destination\n}\n\n/**\n * Returns a list containing all elements that are instances of specified type parameter R.\n * \n * @sample samples.collections.Collections.Filtering.filterIsInstance\n */\npublic inline fun <reified R> Iterable<*>.filterIsInstance(): List<@kotlin.internal.NoInfer R> {\n  return filterIsInstanceTo(ArrayList<R>())\n}\n\n/**\n * Appends all elements that are instances of specified type parameter R to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterIsInstanceTo\n */\npublic inline fun <reified R, C : MutableCollection<in R>> Iterable<*>.filterIsInstanceTo(destination: C): C {\n  for (element in this) if (element is R) destination.add(element)\n  return destination\n}\n\n/**\n * Returns a list containing all elements not matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n */\npublic inline fun <T>

```

```

Iterable<T>.filterNot(predicate: (T) -> Boolean): List<T> {\n  return filterNotTo(ArrayList<T>(),
predicate)\n}\n\n/**\n * Returns a list containing all elements that are not `null`.\n * \n * @sample
samples.collections.Collections.Filtering.filterNotNull\n */\npublic fun <T : Any> Iterable<T?>.filterNotNull():
List<T> {\n  return filterNotNullTo(ArrayList<T>())\n}\n\n/**\n * Appends all elements that are not `null` to the
given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterNotNullTo\n */\npublic fun <C
: MutableCollection<in T>, T : Any> Iterable<T?>.filterNotNullTo(destination: C): C {\n  for (element in this) if
(element != null) destination.add(element)\n  return destination\n}\n\n/**\n * Appends all elements not matching
the given [predicate] to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n
*/\npublic inline fun <T, C : MutableCollection<in T>> Iterable<T>.filterNotTo(destination: C, predicate: (T) ->
Boolean): C {\n  for (element in this) if (!predicate(element)) destination.add(element)\n  return
destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n * \n *
@sample samples.collections.Collections.Filtering.filterTo\n */\npublic inline fun <T, C : MutableCollection<in
T>> Iterable<T>.filterTo(destination: C, predicate: (T) -> Boolean): C {\n  for (element in this) if
(predicate(element)) destination.add(element)\n  return destination\n}\n\n/**\n * Returns a list containing elements
at indices in the specified [indices] range.\n */\npublic fun <T> List<T>.slice(indices: IntRange): List<T> {\n  if
(indices.isEmpty()) return listOf()\n  return this.subList(indices.start, indices.endInclusive + 1).toList()\n}\n\n/**\n
* Returns a list containing elements at specified [indices].\n */\npublic fun <T> List<T>.slice(indices:
Iterable<Int>): List<T> {\n  val size = indices.collectionSizeOrDefault(10)\n  if (size == 0) return emptyList()\n
val list = ArrayList<T>(size)\n  for (index in indices) {\n    list.add(get(index))\n  }\n  return list\n}\n\n/**\n
* Returns a list containing first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n *
@sample samples.collections.Collections.Transformations.take\n */\npublic fun <T> Iterable<T>.take(n: Int):
List<T> {\n  require(n >= 0) { \"Requested element count $n is less than zero.\" }\n  if (n == 0) return
emptyList()\n  if (this is Collection<T>) {\n    if (n >= size) return toList()\n    if (n == 1) return
listOf(first())\n  }\n  var count = 0\n  val list = ArrayList<T>(n)\n  for (item in this) {\n    list.add(item)\n
if (++count == n)\n    break\n  }\n  return list.optimizeReadOnlyList()\n}\n\n/**\n * Returns a list containing
last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n */\npublic fun <T> List<T>.takeLast(n: Int): List<T> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n  if (n == 0) return emptyList()\n  val size =
size\n  if (n >= size) return toList()\n  if (n == 1) return listOf(last())\n  val list = ArrayList<T>(n)\n  if (this is
RandomAccess) {\n    for (index in size - n until size)\n      list.add(this[index])\n  } else {\n    for (item in
listIterator(size - n))\n      list.add(item)\n  }\n  return list\n}\n\n/**\n * Returns a list containing last elements
satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n */\npublic
inline fun <T> List<T>.takeLastWhile(predicate: (T) -> Boolean): List<T> {\n  if (isEmpty())\n    return
emptyList()\n  val iterator = listIterator(size)\n  while (iterator.hasPrevious()) {\n    if
(!predicate(iterator.previous())) {\n      iterator.next()\n      val expectedSize = size - iterator.nextIndex()\n
if (expectedSize == 0) return emptyList()\n      return ArrayList<T>(expectedSize).apply {\n        while
(iterator.hasNext())\n          add(iterator.next())\n        }\n      }\n    }\n  }\n  return toList()\n}\n\n/**\n
* Returns a list containing first elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n */\npublic inline fun <T> Iterable<T>.takeWhile(predicate:
(T) -> Boolean): List<T> {\n  val list = ArrayList<T>()\n  for (item in this) {\n    if (!predicate(item))\n
break\n    list.add(item)\n  }\n  return list\n}\n\n/**\n * Reverses elements in the list in-place.\n */\npublic
expect fun <T> MutableList<T>.reverse(): Unit\n\n/**\n * Returns a list with elements in reversed order.\n */\n
public fun <T> Iterable<T>.reversed(): List<T> {\n  if (this is Collection && size <= 1) return toList()\n  val
list = toMutableList()\n  list.reverse()\n  return list\n}\n\n/**\n * Randomly shuffles elements in this list in-place
using the specified [random] instance as the source of randomness.\n * \n * See:
https://en.wikipedia.org/wiki/Fisher%20%80%93Yates\_shuffle#The\_modern\_algorithm\n
*/\n\n@SinceKotlin(\"1.3\")\npublic fun <T> MutableList<T>.shuffle(random: Random): Unit {\n  for (i in lastIndex
downTo 1) {\n    val j = random.nextInt(i + 1)\n    this[j] = this.set(i, this[j])\n  }\n}\n\n/**\n * Sorts elements

```

in the list in-place according to natural sort order of the value returned by specified [selector] function.

`public inline fun <T, R : Comparable<R>> MutableList<T>.sortBy(crossinline selector: (T) -> R?): Unit` { if (size > 1) sortByWith(compareBy(selector)) }

`public inline fun <T, R : Comparable<R>> MutableList<T>.sortByDescending(crossinline selector: (T) -> R?): Unit` { if (size > 1) sortByWith(compareByDescending(selector)) }

`public fun <T : Comparable<T>> MutableList<T>.sortDescending(): Unit` { sortByWith(reverseOrder()) }

`public fun <T : Comparable<T>> Iterable<T>.sorted(): List<T>` { if (this is Collection) { if (size <= 1) return this.toList() @Suppress("UNCHECKED_CAST") return (toArray<Comparable<T>>() as Array<T>).apply { sort() }.asList() } return toMutableList().apply { sort() } }

`public fun <T : Comparable<T>> Iterable<T>.sortedBy(crossinline selector: (T) -> R?): List<T>` { return sortedWith(compareBy(selector)) }

`public fun <T, R : Comparable<R>> Iterable<T>.sortedByDescending(crossinline selector: (T) -> R?): List<T>` { return sortedWith(compareByDescending(selector)) }

`public fun <T : Comparable<T>> Iterable<T>.sortedDescending(): List<T>` { return sortedWith(reverseOrder()) }

`public fun <T> Iterable<T>.sortedWith(comparator: Comparator<in T>): List<T>` { if (this is Collection) { if (size <= 1) return this.toList() @Suppress("UNCHECKED_CAST") return (toArray<Any?>() as Array<T>).apply { sortByWith(comparator) }.asList() } return toMutableList().apply { sortByWith(comparator) } }

`public fun Collection<Boolean>.toBooleanArray(): BooleanArray` { val result = BooleanArray(size) var index = 0 for (element in this) result[index++] = element return result }

`public fun Collection<Byte>.toByteArray(): ByteArray` { val result = ByteArray(size) var index = 0 for (element in this) result[index++] = element return result }

`public fun Collection<Char>.toCharArray(): CharArray` { val result = CharArray(size) var index = 0 for (element in this) result[index++] = element return result }

`public fun Collection<Double>.toDoubleArray(): DoubleArray` { val result = DoubleArray(size) var index = 0 for (element in this) result[index++] = element return result }

`public fun Collection<Float>.toFloatArray(): FloatArray` { val result = FloatArray(size) var index = 0 for (element in this) result[index++] = element return result }

`public fun Collection<Int>.toIntArray(): IntArray` { val result = IntArray(size) var index = 0 for (element in this) result[index++] = element return result }

`public fun Collection<Long>.toLongArray(): LongArray` { val result =

```

LongArray(size)\n  var index = 0\n  for (element in this)\n    result[index++] = element\n  return
result\n}\n\n/**\n * Returns an array of Short containing all of the elements of this collection.\n */\npublic fun
Collection<Short>.toShortArray(): ShortArray {\n  val result = ShortArray(size)\n  var index = 0\n  for (element
in this)\n    result[index++] = element\n  return result\n}\n\n/**\n * Returns a [Map] containing key-value pairs
provided by [transform] function\n * applied to elements of the given collection.\n * \n * If any of two pairs would
have the same key the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order
of the original collection.\n * \n * @sample samples.collections.Collections.Transformations.associate\n */\npublic
inline fun <T, K, V> Iterable<T>.associate(transform: (T) -> Pair<K, V>): Map<K, V> {\n  val capacity =
mapCapacity(collectionSizeOrDefault(10)).coerceAtLeast(16)\n  return associateTo(LinkedHashMap<K,
V>(capacity), transform)\n}\n\n/**\n * Returns a [Map] containing the elements from the given collection indexed
by the key\n * returned from [keySelector] function applied to each element.\n * \n * If any two elements would
have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves
the entry iteration order of the original collection.\n * \n * @sample
samples.collections.Collections.Transformations.associateBy\n */\npublic inline fun <T, K>
Iterable<T>.associateBy(keySelector: (T) -> K): Map<K, T> {\n  val capacity =
mapCapacity(collectionSizeOrDefault(10)).coerceAtLeast(16)\n  return associateByTo(LinkedHashMap<K,
T>(capacity), keySelector)\n}\n\n/**\n * Returns a [Map] containing the values provided by [valueTransform] and
indexed by [keySelector] functions applied to elements of the given collection.\n * \n * If any two elements would
have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves
the entry iteration order of the original collection.\n * \n * @sample
samples.collections.Collections.Transformations.associateByWithValueTransform\n */\npublic inline fun <T, K, V>
Iterable<T>.associateBy(keySelector: (T) -> K, valueTransform: (T) -> V): Map<K, V> {\n  val capacity =
mapCapacity(collectionSizeOrDefault(10)).coerceAtLeast(16)\n  return associateByTo(LinkedHashMap<K,
V>(capacity), keySelector, valueTransform)\n}\n\n/**\n * Populates and returns the [destination] mutable map with
key-value pairs,\n * where key is provided by the [keySelector] function applied to each element of the given
collection\n * and value is the element itself.\n * \n * If any two elements would have the same key returned by
[keySelector] the last one gets added to the map.\n * \n * @sample
samples.collections.Collections.Transformations.associateByTo\n */\npublic inline fun <T, K, M : MutableMap<in
K, in T>> Iterable<T>.associateByTo(destination: M, keySelector: (T) -> K): M {\n  for (element in this) {\n
destination.put(keySelector(element), element)\n  }\n  return destination\n}\n\n/**\n * Populates and returns the
[destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function and\n *
value is provided by the [valueTransform] function applied to elements of the given collection.\n * \n * If any two
elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * @sample
samples.collections.Collections.Transformations.associateByToWithValueTransform\n */\npublic inline fun <T, K,
V, M : MutableMap<in K, in V>> Iterable<T>.associateByTo(destination: M, keySelector: (T) -> K,
valueTransform: (T) -> V): M {\n  for (element in this) {\n    destination.put(keySelector(element),
valueTransform(element))\n  }\n  return destination\n}\n\n/**\n * Populates and returns the [destination] mutable
map with key-value pairs\n * provided by [transform] function applied to each element of the given collection.\n *
\n * If any of two pairs would have the same key the last one gets added to the map.\n * \n * @sample
samples.collections.Collections.Transformations.associateTo\n */\npublic inline fun <T, K, V, M : MutableMap<in
K, in V>> Iterable<T>.associateTo(destination: M, transform: (T) -> Pair<K, V>): M {\n  for (element in this) {\n
destination += transform(element)\n  }\n  return destination\n}\n\n/**\n * Returns a [Map] where keys are
elements from the given collection and values are\n * produced by the [valueSelector] function applied to each
element.\n * \n * If any two elements are equal, the last one gets added to the map.\n * \n * The returned map
preserves the entry iteration order of the original collection.\n * \n * @sample
samples.collections.Collections.Transformations.associateWith\n */\n@SinceKotlin("1.3")\npublic inline fun <K,
V> Iterable<K>.associateWith(valueSelector: (K) -> V): Map<K, V> {\n  val result = LinkedHashMap<K,
V>(mapCapacity(collectionSizeOrDefault(10)).coerceAtLeast(16))\n  return associateWithTo(result,

```

valueSelector)\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs for each element of the given collection,\n * where key is the element itself and value is provided by the [valueSelector] function applied to that key.\n * \n * If any two elements are equal, the last one overwrites the former value in the map.\n * \n * @sample samples.collections.Collections.Transformations.associateWithTo\n

```

*/\n@SinceKotlin("1.3")\npublic inline fun <K, V, M : MutableMap<in K, in V>>
Iterable<K>.associateWithTo(destination: M, valueSelector: (K) -> V): M {\n    for (element in this) {\n        destination.put(element, valueSelector(element))\n    }\n    return destination\n}\n\n/**\n * Appends all elements to the given [destination] collection.\n */\npublic fun <T, C : MutableCollection<in T>>
Iterable<T>.toCollection(destination: C): C {\n    for (item in this) {\n        destination.add(item)\n    }\n    return destination\n}\n\n/**\n * Returns a new [HashSet] of all elements.\n */\npublic fun <T> Iterable<T>.toHashSet(): HashSet<T> {\n    return toCollection(HashSet<T>(mapCapacity(collectionSizeOrDefault(12))))\n}\n\n/**\n * Returns a [List] containing all elements.\n */\npublic fun <T> Iterable<T>.toList(): List<T> {\n    if (this is Collection) {\n        return when (size) {\n            0 -> emptyList()\n            1 -> listOf(if (this is List) get(0) else iterator().next())\n            else -> this.toMutableList()\n        }\n    }\n    return this.toMutableList().optimizeReadOnlyList()\n}\n\n/**\n * Returns a new [MutableList] filled with all elements of this collection.\n */\npublic fun <T> Iterable<T>.toMutableList(): MutableList<T> {\n    if (this is Collection<T>)\n        return this.toMutableList()\n    return toCollection(ArrayList<T>())\n}\n\n/**\n * Returns a new [MutableList] filled with all elements of this collection.\n */\npublic fun <T> Collection<T>.toMutableList(): MutableList<T> {\n    return ArrayList(this)\n}\n\n/**\n * Returns a [Set] of all elements.\n * \n * The returned set preserves the element iteration order of the original collection.\n */\npublic fun <T> Iterable<T>.toSet(): Set<T> {\n    if (this is Collection) {\n        return when (size) {\n            0 -> emptySet()\n            1 -> setOf(if (this is List) this[0] else iterator().next())\n            else -> toCollection(LinkedHashSet<T>(mapCapacity(size)))\n        }\n    }\n    return toCollection(LinkedHashSet<T>()).optimizeReadOnlySet()\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being invoked on each element of original collection.\n * \n * @sample samples.collections.Collections.Transformations.flatMap\n */\npublic inline fun <T, R>
Iterable<T>.flatMap(transform: (T) -> Iterable<R>): List<R> {\n    return flatMapTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being invoked on each element of original collection.\n * \n * @sample samples.collections.Collections.Transformations.flatMap\n */\npublic inline fun <T, R>
Iterable<T>.flatMap(transform: (T) -> Sequence<R>): List<R> {\n    return flatMapTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being invoked on each element\n * and its index in the original collection.\n * \n * @sample samples.collections.Collections.Transformations.flatMapIndexed\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapSequence")\npublic inline fun <T, R>
Iterable<T>.flatMapIndexed(transform: (index: Int, T) -> Iterable<R>): List<R> {\n    return flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being invoked on each element\n * and its index in the original collection.\n * \n * @sample samples.collections.Collections.Transformations.flatMapIndexed\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterable")\n@kotlin.internal.InlineOnly\npublic inline fun <T, R>
Iterable<T>.flatMapIndexed(transform: (index: Int, T) -> Sequence<R>): List<R> {\n    return flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each element\n * and its index in the original collection, to the given [destination].\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution

```

```

ByLambdaReturnType\n@kotlin.jvm.JvmName("\flatMapIndexedIterableTo")\n@kotlin.internal.InlineOnly\npubli
c inline fun <T, R, C : MutableCollection<in R>> Iterable<T>.flatMapIndexedTo(destination: C, transform: (index:
Int, T) -> Iterable<R>): C {\n    var index = 0\n    for (element in this) {\n        val list =
transform(checkIndexOverflow(index++), element)\n        destination.addAll(list)\n    }\n    return
destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each
element\n * and its index in the original collection, to the given [destination].\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("\flatMapIndexedSequenceTo")\n@kotlin.internal.InlineOnly\npu
blic inline fun <T, R, C : MutableCollection<in R>> Iterable<T>.flatMapIndexedTo(destination: C, transform:
(index: Int, T) -> Sequence<R>): C {\n    var index = 0\n    for (element in this) {\n        val list =
transform(checkIndexOverflow(index++), element)\n        destination.addAll(list)\n    }\n    return
destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each
element of original collection, to the given [destination].\n
*\npublic inline fun <T, R, C : MutableCollection<in
R>> Iterable<T>.flatMapTo(destination: C, transform: (T) -> Iterable<R>): C {\n    for (element in this) {\n        val
list = transform(element)\n        destination.addAll(list)\n    }\n    return destination\n}\n\n/**\n * Appends all
elements yielded from results of [transform] function being invoked on each element of original collection, to the
given [destination].\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("\flatMapSequenceTo")\npublic inline fun <T, R, C :
MutableCollection<in R>> Iterable<T>.flatMapTo(destination: C, transform: (T) -> Sequence<R>): C {\n    for
(element in this) {\n        val list = transform(element)\n        destination.addAll(list)\n    }\n    return
destination\n}\n\n/**\n * Groups elements of the original collection by the key returned by the given [keySelector]
function\n * applied to each element and returns a map where each group key is associated with a list of
corresponding elements.\n * \n * The returned map preserves the entry iteration order of the keys produced from the
original collection.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n
*\npublic inline
fun <T, K> Iterable<T>.groupBy(keySelector: (T) -> K): Map<K, List<T>> {\n    return
groupByTo(LinkedHashMap<K, MutableList<T>>(), keySelector)\n}\n\n/**\n * Groups values returned by the
[valueTransform] function applied to each element of the original collection\n * by the key returned by the given
[keySelector] function applied to the element\n * and returns a map where each group key is associated with a list of
corresponding values.\n * \n * The returned map preserves the entry iteration order of the keys produced from the
original collection.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n
*\npublic inline
fun <T, K, V> Iterable<T>.groupBy(keySelector: (T) -> K, valueTransform: (T) -> V): Map<K,
List<V>> {\n    return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector,
valueTransform)\n}\n\n/**\n * Groups elements of the original collection by the key returned by the given
[keySelector] function\n * applied to each element and puts to the [destination] map each group key associated with
a list of corresponding elements.\n * \n * @return The [destination] map.\n * \n * @sample
samples.collections.Collections.Transformations.groupBy\n
*\npublic inline fun <T, K, M : MutableMap<in K,
MutableList<T>>> Iterable<T>.groupByTo(destination: M, keySelector: (T) -> K): M {\n    for (element in this) {\n
        val key = keySelector(element)\n        val list = destination.getOrPut(key) { ArrayList<T>() }\n
list.add(element)\n    }\n    return destination\n}\n\n/**\n * Groups values returned by the [valueTransform] function
applied to each element of the original collection\n * by the key returned by the given [keySelector] function applied
to the element\n * and puts to the [destination] map each group key associated with a list of corresponding values.\n
*\n * @return The [destination] map.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeysAndValues\n
*\npublic inline fun <T, K, V, M :
MutableMap<in K, MutableList<V>>> Iterable<T>.groupByTo(destination: M, keySelector: (T) -> K,
valueTransform: (T) -> V): M {\n    for (element in this) {\n        val key = keySelector(element)\n        val list =
destination.getOrPut(key) { ArrayList<V>() }\n        list.add(valueTransform(element))\n    }\n    return
destination\n}\n\n/**\n * Creates a [Grouping] source from a collection to be used later with one of group-and-fold

```

operations using the specified [keySelector] function to extract a key from each element.

```

@sample
samples.collections.Grouping.groupingByEachCount
public inline fun <T, K>
Iterable<T>.groupingBy(crossinline keySelector: (T) -> K): Grouping<T, K> {
    return object : Grouping<T, K> {
        override fun sourceIterator(): Iterator<T> = this@groupingBy.iterator()
        override fun keyOf(element: T): K = keySelector(element)
    }
}

```

Returns a list containing the results of applying the given [transform] function to each element in the original collection.

```

@sample
samples.collections.Collections.Transformations.map
public inline fun <T, R> Iterable<T>.map(transform: (T) -> R): List<R> {
    return mapTo(ArrayList<R>(collectionSizeOrDefault(10)), transform)
}

```

Returns a list containing the results of applying the given [transform] function to each element and its index in the original collection.

```

@param [transform] function that takes the index of an element and the element itself
and returns the result of the transform applied to the element.
public inline fun <T, R>
Iterable<T>.mapIndexed(transform: (index: Int, T) -> R): List<R> {
    return
mapIndexedTo(ArrayList<R>(collectionSizeOrDefault(10)), transform)
}

```

Returns a list containing only the non-null results of applying the given [transform] function to each element and its index in the original collection.

```

@param [transform] function that takes the index of an element and the element itself
and returns the result of the transform applied to the element.
public inline fun <T, R : Any>
Iterable<T>.mapIndexedNotNull(transform: (index: Int, T) -> R?): List<R> {
    return
mapIndexedNotNullTo(ArrayList<R>(), transform)
}

```

Applies the given [transform] function to each element and its index in the original collection and appends only the non-null results to the given [destination].

```

@param [transform] function that takes the index of an element and the element itself
and returns the result of the transform applied to the element.
public inline fun <T, R : Any, C : MutableCollection<in R>>
Iterable<T>.mapIndexedNotNullTo(destination: C, transform: (index: Int, T) -> R?): C {
    forEachIndexed { index, element -> transform(index, element)?.let { destination.add(it) }
}
return destination
}

```

Applies the given [transform] function to each element and its index in the original collection and appends the results to the given [destination].

```

@param [transform] function that takes the index of an element and the
element itself
and returns the result of the transform applied to the element.
public inline fun <T, R, C :
MutableCollection<in R>> Iterable<T>.mapIndexedTo(destination: C, transform: (index: Int, T) -> R): C {
    var index = 0
    for (item in this)
        destination.add(transform(checkIndexOverflow(index++), item))
    return destination
}

```

Returns a list containing only the non-null results of applying the given [transform] function to each element in the original collection.

```

@sample
samples.collections.Collections.Transformations.mapNotNull
public inline fun <T, R : Any>
Iterable<T>.mapNotNull(transform: (T) -> R?): List<R> {
    return mapNotNullTo(ArrayList<R>(),
transform)
}

```

Applies the given [transform] function to each element in the original collection and appends only the non-null results to the given [destination].

```

public inline fun <T, R : Any, C :
MutableCollection<in R>> Iterable<T>.mapNotNullTo(destination: C, transform: (T) -> R?): C {
    forEach { element -> transform(element)?.let { destination.add(it) }
}
return destination
}

```

Applies the given [transform] function to each element of the original collection and appends the results to the given [destination].

```

public inline fun <T, R, C : MutableCollection<in R>> Iterable<T>.mapTo(destination: C,
transform: (T) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

```

Returns a lazy [Iterable] that wraps each element of the original collection into an [IndexedValue] containing the index of that element and the element itself.

```

public fun <T>
Iterable<T>.withIndex(): Iterable<IndexedValue<T>> {
    return IndexingIterable { iterator() }
}

```

Returns a list containing only distinct elements from the given collection.

```

Among equal elements of the
given collection, only the first one will be present in the resulting list.
The elements in the resulting list are in the
same order as they were in the source collection.
@sample
samples.collections.Collections.Transformations.distinctAndDistinctBy
public fun <T> Iterable<T>.distinct():
List<T> {
    return this.toMutableSet().toList()
}

```

Returns a list containing only elements from the given collection having distinct keys returned by the given [selector] function.

```

Among elements of the

```

given collection with equal keys, only the first one will be present in the resulting list.
 The elements in the resulting list are in the same order as they were in the source collection.
 @sample
 samples.collections.Collections.Transformations.distinctAndDistinctBy
 public inline fun <T, K> Iterable<T>.distinctBy(selector: (T) -> K): List<T> {
 val set = HashSet<K>()
 val list = ArrayList<T>()
 for (e in this) {
 val key = selector(e)
 if (set.add(key))
 list.add(e)
 }
 return list
 }
 Returns a set containing all elements that are contained by both this collection and the specified collection.
 The returned set preserves the element iteration order of the original collection.
 To get a set containing all elements that are contained at least in one of these collections use [union].
 public infix fun <T> Iterable<T>.intersect(other: Iterable<T>): Set<T> {
 val set = this.toMutableSet()
 set.retainAll(other)
 return set
 }
 Returns a set containing all elements that are contained by this collection and not contained by the specified collection.
 The returned set preserves the element iteration order of the original collection.
 public infix fun <T> Iterable<T>.subtract(other: Iterable<T>): Set<T> {
 val set = this.toMutableSet()
 set.removeAll(other)
 return set
 }
 Returns a new [MutableSet] containing all distinct elements from the given collection.
 The returned set preserves the element iteration order of the original collection.
 public fun <T> Iterable<T>.toMutableSet(): MutableSet<T> {
 return when (this) {
 is Collection<T> -> LinkedHashSet(this)
 else -> toCollection(LinkedHashSet<T>())
 }
 }
 Returns a set containing all distinct elements from both collections.
 The returned set preserves the element iteration order of the original collection.
 Those elements of the [other] collection that are unique are iterated in the end
 in the order of the [other] collection.
 To get a set containing all elements that are contained in both collections use [intersect].
 public infix fun <T> Iterable<T>.union(other: Iterable<T>): Set<T> {
 val set = this.toMutableSet()
 set.addAll(other)
 return set
 }
 Returns `true` if all elements match the given [predicate].
 @sample
 samples.collections.Collections.Aggregates.all
 public inline fun <T> Iterable<T>.all(predicate: (T) -> Boolean): Boolean {
 if (this is Collection && isEmpty()) return true
 for (element in this) if (!predicate(element)) return false
 return true
 }
 Returns `true` if collection has at least one element.
 @sample
 samples.collections.Collections.Aggregates.any
 public fun <T> Iterable<T>.any(): Boolean {
 if (this is Collection) return !isEmpty()
 return iterator().hasNext()
 }
 Returns `true` if at least one element matches the given [predicate].
 @sample
 samples.collections.Collections.Aggregates.anyWithPredicate
 public inline fun <T> Iterable<T>.any(predicate: (T) -> Boolean): Boolean {
 if (this is Collection && isEmpty()) return false
 for (element in this) if (predicate(element)) return true
 return false
 }
 Returns the number of elements in this collection.
 public fun <T> Iterable<T>.count(): Int {
 if (this is Collection) return size
 var count = 0
 for (element in this) checkCountOverflow(++count)
 return count
 }
 Returns the number of elements in this collection.
 @kotlin.internal.InlineOnly
 public inline fun <T> Collection<T>.count(): Int {
 return size
 }
 Returns the number of elements matching the given [predicate].
 public inline fun <T> Iterable<T>.count(predicate: (T) -> Boolean): Int {
 if (this is Collection && isEmpty()) return 0
 var count = 0
 for (element in this) if (predicate(element)) checkCountOverflow(++count)
 return count
 }
 Accumulates value starting with [initial] value and applying [operation] from left to right
 to current accumulator value and each element.
 Returns the specified [initial] value if the collection is empty.
 @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 public inline fun <T, R> Iterable<T>.fold(initial: R, operation: (acc: R, T) -> R): R {
 var accumulator = initial
 for (element in this) accumulator = operation(accumulator, element)
 return accumulator
 }
 Accumulates value starting with [initial] value and applying [operation] from left to right
 to current accumulator value and each element with its index in the original collection.
 Returns the specified [initial] value if the collection is empty.
 @param [operation] function that takes the index of an element, current accumulator value
 and the element itself, and calculates the next accumulator value.
 public inline fun <T, R> Iterable<T>.foldIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): R {
 var index = 0
 var accumulator = initial
 for (element in this) accumulator = operation(checkIndexOverflow(index++), accumulator, element)
 return accumulator
 }
 Accumulates

if the collection is empty.\n

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.maxOf(selector: (T) -> Float): Float {\n    val iterator = iterator()\n    if (!iterator.hasNext()) throw NoSuchElementException()\n    var\n    max\n    Value = selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v = selector(iterator.next())\n        max\n    Value = maxOf(max\n    Value, v)\n    }\n    return max\n    Value\n}\n\n/**\n * Returns the largest value among all\n * values produced by [selector] function\n * applied to each element in the collection.\n * @throws\n * NoSuchElementException if the collection is empty.\n */
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Comparable<R>>\nIterable<T>.maxOf(selector: (T) -> R): R {\n    val iterator = iterator()\n    if (!iterator.hasNext()) throw\n    NoSuchElementException()\n    var max\n    Value = selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v\n    = selector(iterator.next())\n        if (max\n    Value < v) {\n            max\n    Value = v\n        }\n    }\n    return\n    max\n    Value\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the collection or `null` if there are no elements.\n * If any of values produced by [selector]\n * function is `NaN`, the returned result is `NaN`.\n */
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.maxOfOrNull(selector: (T)\n-> Double): Double? {\n    val iterator = iterator()\n    if (!iterator.hasNext()) return null\n    var max\n    Value =\n    selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v = selector(iterator.next())\n        max\n    Value =\n    maxOf(max\n    Value, v)\n    }\n    return max\n    Value\n}\n\n/**\n * Returns the largest value among all values produced\n * by [selector] function\n * applied to each element in the collection or `null` if there are no elements.\n * If any\n * of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n */
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.maxOfOrNull(selector: (T)\n-> Float): Float? {\n    val iterator = iterator()\n    if (!iterator.hasNext()) return null\n    var max\n    Value =\n    selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v = selector(iterator.next())\n        max\n    Value =\n    maxOf(max\n    Value, v)\n    }\n    return max\n    Value\n}\n\n/**\n * Returns the largest value among all values produced\n * by [selector] function\n * applied to each element in the collection or `null` if there are no elements.\n */
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Comparable<R>>\nIterable<T>.maxOfOrNull(selector: (T) -> R): R? {\n    val iterator = iterator()\n    if (!iterator.hasNext()) return\n    null\n    var max\n    Value = selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v =\n    selector(iterator.next())\n        if (max\n    Value < v) {\n            max\n    Value = v\n        }\n    }\n    return\n    max\n    Value\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values\n * produced by [selector] function applied to each element in the collection.\n * @throws\n * NoSuchElementException if the collection is empty.\n */
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R>\nIterable<T>.maxOfWith(comparator: Comparator<in R>, selector: (T) -> R): R {\n    val iterator = iterator()\n    if\n    (!iterator.hasNext()) throw NoSuchElementException()\n    var max\n    Value = selector(iterator.next())\n    while\n    (iterator.hasNext()) {\n        val v = selector(iterator.next())\n        if (comparator.compare(max\n    Value, v) < 0) {\n            max\n    Value = v\n        }\n    }\n    return max\n    Value\n}\n\n/**\n * Returns the largest value according to the provided\n * [comparator]\n * among all values produced by [selector] function applied to each element in the collection or `null`\n * if there are no elements.\n */
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R>\nIterable<T>.maxOfWithOrNull(comparator: Comparator<in R>, selector: (T) -> R): R? {\n    val iterator =
```

```

iterator()\n if (!iterator.hasNext()) return null\n var max = selector(iterator.next())\n while
(iterator.hasNext()) {\n val v = selector(iterator.next())\n if (comparator.compare(max, v) < 0) {\n
max = v\n }\n }\n return max\n}\n\n/**\n * Returns the largest element or `null` if there are no
elements.\n * \n * If any of elements is `NaN` returns `NaN`.\n */\n@SinceKotlin("1.4")\npublic fun
Iterable<Double>.maxOrNull(): Double? {\n val iterator = iterator()\n if (!iterator.hasNext()) return null\n var
max = iterator.next()\n while (iterator.hasNext()) {\n val e = iterator.next()\n max = maxOf(max, e)\n
}\n return max\n}\n\n/**\n * Returns the largest element or `null` if there are no elements.\n * \n * If any of
elements is `NaN` returns `NaN`.\n */\n@SinceKotlin("1.4")\npublic fun Iterable<Float>.maxOrNull(): Float? {\n
val iterator = iterator()\n if (!iterator.hasNext()) return null\n var max = iterator.next()\n while
(iterator.hasNext()) {\n val e = iterator.next()\n max = maxOf(max, e)\n }\n return max\n}\n\n/**\n *
Returns the largest element or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\npublic fun <T> :
Comparable<T>> Iterable<T>.maxOrNull(): T? {\n val iterator = iterator()\n if (!iterator.hasNext()) return null\n
var max = iterator.next()\n while (iterator.hasNext()) {\n val e = iterator.next()\n if (max < e) max = e\n
}\n return max\n}\n\n@Deprecated("Use maxWithOrNull instead.")\nReplaceWith("this.maxWithOrNull(comparator)")\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince
= "1.5", hiddenSince = "1.6")\npublic fun <T> Iterable<T>.maxWith(comparator: Comparator<in T>): T? {\n
return maxWithOrNull(comparator)\n}\n\n/**\n * Returns the first element having the largest value according to the
provided [comparator] or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\npublic fun <T>
Iterable<T>.maxWithOrNull(comparator: Comparator<in T>): T? {\n val iterator = iterator()\n if
(!iterator.hasNext()) return null\n var max = iterator.next()\n while (iterator.hasNext()) {\n val e =
iterator.next()\n if (comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\n@Deprecated("Use
minOrNull instead.")\nReplaceWith("this.minOrNull()")\n@DeprecatedSinceKotlin(warningSince = "1.4",
errorSince = "1.5", hiddenSince = "1.6")\n@SinceKotlin("1.1")\npublic fun Iterable<Double>.min(): Double?
{\n return minOrNull()\n}\n\n@Deprecated("Use minOrNull instead.")\nReplaceWith("this.minOrNull()")\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5",
hiddenSince = "1.6")\n@SinceKotlin("1.1")\npublic fun Iterable<Float>.min(): Float? {\n return
minOrNull()\n}\n\n@Deprecated("Use minOrNull instead.")\nReplaceWith("this.minOrNull()")\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5",
hiddenSince = "1.6")\npublic fun <T : Comparable<T>> Iterable<T>.min(): T? {\n return
minOrNull()\n}\n\n@Deprecated("Use minByOrNull instead.")\nReplaceWith("this.minByOrNull(selector)")\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince =
"1.5", hiddenSince = "1.6")\npublic inline fun <T, R : Comparable<R>> Iterable<T>.minBy(selector: (T) -> R):
T? {\n return minByOrNull(selector)\n}\n\n/**\n * Returns the first element yielding the smallest value of the
given function or `null` if there are no elements.\n * \n * @sample
samples.collections.Collections.Aggregates.minByOrNull\n */\n@SinceKotlin("1.4")\npublic inline fun <T, R :
Comparable<R>> Iterable<T>.minByOrNull(selector: (T) -> R): T? {\n val iterator = iterator()\n if
(!iterator.hasNext()) return null\n var minElem = iterator.next()\n if (!iterator.hasNext()) return minElem\n
var min = selector(minElem)\n do {\n val e = iterator.next()\n val v = selector(e)\n if (min >
v) {\n min = v\n minElem = e\n }\n } while (iterator.hasNext())\n return
minElem\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to
each element in the collection.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result
is `NaN`.\n * \n * @throws NoSuchElementException if the collection is empty.\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.minOf(selector: (T) ->
Double): Double {\n val iterator = iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var
min = selector(iterator.next())\n while (iterator.hasNext()) {\n val v = selector(iterator.next())\n
min = minOf(min, v)\n }\n return min\n}\n\n/**\n * Returns the smallest value among all
values produced by [selector] function\n * applied to each element in the collection.\n * \n * If any of values

```

produced by [selector] function is `NaN`, the returned result is `NaN`.
if the collection is empty.

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.minOf(selector: (T) -> Float): Float {\n    val iterator = iterator()\n    if (!iterator.hasNext()) throw NoSuchElementException()\n    var\n    minValue = selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v = selector(iterator.next())\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all\n * values produced by [selector] function\n * applied to each element in the collection.\n * @throws\n * NoSuchElementException if the collection is empty.
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Comparable<R>>\nIterable<T>.minOf(selector: (T) -> R): R {\n    val iterator = iterator()\n    if (!iterator.hasNext()) throw\n    NoSuchElementException()\n    var minValue = selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v\n        = selector(iterator.next())\n        if (minValue > v) {\n            minValue = v\n        }\n    }\n    return\n    minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to\n * each element in the collection or `null` if there are no elements.\n * @throws\n * NoSuchElementException if any of values produced by [selector]\n * function is `NaN`, the returned result is `NaN`.
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.minOfOrNull(selector: (T)\n-> Double): Double? {\n    val iterator = iterator()\n    if (!iterator.hasNext()) return null\n    var minValue =\n    selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v = selector(iterator.next())\n        minValue =\n        minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all values produced\n * by [selector] function\n * applied to each element in the collection or `null` if there are no elements.\n * @throws\n * NoSuchElementException if any\n * of values produced by [selector] function is `NaN`, the returned result is `NaN`.
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.minOfOrNull(selector: (T)\n-> Float): Float? {\n    val iterator = iterator()\n    if (!iterator.hasNext()) return null\n    var minValue =\n    selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v = selector(iterator.next())\n        minValue =\n        minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all values produced\n * by [selector] function\n * applied to each element in the collection or `null` if there are no elements.
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Comparable<R>>\nIterable<T>.minOfOrNull(selector: (T) -> R): R? {\n    val iterator = iterator()\n    if (!iterator.hasNext()) return\n    null\n    var minValue = selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v =\n        selector(iterator.next())\n        if (minValue > v) {\n            minValue = v\n        }\n    }\n    return\n    minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values\n * produced by [selector] function applied to each element in the collection.\n * @throws\n * NoSuchElementException if the collection is empty.
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R> Iterable<T>.minOfWith(comparator:\nComparator<in R>, selector: (T) -> R): R {\n    val iterator = iterator()\n    if (!iterator.hasNext()) throw\n    NoSuchElementException()\n    var minValue = selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v\n        = selector(iterator.next())\n        if (comparator.compare(minValue, v) > 0) {\n            minValue = v\n        }\n    }\n    return\n    minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all\n * values produced by [selector] function applied to each element in the collection or `null` if there are no elements.
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R>\nIterable<T>.minOfWithOrNull(comparator: Comparator<in R>, selector: (T) -> R): R? {\n    val iterator =
```

```

iterator()\n if (!iterator.hasNext()) return null\n var minValue = selector(iterator.next())\n while
(iterator.hasNext()) {\n val v = selector(iterator.next())\n if (comparator.compare(minValue, v) > 0) {\n
minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest element or `null` if there are
no elements.\n * \n * If any of elements is `NaN` returns `NaN`.\n */\n@SinceKotlin("1.4")\npublic fun
Iterable<Double>.minOrNull(): Double? {\n val iterator = iterator()\n if (!iterator.hasNext()) return null\n var
min = iterator.next()\n while (iterator.hasNext()) {\n val e = iterator.next()\n min = minOf(min, e)\n }\n
return min\n}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n * \n * If any of elements
is `NaN` returns `NaN`.\n */\n@SinceKotlin("1.4")\npublic fun Iterable<Float>.minOrNull(): Float? {\n val
iterator = iterator()\n if (!iterator.hasNext()) return null\n var min = iterator.next()\n while (iterator.hasNext())
{\n val e = iterator.next()\n min = minOf(min, e)\n }\n return min\n}\n\n/**\n * Returns the smallest
element or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\npublic fun <T : Comparable<T>>
Iterable<T>.minOrNull(): T? {\n val iterator = iterator()\n if (!iterator.hasNext()) return null\n var min =
iterator.next()\n while (iterator.hasNext()) {\n val e = iterator.next()\n if (min > e) min = e\n }\n return
min\n}\n\n@Deprecated("Use minWithOrNull instead.")
ReplaceWith("this.minWithOrNull(comparator)")\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince
= "1.5", hiddenSince = "1.6")\npublic fun <T> Iterable<T>.minWith(comparator: Comparator<in T>): T? {\n
return minWithOrNull(comparator)\n}\n\n/**\n * Returns the first element having the smallest value according to
the provided [comparator] or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\npublic fun <T>
Iterable<T>.minWithOrNull(comparator: Comparator<in T>): T? {\n val iterator = iterator()\n if
(!iterator.hasNext()) return null\n var min = iterator.next()\n while (iterator.hasNext()) {\n val e =
iterator.next()\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns `true` if
the collection has no elements.\n * \n * @sample samples.collections.Collections.Aggregates.none\n */\n@public fun
<T> Iterable<T>.none(): Boolean {\n if (this is Collection) return isEmpty()\n return
!iterator().hasNext()\n}\n\n/**\n * Returns `true` if no elements match the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.noneWithPredicate\n */\n@public inline fun <T>
Iterable<T>.none(predicate: (T) -> Boolean): Boolean {\n if (this is Collection && isEmpty()) return true\n for
(element in this) if (predicate(element)) return false\n return true\n}\n\n/**\n * Performs the given [action] on each
element and returns the collection itself afterwards.\n */\n@SinceKotlin("1.1")\n@public inline fun <T, C :
Iterable<T>> C.onEach(action: (T) -> Unit): C {\n return apply { for (element in this) action(element)
}\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element,\n * and
returns the collection itself afterwards.\n * \n * @param [action] function that takes the index of an element and the
element itself\n * and performs the action on the element.\n */\n@SinceKotlin("1.4")\n@public inline fun <T, C :
Iterable<T>> C.onEachIndexed(action: (index: Int, T) -> Unit): C {\n return apply { forEachIndexed(action)
}\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to
current accumulator value and each element.\n * \n * Throws an exception if this collection is empty. If the
collection can be empty in an expected way,\n * please use [reduceOrNull] instead. It returns `null` when its receiver
is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and
calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\n@public inline fun <S, T : S> Iterable<T>.reduce(operation: (acc: S, T) -> S): S {\n val iterator = this.iterator()\n
if (!iterator.hasNext()) throw UnsupportedOperationException("Empty collection can't be reduced.")\n var
accumulator: S = iterator.next()\n while (iterator.hasNext()) {\n accumulator = operation(accumulator,
iterator.next())\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and
applying [operation] from left to right\n * to current accumulator value and each element with its index in the
original collection.\n * \n * Throws an exception if this collection is empty. If the collection can be empty in an
expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n *
@param [operation] function that takes the index of an element, current accumulator value and the element itself,\n
* and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\n@public inline fun <S, T : S> Iterable<T>.reduceIndexed(operation: (index: Int, acc: S, T) -> S): S {\n val

```

```

iterator = this.iterator()\n  if (!iterator.hasNext()) throw UnsupportedOperationException("\nEmpty collection can't\nbe reduced.\n")\n  var index = 1\n  var accumulator: S = iterator.next()\n  while (iterator.hasNext()) {\n  accumulator = operation(checkIndexOverflow(index++), accumulator, iterator.next())\n  }\n  return\n  accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to\nright\n * to current accumulator value and each element with its index in the original collection.\n * \n * Returns\n`null` if the collection is empty.\n * \n * @param [operation] function that takes the index of an element, current\naccumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample\nsamples.collections.Collections.Aggregates.reduceOrNull\n * \n * @SinceKotlin("1.4")\n\npublic inline fun <S, T : S>\nIterable<T>.reduceIndexedOrNull(operation: (index: Int, acc: S, T) -> S): S? {\n  val iterator = this.iterator()\n  if\n(!iterator.hasNext()) return null\n  var index = 1\n  var accumulator: S = iterator.next()\n  while\n(iterator.hasNext()) {\n    accumulator = operation(checkIndexOverflow(index++), accumulator, iterator.next())\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation]\nfrom left to right\n * to current accumulator value and each element.\n * \n * Returns `null` if the collection is\nempty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates\nthe next accumulator value.\n * \n * @sample\nsamples.collections.Collections.Aggregates.reduceOrNull\n * \n * @SinceKotlin("1.4")\n * @WasExperimental(ExperimentalStdlibApi::class)\n\npublic inline fun <S, T : S>\nIterable<T>.reduceOrNull(operation: (acc: S, T) -> S): S? {\n  val iterator = this.iterator()\n  if\n(!iterator.hasNext()) return null\n  var accumulator: S = iterator.next()\n  while (iterator.hasNext()) {\n  accumulator = operation(accumulator, iterator.next())\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value\nstarting with the last element and applying [operation] from right to left\n * to each element and current accumulator\nvalue.\n * \n * Throws an exception if this list is empty. If the list can be empty in an expected way,\n * please use\n[reduceRightOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that\ntakes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample\nsamples.collections.Collections.Aggregates.reduceRight\n * \n * @SinceKotlin("1.4")\n\npublic inline fun <S, T : S>\nList<T>.reduceRight(operation: (T, acc: S) -> S): S {\n  val iterator = listIterator(size)\n  if\n(!iterator.hasPrevious())\n    throw UnsupportedOperationException("\nEmpty list can't be reduced.\n")\n  var\naccumulator: S = iterator.previous()\n  while (iterator.hasPrevious()) {\n    accumulator =\noperation(iterator.previous(), accumulator)\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting\nwith the last element and applying [operation] from right to left\n * to each element with its index in the original list\nand current accumulator value.\n * \n * Throws an exception if this list is empty. If the list can be empty in an\nexpected way,\n * please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * \n * @param [operation] function that takes the index of an element, the element itself and current accumulator\nvalue,\n * and calculates the next accumulator value.\n * \n * @sample\nsamples.collections.Collections.Aggregates.reduceRight\n * \n * @SinceKotlin("1.4")\n\npublic inline fun <S, T : S>\nList<T>.reduceRightIndexed(operation: (index: Int, T, acc: S) -> S): S {\n  val iterator = listIterator(size)\n  if\n(!iterator.hasPrevious())\n    throw UnsupportedOperationException("\nEmpty list can't be reduced.\n")\n  var\naccumulator: S = iterator.previous()\n  while (iterator.hasPrevious()) {\n    val index = iterator.previousIndex()\n    accumulator = operation(index, iterator.previous(), accumulator)\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with\nits index in the original list and current accumulator value.\n * \n * Returns `null` if the list is empty.\n * \n * \n * @param [operation] function that takes the index of an element, the element itself and current accumulator value,\n * and calculates\nthe next accumulator value.\n * \n * @sample\nsamples.collections.Collections.Aggregates.reduceRightOrNull\n * \n * @SinceKotlin("1.4")\n\npublic inline fun <S,\nT : S> List<T>.reduceRightIndexedOrNull(operation: (index: Int, T, acc: S) -> S): S? {\n  val iterator =\nlistIterator(size)\n  if (!iterator.hasPrevious())\n    return null\n  var accumulator: S = iterator.previous()\n  while (iterator.hasPrevious()) {\n    val index = iterator.previousIndex()\n    accumulator = operation(index,\niterator.previous(), accumulator)\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the last\nelement and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * \n *

```

Returns `null` if the list is empty.

`@param [operation]` function that takes an element and current accumulator value, and calculates the next accumulator value.

`@sample`

```
samples.collections.Collections.Aggregates.reduceRightOrNull
```

```
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun <S, T : S>\nList<T>.reduceRightOrNull(operation: (T, acc: S) -> S): S? {\n    val iterator = listIterator(size)\n    if\n    (!iterator.hasPrevious())\n        return null\n    var accumulator: S = iterator.previous()\n    while\n    (iterator.hasPrevious()) {\n        accumulator = operation(iterator.previous(), accumulator)\n    }\n    return\n    accumulator\n}
```

Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with [initial] value.

Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.

`@param [operation]` function that takes current accumulator value and an element, and calculates the next accumulator value.

`@sample`

```
samples.collections.Collections.Aggregates.runningFold
```

```
*\n@SinceKotlin("1.4")\npublic inline fun <T, R>\nIterable<T>.runningFold(initial: R, operation: (acc: R, T) -> R): List<R> {\n    val estimatedSize =\n    collectionSizeOrDefault(9)\n    if (estimatedSize == 0) return listOf(initial)\n    val result =\n    ArrayList<R>(estimatedSize + 1).apply { add(initial) }\n    var accumulator = initial\n    for (element in this) {\n        accumulator = operation(accumulator, element)\n        result.add(accumulator)\n    }\n    return result\n}
```

Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original collection and current accumulator value that starts with [initial] value.

Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.

`@param [operation]` function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.

`@sample`

```
samples.collections.Collections.Aggregates.runningFold
```

```
*\n@SinceKotlin("1.4")\npublic inline fun <T, R>\nIterable<T>.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): List<R> {\n    val estimatedSize =\n    collectionSizeOrDefault(9)\n    if (estimatedSize == 0) return listOf(initial)\n    val result =\n    ArrayList<R>(estimatedSize + 1).apply { add(initial) }\n    var index = 0\n    var accumulator = initial\n    for\n    (element in this) {\n        accumulator = operation(index++, accumulator, element)\n        result.add(accumulator)\n    }\n    return result\n}
```

Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with the first element of this collection.

Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.

`@param [operation]` function that takes current accumulator value and the element, and calculates the next accumulator value.

`@sample`

```
samples.collections.Collections.Aggregates.runningReduce
```

```
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun <S, T : S>\nIterable<T>.runningReduce(operation: (acc: S, T) -> S): List<S> {\n    val iterator = this.iterator()\n    if\n    (!iterator.hasNext()) return emptyList()\n    var accumulator: S = iterator.next()\n    val result =\n    ArrayList<S>(collectionSizeOrDefault(10)).apply { add(accumulator) }\n    while (iterator.hasNext()) {\n        accumulator = operation(accumulator, iterator.next())\n        result.add(accumulator)\n    }\n    return\n    result\n}
```

Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original collection and current accumulator value that starts with the first element of this collection.

Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.

`@param [operation]` function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.

`@sample`

```
samples.collections.Collections.Aggregates.runningReduce
```

```
*\n@SinceKotlin("1.4")\npublic inline fun <S, T : S>\nIterable<T>.runningReduceIndexed(operation: (index: Int,\nacc: S, T) -> S): List<S> {\n    val iterator = this.iterator()\n    if (!iterator.hasNext()) return emptyList()\n    var\n    accumulator: S = iterator.next()\n    val result = ArrayList<S>(collectionSizeOrDefault(10)).apply {\n        add(accumulator) }\n    var index = 1\n    while (iterator.hasNext()) {\n        accumulator = operation(index++,
```

```

accumulator, iterator.next())\n    result.add(accumulator)\n  }\n  return result\n}\n\n/**\n * Returns a list
containing successive accumulation values generated by applying [operation] from left to right\n * to each element
and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation]
function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param
[operation] function that takes current accumulator value and an element, and calculates the next accumulator
value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun <T, R>
Iterable<T>.scan(initial: R, operation: (acc: R, T) -> R): List<R> {\n  return runningFold(initial,
operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation]
from left to right\n * to each element, its index in the original collection and current accumulator value that starts
with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n *
otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the
index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator
value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun <T, R>
Iterable<T>.scanIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): List<R> {\n  return
runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the collection.\n *\n@Deprecated("Use sumOf instead.")\nReplaceWith("this.sumOf(selector)")\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic inline fun <T>
Iterable<T>.sumBy(selector: (T) -> Int): Int {\n  var sum: Int = 0\n  for (element in this) {\n    sum +=
selector(element)\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the collection.\n *\n@Deprecated("Use sumOf instead.")\nReplaceWith("this.sumOf(selector)")\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic inline fun <T>
Iterable<T>.sumByDouble(selector: (T) -> Double): Double {\n  var sum: Double = 0.0\n  for (element in this)
{\n    sum += selector(element)\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the collection.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfDouble")\n@kotlin.internal.InlineOnly\npublic inline fun
<T> Iterable<T>.sumOf(selector: (T) -> Double): Double {\n  var sum: Double = 0.toDouble()\n  for (element in
this) {\n    sum += selector(element)\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the collection.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfInt")\n@kotlin.internal.InlineOnly\npublic inline fun <T>
Iterable<T>.sumOf(selector: (T) -> Int): Int {\n  var sum: Int = 0.toInt()\n  for (element in this) {\n    sum +=
selector(element)\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the collection.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfLong")\n@kotlin.internal.InlineOnly\npublic inline fun
<T> Iterable<T>.sumOf(selector: (T) -> Long): Long {\n  var sum: Long = 0.toLong()\n  for (element in this) {\n
sum += selector(element)\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector]
function applied to each element in the collection.\n
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfUInt")\n@WasExperimental(ExperimentalUnsignedType
s::class)\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.sumOf(selector: (T) -> UInt): UInt {\n
var sum: UInt = 0.toUInt()\n  for (element in this) {\n    sum += selector(element)\n  }\n  return
sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the
collection.\n
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution

```

```

ByLambdaReturnType\n@kotlin.jvm.JvmName("\sumOfULong")\n@WasExperimental(ExperimentalUnsignedTy
pes::class)\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.sumOf(selector: (T) -> ULong): ULong
{\n    var sum: ULong = 0.toULong()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return
sum}\n\n/**\n * Returns an original collection containing all the non-`null` elements, throwing an
[IllegalArgumentException] if there are any `null` elements.\n */\npublic fun <T : Any>
Iterable<T?>.requireNoNulls(): Iterable<T> {\n    for (element in this) {\n        if (element == null) {\n            throw
IllegalArgumentException("\null element found in $this.")\n        }\n    }\n}
\n\n@Suppress("\UNCHECKED_CAST")\n    return this as Iterable<T>\n}\n\n/**\n * Returns an original collection
containing all the non-`null` elements, throwing an [IllegalArgumentException] if there are any `null` elements.\n
*/\npublic fun <T : Any> List<T?>.requireNoNulls(): List<T> {\n    for (element in this) {\n        if (element ==
null) {\n            throw IllegalArgumentException("\null element found in $this.")\n        }\n    }\n}
\n\n@Suppress("\UNCHECKED_CAST")\n    return this as List<T>\n}\n\n/**\n * Splits this collection into a list of
lists each not exceeding the given [size].\n * \n * The last list in the resulting list may have fewer elements than the
given [size].\n * \n * @param size the number of elements to take in each list, must be positive and can be greater
than the number of elements in this collection.\n * \n * @sample
samples.collections.Collections.Transformations.chunked\n */\n@SinceKotlin("1.2")\npublic fun <T>
Iterable<T>.chunked(size: Int): List<List<T>> {\n    return windowed(size, size, partialWindows = true)\n}\n\n/**\n
* Splits this collection into several lists each not exceeding the given [size]\n * and applies the given [transform]
function to an each.\n * \n * @return list of results of the [transform] applied to an each list.\n * \n * Note that the
list passed to the [transform] function is ephemeral and is valid only inside that function.\n * You should not store it
or allow it to escape in some way, unless you made a snapshot of it.\n * The last list may have fewer elements than
the given [size].\n * \n * @param size the number of elements to take in each list, must be positive and can be
greater than the number of elements in this collection.\n * \n * @sample samples.text.Strings.chunkedTransform\n
*/\n@SinceKotlin("1.2")\npublic fun <T, R> Iterable<T>.chunked(size: Int, transform: (List<T>) -> R): List<R>
{\n    return windowed(size, size, partialWindows = true, transform = transform)\n}\n\n/**\n * Returns a list
containing all elements of the original collection without the first occurrence of the given [element].\n */\npublic
operator fun <T> Iterable<T>.minus(element: T): List<T> {\n    val result =
ArrayList<T>(collectionSizeOrDefault(10))\n    var removed = false\n    return this.filterTo(result) { if (!removed
&& it == element) { removed = true; false } else true }\n}\n\n/**\n * Returns a list containing all elements of the
original collection except the elements contained in the given [elements] array.\n * \n * Before Kotlin 1.6, the
[elements] array may have been converted to a [HashSet] to speed up the operation, thus the elements were required
to have\n * a correct and stable implementation of `hashCode()` that didn't change between successive
invocations.\n * \n * On JVM, you can enable this behavior back with the system property
`kotlin.collections.convert_arg_to_set_in_removeAll` set to `true`.\n */\npublic operator fun <T>
Iterable<T>.minus(elements: Array<out T>): List<T> {\n    if (elements.isEmpty()) return this.toList()\n    val other
= elements.convertToSetForSetOperation()\n    return this.filterNot { it in other }\n}\n\n/**\n * Returns a list
containing all elements of the original collection except the elements contained in the given [elements] collection.\n
*\n * \n * Before Kotlin 1.6, the [elements] collection may have been converted to a [HashSet] to speed up the
operation, thus the elements were required to have\n * a correct and stable implementation of `hashCode()` that
didn't change between successive invocations.\n * \n * On JVM, you can enable this behavior back with the system
property `kotlin.collections.convert_arg_to_set_in_removeAll` set to `true`.\n */\npublic operator fun <T>
Iterable<T>.minus(elements: Iterable<T>): List<T> {\n    val other =
elements.convertToSetForSetOperationWith(this)\n    if (other.isEmpty())\n        return this.toList()\n    return
this.filterNot { it in other }\n}\n\n/**\n * Returns a list containing all elements of the original collection except the
elements contained in the given [elements] sequence.\n * \n * Before Kotlin 1.6, the [elements] sequence may have
been converted to a [HashSet] to speed up the operation, thus the elements were required to have\n * a correct and
stable implementation of `hashCode()` that didn't change between successive invocations.\n * \n * On JVM, you can
enable this behavior back with the system property `kotlin.collections.convert_arg_to_set_in_removeAll` set to

```

```

`true`.n */npublic operator fun <T> Iterable<T>.minus(elements: Sequence<T>): List<T> {n    val other =
elements.convertToSetForSetOperation()\n    if (other.isEmpty())\n        return this.toList()\n    return this.filterNot {
it in other } }\n\n/**\n * Returns a list containing all elements of the original collection without the first occurrence
of the given [element].\n */n@kotlin.internal.InlineOnly\npublic inline fun <T>
Iterable<T>.minusElement(element: T): List<T> {n    return minus(element)\n}\n\n/**\n * Splits the original
collection into pair of lists,\n * where *first* list contains elements for which [predicate] yielded `true`,\n * while
*second* list contains elements for which [predicate] yielded `false`. \n * \n * @sample
samples.collections.Iterables.Operations.partition\n */npublic inline fun <T> Iterable<T>.partition(predicate: (T) ->
Boolean): Pair<List<T>, List<T>> {n    val first = ArrayList<T>()\n    val second = ArrayList<T>()\n    for
(element in this) {\n        if (predicate(element)) {\n            first.add(element)\n        } else {\n
second.add(element)\n        }\n    }\n    return Pair(first, second)\n}\n\n/**\n * Returns a list containing all elements
of the original collection and then the given [element].\n */npublic operator fun <T> Iterable<T>.plus(element: T):
List<T> {n    if (this is Collection) return this.plus(element)\n    val result = ArrayList<T>()\n
result.addAll(this)\n    result.add(element)\n    return result\n}\n\n/**\n * Returns a list containing all elements of the
original collection and then the given [element].\n */npublic operator fun <T> Collection<T>.plus(element: T):
List<T> {n    val result = ArrayList<T>(size + 1)\n    result.addAll(this)\n    result.add(element)\n    return
result\n}\n\n/**\n * Returns a list containing all elements of the original collection and then all elements of the given
[elements] array.\n */npublic operator fun <T> Iterable<T>.plus(elements: Array<out T>): List<T> {n    if (this is
Collection) return this.plus(elements)\n    val result = ArrayList<T>()\n    result.addAll(this)\n
result.addAll(elements)\n    return result\n}\n\n/**\n * Returns a list containing all elements of the original
collection and then all elements of the given [elements] array.\n */npublic operator fun <T>
Collection<T>.plus(elements: Array<out T>): List<T> {n    val result = ArrayList<T>(this.size + elements.size)\n
result.addAll(this)\n    result.addAll(elements)\n    return result\n}\n\n/**\n * Returns a list containing all elements
of the original collection and then all elements of the given [elements] collection.\n */npublic operator fun <T>
Iterable<T>.plus(elements: Iterable<T>): List<T> {n    if (this is Collection) return this.plus(elements)\n    val
result = ArrayList<T>()\n    result.addAll(this)\n    result.addAll(elements)\n    return result\n}\n\n/**\n * Returns a
list containing all elements of the original collection and then all elements of the given [elements] collection.\n
*/npublic operator fun <T> Collection<T>.plus(elements: Iterable<T>): List<T> {n    if (elements is Collection)
{\n        val result = ArrayList<T>(this.size + elements.size)\n        result.addAll(this)\n
result.addAll(elements)\n        return result\n    } else {\n        val result = ArrayList<T>(this)\n
result.addAll(elements)\n        return result\n    }\n}\n\n/**\n * Returns a list containing all elements of the original
collection and then all elements of the given [elements] sequence.\n */npublic operator fun <T>
Iterable<T>.plus(elements: Sequence<T>): List<T> {n    val result = ArrayList<T>()\n    result.addAll(this)\n
result.addAll(elements)\n    return result\n}\n\n/**\n * Returns a list containing all elements of the original
collection and then all elements of the given [elements] sequence.\n */npublic operator fun <T>
Collection<T>.plus(elements: Sequence<T>): List<T> {n    val result = ArrayList<T>(this.size + 10)\n
result.addAll(this)\n    result.addAll(elements)\n    return result\n}\n\n/**\n * Returns a list containing all elements
of the original collection and then the given [element].\n */n@kotlin.internal.InlineOnly\npublic inline fun <T>
Iterable<T>.plusElement(element: T): List<T> {n    return plus(element)\n}\n\n/**\n * Returns a list containing all
elements of the original collection and then the given [element].\n */n@kotlin.internal.InlineOnly\npublic inline fun
<T> Collection<T>.plusElement(element: T): List<T> {n    return plus(element)\n}\n\n/**\n * Returns a list of
snapshots of the window of the given [size]\n * sliding along this collection with the given [step], where each\n *
snapshot is a list.\n * \n * Several last lists may have fewer elements than the given [size].\n * \n * Both [size] and
[step] must be positive and can be greater than the number of elements in this collection.\n * \n * @param size the
number of elements to take in each window\n * @param step the number of elements to move the window forward
by on an each step, by default 1\n * @param partialWindows controls whether or not to keep partial windows in the
end if any,\n * by default `false` which means partial windows won't be preserved\n * \n * @sample
samples.collections.Sequences.Transformations.takeWindows\n */n@SinceKotlin("1.2")\npublic fun <T>

```

```

Iterable<T>.windowed(size: Int, step: Int = 1, partialWindows: Boolean = false): List<List<T>> {\n
checkWindowSizeStep(size, step)\n  if (this is RandomAccess && this is List) {\n    val thisSize = this.size\n    val resultCapacity = thisSize / step + if (thisSize % step == 0) 0 else 1\n    val result =\n    ArrayList<List<T>>(resultCapacity)\n    var index = 0\n    while (index in 0 until thisSize) {\n      val\n      windowSize = size.coerceAtMost(thisSize - index)\n      if (windowSize < size && !partialWindows) break\n      result.add(List(windowSize) { this[it + index] })\n      index += step\n    }\n    return result\n  }\n  val\n  result = ArrayList<List<T>>()\n  windowedIterator(iterator(), size, step, partialWindows, reuseBuffer =\n  false).forEach {\n    result.add(it)\n  }\n  return result\n}\n\n/n/**\n * Returns a list of results of applying the\n * given [transform] function to\n * an each list representing a view over the window of the given [size]\n * sliding\n * along this collection with the given [step].\n * \n * Note that the list passed to the [transform] function is ephemeral\n * and is valid only inside that function.\n * You should not store it or allow it to escape in some way, unless you made\n * a snapshot of it.\n * Several last lists may have fewer elements than the given [size].\n * \n * Both [size] and [step]\n * must be positive and can be greater than the number of elements in this collection.\n * @param size the number of\n * elements to take in each window\n * @param step the number of elements to move the window forward by on an\n * each step, by default 1\n * @param partialWindows controls whether or not to keep partial windows in the end if\n * any,\n * by default `false` which means partial windows won't be preserved\n * \n * @sample\n * samples.collections.Sequences.Transformations.averageWindows\n */\n@SinceKotlin("1.2")\npublic fun <T, R>\nIterable<T>.windowed(size: Int, step: Int = 1, partialWindows: Boolean = false, transform: (List<T>) -> R):\nList<R> {\n  checkWindowSizeStep(size, step)\n  if (this is RandomAccess && this is List) {\n    val thisSize =\n    this.size\n    val resultCapacity = thisSize / step + if (thisSize % step == 0) 0 else 1\n    val result =\n    ArrayList<R>(resultCapacity)\n    val window = MovingSubList(this)\n    var index = 0\n    while (index in 0\n    until thisSize) {\n      val windowSize = size.coerceAtMost(thisSize - index)\n      if (!partialWindows &&\n      windowSize < size) break\n      window.move(index, index + windowSize)\n      result.add(transform(window))\n      index += step\n    }\n    return result\n  }\n  val result =\n  ArrayList<R>()\n  windowedIterator(iterator(), size, step, partialWindows, reuseBuffer = true).forEach {\n    result.add(transform(it))\n  }\n  return result\n}\n\n/n/**\n * Returns a list of pairs built from the elements of `this`\n * collection and the [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample\n * samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun <T, R> Iterable<T>.zip(other:\nArray<out R>): List<Pair<T, R>> {\n  return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/n/**\n * Returns a list of values\n * built from the elements of `this` collection and the [other] array with the same index\n * using the provided\n * [transform] function applied to each pair of elements.\n * The returned list has length of the shortest collection.\n * \n * @sample\n * samples.collections.Iterables.Operations.zipIterableWithTransform\n */\npublic inline fun <T, R, V>\nIterable<T>.zip(other: Array<out R>, transform: (a: T, b: R) -> V): List<V> {\n  val arraySize = other.size\n  val\n  list = ArrayList<V>(minOf(collectionSizeOrDefault(10), arraySize))\n  var i = 0\n  for (element in this) {\n    if\n    (i >= arraySize) break\n    list.add(transform(element, other[i++]))\n  }\n  return list\n}\n\n/n/**\n * Returns a list\n * of pairs built from the elements of `this` collection and [other] collection with the same index.\n * The returned list\n * has length of the shortest collection.\n * \n * @sample\n * samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun <T, R> Iterable<T>.zip(other: Iterable<R>): List<Pair<T, R>> {\n  return zip(other) { t1, t2 ->\nt1 to t2 }\n}\n\n/n/**\n * Returns a list of values built from the elements of `this` collection and the [other] collection\n * with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned\n * list has length of the shortest collection.\n * \n * @sample\n * samples.collections.Iterables.Operations.zipIterableWithTransform\n */\npublic inline fun <T, R, V>\nIterable<T>.zip(other: Iterable<R>, transform: (a: T, b: R) -> V): List<V> {\n  val first = iterator()\n  val second\n  = other.iterator()\n  val list = ArrayList<V>(minOf(collectionSizeOrDefault(10),\n  other.collectionSizeOrDefault(10)))\n  while (first.hasNext() && second.hasNext()) {\n    list.add(transform(first.next(), second.next()))\n  }\n  return list\n}\n\n/n/**\n * Returns a list of pairs of each two\n * adjacent elements in this collection.\n * The returned list is empty if this collection contains less than two\n * elements.\n * \n * @sample\n * samples.collections.Collections.Transformations.zipWithNext\n */

```

```

*  

@SinceKotlin("1.2")  

public fun <T> Iterable<T>.zipWithNext(): List<Pair<T, T>> {  

    return zipWithNext { a, b -> a to b }  

}  

  

* Returns a list containing the results of applying the given [transform] function  

* to an each pair of two adjacent elements in this collection.  

* The returned list is empty if this collection contains less than two elements.  

* @sample  

samples.collections.Collections.Transformations.zipWithNextToFindDeltas  

*  

@SinceKotlin("1.2")  

public inline fun <T, R> Iterable<T>.zipWithNext(transform: (a: T, b: T) -> R): List<R> {  

    val iterator = iterator()  

    if (!iterator.hasNext()) return emptyList()  

    val result = mutableListOf<R>()  

    var current = iterator.next()  

    while (iterator.hasNext()) {  

        val next = iterator.next()  

        result.add(transform(current, next))  

        current = next  

    }  

    return result  

}  

  

* Appends the string from all the elements separated using [separator] and  

* using the given [prefix] and [postfix] if supplied.  

* If the collection could be huge, you can specify a non-negative value of [limit],  

* in which case only the first [limit] elements will be appended, followed by the  

* [truncated] string (which defaults to "...").  

* @sample  

samples.collections.Collections.Transformations.joinTo  

*  

public fun <T, A : Appendable> Iterable<T>.joinTo(buffer: A, separator: CharSequence = ",",  

prefix: CharSequence = "\", postfix: CharSequence = "\"", limit: Int = -1,  

truncated: CharSequence = "...", transform: ((T) -> CharSequence)? = null): A {  

    buffer.append(prefix)  

    var count = 0  

    for (element in this) {  

        if (++count > 1) buffer.append(separator)  

        if (limit < 0 || count <= limit) {  

            buffer.appendElement(element, transform)  

        } else break  

    }  

    if (limit >= 0 && count > limit) buffer.append(truncated)  

    buffer.append(postfix)  

    return buffer  

}  

  

* Creates a string from all the elements separated using [separator] and using the  

* given [prefix] and [postfix] if supplied.  

* If the collection could be huge, you can specify a non-negative value of [limit],  

* in which case only the first [limit] elements will be appended, followed by the  

* [truncated] string (which defaults to "...").  

* @sample  

samples.collections.Collections.Transformations.joinToString  

*  

public fun <T> Iterable<T>.joinToString(separator: CharSequence = ",", prefix: CharSequence = "\",  

postfix: CharSequence = "\"", limit: Int = -1, truncated: CharSequence = "...",  

transform: ((T) -> CharSequence)? = null): String {  

    return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()  

}  

  

* Returns this collection as an [Iterable].  

*  

@kotlin.internal.InlineOnly  

public inline fun <T> Iterable<T>.asIterable(): Iterable<T> {  

    return this  

}  

  

* Creates a [Sequence] instance that wraps the original collection returning  

* its elements when being iterated.  

* @sample  

samples.collections.Sequences.Building.sequenceFromCollection  

*  

public fun <T> Iterable<T>.asSequence(): Sequence<T> {  

    return Sequence { this.iterator() }  

}  

  

* Returns an average value of elements in the collection.  

*  

@kotlin.jvm.JvmName("averageOfByte")  

public fun Iterable<Byte>.average(): Double {  

    var sum: Double = 0.0  

    var count: Int = 0  

    for (element in this) {  

        sum += element  

        checkCountOverflow(++count)  

    }  

    return if (count == 0) Double.NaN else sum / count  

}  

  

* Returns an average value of elements in the collection.  

*  

@kotlin.jvm.JvmName("averageOfShort")  

public fun Iterable<Short>.average(): Double {  

    var sum: Double = 0.0  

    var count: Int = 0  

    for (element in this) {  

        sum += element  

        checkCountOverflow(++count)  

    }  

    return if (count == 0) Double.NaN else sum / count  

}  

  

* Returns an average value of elements in the collection.  

*  

@kotlin.jvm.JvmName("averageOfInt")  

public fun Iterable<Int>.average(): Double {  

    var sum: Double = 0.0  

    var count: Int = 0  

    for (element in this) {  

        sum += element  

        checkCountOverflow(++count)  

    }  

    return if (count == 0) Double.NaN else sum / count  

}  

  

* Returns an average value of elements in the collection.  

*  

@kotlin.jvm.JvmName("averageOfLong")  

public fun Iterable<Long>.average(): Double {  

    var sum: Double = 0.0  

    var count: Int = 0  

    for (element in this) {  

        sum += element  

        checkCountOverflow(++count)  

    }  

    return if (count == 0) Double.NaN else sum / count  

}  

  

* Returns an average value of elements in the collection.  

*  

@kotlin.jvm.JvmName("averageOfFloat")  

public fun Iterable<Float>.average(): Double {  

    var sum: Double = 0.0  

    var count: Int = 0  

    for (element in this) {  

        sum += element  

        checkCountOverflow(++count)  

    }  

    return if (count == 0) Double.NaN else sum / count  

}  

  

* Returns an average value of elements in the collection.  

*  

@kotlin.jvm.JvmName("averageOfDouble")  

public fun Iterable<Double>.average(): Double {  

    var sum:

```

```

Double = 0.0\n  var count: Int = 0\n  for (element in this) {\n    sum += element\n  }\n  checkCountOverflow(++count)\n  }\n  return if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns the sum of all elements in the collection.\n */\n@kotlin.jvm.JvmName("sumOfByte")\npublic fun Iterable<Byte>.sum(): Int {\n  var sum: Int = 0\n  for (element in this) {\n    sum += element\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all elements in the collection.\n */\n@kotlin.jvm.JvmName("sumOfShort")\npublic fun Iterable<Short>.sum(): Int {\n  var sum: Int = 0\n  for (element in this) {\n    sum += element\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all elements in the collection.\n */\n@kotlin.jvm.JvmName("sumOfInt")\npublic fun Iterable<Int>.sum(): Int {\n  var sum: Int = 0\n  for (element in this) {\n    sum += element\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all elements in the collection.\n */\n@kotlin.jvm.JvmName("sumOfLong")\npublic fun Iterable<Long>.sum(): Long {\n  var sum: Long = 0L\n  for (element in this) {\n    sum += element\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all elements in the collection.\n */\n@kotlin.jvm.JvmName("sumOfFloat")\npublic fun Iterable<Float>.sum(): Float {\n  var sum: Float = 0.0f\n  for (element in this) {\n    sum += element\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all elements in the collection.\n */\n@kotlin.jvm.JvmName("sumOfDouble")\npublic fun Iterable<Double>.sum(): Double {\n  var sum: Double = 0.0\n  for (element in this) {\n    sum += element\n  }\n  return sum\n}\n\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.collections\n\nimport kotlin.comparisons.naturalOrder\nimport kotlin.random.Random\n\n/**\n * Returns the array if it's not `null`, or an empty array otherwise.\n */\n@sample samples.collections.Arrays.Usage.arrayOrEmpty\n\n@kotlin.internal.InlineOnly\npublic actual inline fun <T> Array<out T>?.orEmpty(): Array<out T> = this ?: emptyArray<T>()\n\n/**\n * Returns a *typed* array containing all of the elements of this collection.\n */\n@kotlin.internal.InlineOnly\npublic actual inline fun <T> Collection<T>.toArray(): Array<T> = copyToArray(this)\n\n@JsName("copyToArray")\n@PublishedApi\ninternal fun <T> copyToArray(collection: Collection<T>): Array<T> {\n  return if (collection.asDynamic().toArray !== undefined)\n    collection.asDynamic().toArray().unsafeCast<Array<T>>()\n  else\n    copyToArrayImpl(collection).unsafeCast<Array<T>>()\n}\n\n@JsName("copyToArrayImpl")\ninternal actual fun copyToArrayImpl(collection: Collection<*>): Array<Any?> {\n  val array = emptyArray<Any?>()\n  val iterator = collection.iterator()\n  while (iterator.hasNext())\n    array.asDynamic().push(iterator.next())\n  return array\n}\n\n@JsName("copyToExistingArrayImpl")\ninternal actual fun <T> copyToArrayImpl(collection: Collection<*>, array: Array<T>): Array<T> {\n  if (array.size < collection.size)\n    return copyToArrayImpl(collection).unsafeCast<Array<T>>()\n  val iterator = collection.iterator()\n  var index = 0\n  while (iterator.hasNext()) {\n    array[index++] = iterator.next().unsafeCast<T>()\n  }\n  if (index < array.size)\n    array[index] = null.unsafeCast<T>()\n  return array\n}\n\n/**\n * Returns an immutable list containing only the specified object [element].\n */\n@kotlin.jvm.JvmName("listOf")\npublic fun <T> listOf(element: T): List<T> = listOfOf(element)\n\n@PublishedApi\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\ninternal actual inline fun <E> buildListInternal(builderAction: MutableList<E>.() -> Unit): List<E> {\n  return ArrayList<E>().apply(builderAction).build()\n}\n\n@PublishedApi\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\ninternal actual inline fun <E> buildListInternal(capacity: Int, builderAction: MutableList<E>.() -> Unit): List<E> {\n  checkBuilderCapacity(capacity)\n  return ArrayList<E>(capacity).apply(builderAction).build()\n}\n\n/**\n * Returns an immutable set containing only the specified object [element].\n */\n@kotlin.jvm.JvmName("setOf")\npublic fun <T> setOf(element: T): Set<T> = hashSetOf(element)\n\n@PublishedApi\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\ninternal actual inline fun <E> buildSetInternal(builderAction: MutableSet<E>.() -> Unit): Set<E> {\n  return LinkedHashSet<E>().apply(builderAction).build()\n}\n\n@PublishedApi\n@SinceKotlin("1.3")\n@kotlin.internal.

```

```

InlineOnly\ninternal actual inline fun <E> buildSetInternal(capacity: Int, builderAction: MutableSet<E>.<E>() -> Unit):
Set<E> {\n    return LinkedHashSet<E>(capacity).apply(builderAction).build()\n}\n\n/**\n * Returns an
immutable map, mapping only the specified key to the\n * specified value.\n */\npublic fun <K, V> mapOf(pair:
Pair<K, V>): Map<K, V> =
hashMapOf(pair)\n\n@PublishedApi\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\ninternal actual inline
fun <K, V> buildMapInternal(builderAction: MutableMap<K, V>.<V>() -> Unit): Map<K, V> {\n    return
LinkedHashMap<K,
V>().apply(builderAction).build()\n}\n\n@PublishedApi\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\ninternal actual inline fun <K, V> buildMapInternal(capacity: Int, builderAction: MutableMap<K, V>.<V>() -> Unit):
Map<K, V> {\n    return LinkedHashMap<K, V>(capacity).apply(builderAction).build()\n}\n\n/**\n * Fills the
list with the provided [value].\n */\n * Each element in the list gets replaced with the [value].\n
*\n\n@SinceKotlin("1.2")\npublic actual fun <T> MutableList<T>.fill(value: T): Unit {\n    for (index in
0..lastIndex) {\n        this[index] = value\n    }\n}\n\n/**\n * Randomly shuffles elements in this list.\n */\n * See:
https://en.wikipedia.org/wiki/Fisher%20%80%93Yates\_shuffle#The\_modern\_algorithm\n
*\n\n@SinceKotlin("1.2")\npublic actual fun <T> MutableList<T>.shuffle(): Unit = shuffle(Random)\n\n/**\n *
Returns a new list with the elements of this list randomly shuffled.\n */\n\n@SinceKotlin("1.2")\npublic actual fun
<T> Iterable<T>.shuffled(): List<T> = toMutableList().apply { shuffle() }\n\n/**\n * Sorts elements in the list
in-place according to their natural sort order.\n */\n * The sort is _stable_. It means that equal elements preserve their
order relative to each other after sorting.\n */\n * @sample samples.collections.Collections.Sorting.sortMutableList\n
*\n\npublic actual fun <T : Comparable<T>> MutableList<T>.sort(): Unit {\n    Collections.sort(this,
naturalOrder())\n}\n\n/**\n * Sorts elements in the list in-place according to the order specified with [comparator].\n
*/\n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n
*/\n * @sample samples.collections.Collections.Sorting.sortMutableListWith\n
*\n\npublic actual fun <T>
MutableList<T>.sortWith(comparator: Comparator<in T>): Unit {\n    Collections.sort(this,
comparator)\n}\n\nprivate fun <T> Collections.sort(list: MutableList<T>, comparator: Comparator<in T>) {\n    if
(list.size <= 1) return\n\n    val array = copyToArray(list)\n    sortArrayWith(array, comparator)\n\n    for (i in 0 until
array.size) {\n        list[i] = array[i]\n    }\n}\n\ninternal actual fun <T> arrayOfNulls(reference: Array<T>, size: Int):
Array<T> {\n    return
arrayOfNulls<Any>(size).unsafeCast<Array<T>>()\n}\n\n@SinceKotlin("1.3")\n@PublishedApi\n@JsName("arrayCopy")\ninternal fun <T> arrayCopy(source: Array<out T>, destination: Array<in T>, destinationOffset: Int,
startIndex: Int, endIndex: Int) {\n    AbstractList.checkRangeIndexes(startIndex, endIndex, source.size)\n    val
rangeSize = endIndex - startIndex\n    AbstractList.checkRangeIndexes(destinationOffset, destinationOffset +
rangeSize, destination.size)\n\n    if (js("ArrayBuffer").isView(destination) &&
js("ArrayBuffer").isView(source)) {\n        val subrange = source.asDynamic().subarray(startIndex, endIndex)\n
        destination.asDynamic().set(subrange, destinationOffset)\n    } else {\n        if (source !== destination ||
destinationOffset <= startIndex) {\n            for (index in 0 until rangeSize) {\n
                destination[destinationOffset + index] = source[startIndex + index]\n            }\n        } else {\n            for (index in
rangeSize - 1 downTo 0) {\n                destination[destinationOffset + index] = source[startIndex + index]\n
            }\n        }\n    }\n}\n\n// no singleton map implementation in js, return map as
is\n\n@Suppress("NOTHING_TO_INLINE")\ninternal actual inline fun <K, V> Map<K,
V>.toSingletonMapOrSelf(): Map<K, V> = this\n\n@Suppress("NOTHING_TO_INLINE")\ninternal actual inline
fun <K, V> Map<out K, V>.toSingletonMap(): Map<K, V> =
this.toMutableMap()\n\n@Suppress("NOTHING_TO_INLINE")\ninternal actual inline fun <T> Array<out
T>.copyToArrayOfAny(isVarargs: Boolean): Array<out Any?> =\n    if (isVarargs)\n        // no need to copy vararg
array in JS\n        this\n    else\n        this.copyOf()\n\n\n@PublishedApi\ninternal actual fun
checkIndexOverflow(index: Int): Int {\n    if (index < 0) {\n        throwIndexOverflow()\n    }\n    return
index\n}\n\n@PublishedApi\ninternal actual fun checkCountOverflow(count: Int): Int {\n    if (count < 0) {\n
        throwCountOverflow()\n    }\n    return count\n}\n\n\n/**\n * JS map and set implementations do not make use of

```


Double, b: Double, c: Double): Double\n\n/**\n * Returns the greater of three values according to the order specified by the given [comparator].\n * \n * If there are multiple equal maximal values, returns the first of them.\n * \n @SinceKotlin("1.1")\n public fun <T> maxOf(a: T, b: T, c: T, comparator: Comparator<in T>): T {\n return maxOf(a, maxOf(b, c, comparator), comparator)\n }\n\n/**\n * Returns the greater of two values according to the order specified by the given [comparator].\n * \n * If values are equal, returns the first one.\n * \n @SinceKotlin("1.1")\n public fun <T> maxOf(a: T, b: T, comparator: Comparator<in T>): T {\n return if (comparator.compare(a, b) >= 0) a else b\n }\n\n/**\n * Returns the greater of the given values.\n * \n * If there are multiple equal maximal values, returns the first of them.\n * \n @SinceKotlin("1.4")\n public expect fun <T : Comparable<T>> maxOf(a: T, vararg other: T): T\n\n/**\n * Returns the greater of the given values.\n * \n @SinceKotlin("1.4")\n public expect fun maxOf(a: Byte, vararg other: Byte): Byte\n\n/**\n * Returns the greater of the given values.\n * \n @SinceKotlin("1.4")\n public expect fun maxOf(a: Short, vararg other: Short): Short\n\n/**\n * Returns the greater of the given values.\n * \n @SinceKotlin("1.4")\n public expect fun maxOf(a: Int, vararg other: Int): Int\n\n/**\n * Returns the greater of the given values.\n * \n @SinceKotlin("1.4")\n public expect fun maxOf(a: Long, vararg other: Long): Long\n\n/**\n * Returns the greater of the given values.\n * \n * If any value is `NaN`, returns `NaN`.\n * \n @SinceKotlin("1.4")\n public expect fun maxOf(a: Float, vararg other: Float): Float\n\n/**\n * Returns the greater of the given values.\n * \n * If any value is `NaN`, returns `NaN`.\n * \n @SinceKotlin("1.4")\n public expect fun maxOf(a: Double, vararg other: Double): Double\n\n/**\n * Returns the greater of the given values according to the order specified by the given [comparator].\n * \n * If there are multiple equal maximal values, returns the first of them.\n * \n @SinceKotlin("1.4")\n public fun <T> maxOf(a: T, vararg other: T, comparator: Comparator<in T>): T {\n var max = a\n for (e in other) if (comparator.compare(max, e) < 0) max = e\n return max\n }\n\n/**\n * Returns the smaller of two values.\n * \n * If values are equal, returns the first one.\n * \n @SinceKotlin("1.1")\n public expect fun <T : Comparable<T>> minOf(a: T, b: T): T\n\n/**\n * Returns the smaller of two values.\n * \n @SinceKotlin("1.1")\n @kotlin.internal.InlineOnly\n public expect inline fun minOf(a: Byte, b: Byte): Byte\n\n/**\n * Returns the smaller of two values.\n * \n @SinceKotlin("1.1")\n @kotlin.internal.InlineOnly\n public expect inline fun minOf(a: Short, b: Short): Short\n\n/**\n * Returns the smaller of two values.\n * \n @SinceKotlin("1.1")\n @kotlin.internal.InlineOnly\n public expect inline fun minOf(a: Int, b: Int): Int\n\n/**\n * Returns the smaller of two values.\n * \n @SinceKotlin("1.1")\n @kotlin.internal.InlineOnly\n public expect inline fun minOf(a: Long, b: Long): Long\n\n/**\n * Returns the smaller of two values.\n * \n * If either value is `NaN`, returns `NaN`.\n * \n @SinceKotlin("1.1")\n @kotlin.internal.InlineOnly\n public expect inline fun minOf(a: Float, b: Float): Float\n\n/**\n * Returns the smaller of two values.\n * \n * If either value is `NaN`, returns `NaN`.\n * \n @SinceKotlin("1.1")\n @kotlin.internal.InlineOnly\n public expect inline fun minOf(a: Double, b: Double): Double\n\n/**\n * Returns the smaller of three values.\n * \n * If there are multiple equal minimal values, returns the first of them.\n * \n @SinceKotlin("1.1")\n public expect fun <T : Comparable<T>> minOf(a: T, b: T, c: T): T\n\n/**\n * Returns the smaller of three values.\n * \n @SinceKotlin("1.1")\n @kotlin.internal.InlineOnly\n public expect inline fun minOf(a: Byte, b: Byte, c: Byte): Byte\n\n/**\n * Returns the smaller of three values.\n * \n @SinceKotlin("1.1")\n @kotlin.internal.InlineOnly\n public expect inline fun minOf(a: Short, b: Short, c: Short): Short\n\n/**\n * Returns the smaller of three values.\n * \n @SinceKotlin("1.1")\n @kotlin.internal.InlineOnly\n public expect inline fun minOf(a: Int, b: Int, c: Int): Int\n\n/**\n * Returns the smaller of three values.\n * \n * If any value is `NaN`, returns `NaN`.\n * \n @SinceKotlin("1.1")\n @kotlin.internal.InlineOnly\n public expect inline fun minOf(a: Float, b: Float, c: Float): Float\n\n/**\n * Returns the smaller of three values.\n * \n * If any value is `NaN`, returns `NaN`.\n * \n @SinceKotlin("1.1")\n @kotlin.internal.InlineOnly\n public expect inline fun minOf(a: Double, b: Double, c: Double): Double\n\n/**\n * Returns the smaller of three values according to the order specified by the given [comparator].\n * \n * If there are multiple equal minimal values, returns the first of them.\n * \n @SinceKotlin("1.1")\n public fun <T> minOf(a: T, b: T, c: T, comparator: Comparator<in T>): T {\n return

```

minOf(a, minOf(b, c, comparator), comparator)\n\n/**\n * Returns the smaller of two values according to the
order specified by the given [comparator].\n * \n * If values are equal, returns the first one.\n
*\n@SinceKotlin("1.1")\npublic fun <T> minOf(a: T, b: T, comparator: Comparator<in T>): T {\n    return if
(comparator.compare(a, b) <= 0) a else b\n}\n\n/**\n * Returns the smaller of the given values.\n * \n * If there are
multiple equal minimal values, returns the first of them.\n *\n@SinceKotlin("1.4")\npublic expect fun <T :
Comparable<T>> minOf(a: T, vararg other: T): T\n\n/**\n * Returns the smaller of the given values.\n
*\n@SinceKotlin("1.4")\npublic expect fun minOf(a: Byte, vararg other: Byte): Byte\n\n/**\n * Returns the
smaller of the given values.\n *\n@SinceKotlin("1.4")\npublic expect fun minOf(a: Short, vararg other: Short):
Short\n\n/**\n * Returns the smaller of the given values.\n *\n@SinceKotlin("1.4")\npublic expect fun minOf(a:
Int, vararg other: Int): Int\n\n/**\n * Returns the smaller of the given values.\n *\n@SinceKotlin("1.4")\npublic
expect fun minOf(a: Long, vararg other: Long): Long\n\n/**\n * Returns the smaller of the given values.\n * \n *
If any value is `NaN`, returns `NaN`.\n *\n@SinceKotlin("1.4")\npublic expect fun minOf(a: Float, vararg other:
Float): Float\n\n/**\n * Returns the smaller of the given values.\n * \n * If any value is `NaN`, returns `NaN`.\n
*\n@SinceKotlin("1.4")\npublic expect fun minOf(a: Double, vararg other: Double): Double\n\n/**\n * Returns
the smaller of the given values according to the order specified by the given [comparator].\n * \n * If there are
multiple equal minimal values, returns the first of them.\n *\n@SinceKotlin("1.4")\npublic fun <T> minOf(a: T,
vararg other: T, comparator: Comparator<in T>): T {\n    var min = a\n    for (e in other) if
(comparator.compare(min, e) > 0) min = e\n    return min\n}\n\n"/**\n * Copyright 2010-2021 JetBrains s.r.o. and
Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file.\n
*\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("MapsKt")\n\npackage
kotlin.collections\n\n/\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See:
https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\nimport kotlin.random.*\nimport
kotlin.ranges.contains\nimport kotlin.ranges.reversed\n\n/**\n * Returns the first non-null value produced by
[transform] function being applied to entries of this map in iteration order,\n * or throws
[NoSuchElementException] if no non-null value was produced.\n * \n * @sample
samples.collections.Collections.Transformations.firstNotNullOf\n
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun <K, V, R : Any> Map<out K,
V>.firstNotNullOf(transform: (Map.Entry<K, V>) -> R?): R {\n    return firstNotNullOfOrNull(transform) ?: throw
NoSuchElementException("No element of the map was transformed to a non-null value.")\n}\n\n/**\n * Returns
the first non-null value produced by [transform] function being applied to entries of this map in iteration order,\n *
or `null` if no non-null value was produced.\n * \n * @sample
samples.collections.Collections.Transformations.firstNotNullOf\n
*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun <K, V, R : Any> Map<out K,
V>.firstNotNullOfOrNull(transform: (Map.Entry<K, V>) -> R?): R? {\n    for (element in this) {\n        val result =
transform(element)\n        if (result != null) {\n            return result\n        }\n    }\n    return null\n}\n\n/**\n *
Returns a [List] containing all key-value pairs.\n *\npublic fun <K, V> Map<out K, V>.toList(): List<Pair<K, V>>
{\n    if (size == 0)\n        return emptyList()\n    val iterator = entries.iterator()\n    if (!iterator.hasNext())\n        return emptyList()\n    val first = iterator.next()\n    if (!iterator.hasNext())\n        return listOf(first.toPair())\n    val
result = ArrayList<Pair<K, V>>(size)\n    result.add(first.toPair())\n    do {\n        result.add(iterator.next().toPair())\n    } while (iterator.hasNext())\n    return result\n}\n\n/**\n * Returns a single list
of all elements yielded from results of [transform] function being invoked on each entry of original map.\n * \n *
@sample samples.collections.Maps.Transformations.flatMap\n *\npublic inline fun <K, V, R> Map<out K,
V>.flatMap(transform: (Map.Entry<K, V>) -> Iterable<R>): List<R> {\n    return flatMapTo(ArrayList<R>(),
transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being
invoked on each entry of original map.\n * \n * @sample samples.collections.Collections.Transformations.flatMap\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapSequence")\npublic inline fun <K, V, R> Map<out K,

```

```

V>.flatMap(transform: (Map.Entry<K, V>) -> Sequence<R>): List<R> {\n  return flatMapTo(ArrayList<R>(),
transform)\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each
entry of original map, to the given [destination].\n */\npublic inline fun <K, V, R, C : MutableCollection<in R>>
Map<out K, V>.flatMapTo(destination: C, transform: (Map.Entry<K, V>) -> Iterable<R>): C {\n  for (element in
this) {\n    val list = transform(element)\n    destination.addAll(list)\n  }\n  return destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each entry of original map, to
the given [destination].\n */\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("\flatMapSequenceTo")\npublic inline fun <K, V, R, C :
MutableCollection<in R>> Map<out K, V>.flatMapTo(destination: C, transform: (Map.Entry<K, V>) ->
Sequence<R>): C {\n  for (element in this) {\n    val list = transform(element)\n    destination.addAll(list)\n
}\n  return destination\n}\n\n/**\n * Returns a list containing the results of applying the given [transform]
function\n * to each entry in the original map.\n */\n * @sample
samples.collections.Maps.Transformations.mapToList\n */\npublic inline fun <K, V, R> Map<out K,
V>.map(transform: (Map.Entry<K, V>) -> R): List<R> {\n  return mapTo(ArrayList<R>(size),
transform)\n}\n\n/**\n * Returns a list containing only the non-null results of applying the given [transform]
function\n * to each entry in the original map.\n */\n * @sample
samples.collections.Maps.Transformations.mapNotNull\n */\npublic inline fun <K, V, R : Any> Map<out K,
V>.mapNotNull(transform: (Map.Entry<K, V>) -> R?): List<R> {\n  return mapNotNullTo(ArrayList<R>(),
transform)\n}\n\n/**\n * Applies the given [transform] function to each entry in the original map\n * and appends
only the non-null results to the given [destination].\n */\npublic inline fun <K, V, R : Any, C : MutableCollection<in
R>> Map<out K, V>.mapNotNullTo(destination: C, transform: (Map.Entry<K, V>) -> R?): C {\n  forEach {
element -> transform(element)?.let { destination.add(it) } }\n  return destination\n}\n\n/**\n * Applies the given
[transform] function to each entry of the original map\n * and appends the results to the given [destination].\n */\n\npublic inline fun <K, V, R, C : MutableCollection<in R>> Map<out K, V>.mapTo(destination: C, transform:
(Map.Entry<K, V>) -> R): C {\n  for (item in this)\n    destination.add(transform(item))\n  return
destination\n}\n\n/**\n * Returns `true` if all entries match the given [predicate].\n */\n * @sample
samples.collections.Collections.Aggregates.all\n */\npublic inline fun <K, V> Map<out K, V>.all(predicate:
(Map.Entry<K, V>) -> Boolean): Boolean {\n  if (isEmpty()) return true\n  for (element in this) if
(!predicate(element)) return false\n  return true\n}\n\n/**\n * Returns `true` if map has at least one entry.\n */\n *
@sample samples.collections.Collections.Aggregates.any\n */\npublic fun <K, V> Map<out K, V>.any(): Boolean
{\n  return !isEmpty()\n}\n\n/**\n * Returns `true` if at least one entry matches the given [predicate].\n */\n *
@sample samples.collections.Collections.Aggregates.anyWithPredicate\n */\npublic inline fun <K, V> Map<out K,
V>.any(predicate: (Map.Entry<K, V>) -> Boolean): Boolean {\n  if (isEmpty()) return false\n  for (element in
this) if (predicate(element)) return true\n  return false\n}\n\n/**\n * Returns the number of entries in this map.\n */\n\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<out K, V>.count(): Int {\n  return size\n}\n\n/**\n * Returns the number of entries matching the given [predicate].\n */\npublic inline fun <K, V> Map<out K,
V>.count(predicate: (Map.Entry<K, V>) -> Boolean): Int {\n  if (isEmpty()) return 0\n  var count = 0\n  for
(element in this) if (predicate(element)) ++count\n  return count\n}\n\n/**\n * Performs the given [action] on each
entry.\n */\n\n@kotlin.internal.HidesMembers\npublic inline fun <K, V> Map<out K, V>.forEach(action:
(Map.Entry<K, V>) -> Unit): Unit {\n  for (element in this) action(element)\n}\n\n@Deprecated("Use
maxByOrNull instead.")\n@kotlin.internal.InlineOnly\n@DeprecatedSinceKotlin(warningSince =
"1.4", errorSince = "1.5", hiddenSince = "1.6")\n@kotlin.internal.InlineOnly\npublic inline fun <K, V, R :
Comparable<R>> Map<out K, V>.maxBy(selector: (Map.Entry<K, V>) -> R): Map.Entry<K, V>? {\n  return
maxByOrNull(selector)\n}\n\n/**\n * Returns the first entry yielding the largest value of the given function or `null`
if there are no entries.\n */\n * @sample samples.collections.Collections.Aggregates.maxByOrNull\n */\n\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <K, V, R : Comparable<R>> Map<out
K, V>.maxByOrNull(selector: (Map.Entry<K, V>) -> R): Map.Entry<K, V>? {\n  return

```

`entries.maxByOrNull(selector)` Returns the largest value among all values produced by [selector] function applied to each entry in the map. If any of values produced by [selector] function is `NaN`, the returned result is `NaN`. @throws NoSuchElementException if the map is empty.

```

*\/@SinceKotlin("1.4")\/@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\/@OverloadResolution
ByLambdaReturnType\/@kotlin.internal.InlineOnly\/public inline fun <K, V> Map<out K, V>.maxOf(selector:
(Map.Entry<K, V>) -> Double): Double {
    return entries.maxOf(selector)
}

```

`entries.maxOf(selector)` Returns the largest value among all values produced by [selector] function applied to each entry in the map. If any of values produced by [selector] function is `NaN`, the returned result is `NaN`. @throws NoSuchElementException if the map is empty.

```

*\/@SinceKotlin("1.4")\/@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\/@OverloadResolution
ByLambdaReturnType\/@kotlin.internal.InlineOnly\/public inline fun <K, V> Map<out K, V>.maxOf(selector:
(Map.Entry<K, V>) -> Float): Float {
    return entries.maxOf(selector)
}

```

`entries.maxOf(selector)` Returns the largest value among all values produced by [selector] function applied to each entry in the map or `null` if there are no entries. If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

```

*\/@SinceKotlin("1.4")\/@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\/@OverloadResolution
ByLambdaReturnType\/@kotlin.internal.InlineOnly\/public inline fun <K, V, R : Comparable<R>> Map<out K,
V>.maxOf(selector: (Map.Entry<K, V>) -> R): R {
    return entries.maxOf(selector)
}

```

`entries.maxOfOrNull(selector)` Returns the largest value among all values produced by [selector] function applied to each entry in the map or `null` if there are no entries. If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

```

*\/@SinceKotlin("1.4")\/@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\/@OverloadResolution
ByLambdaReturnType\/@kotlin.internal.InlineOnly\/public inline fun <K, V> Map<out K,
V>.maxOfOrNull(selector: (Map.Entry<K, V>) -> Double): Double? {
    return
entries.maxOfOrNull(selector)
}

```

`entries.maxOfOrNull(selector)` Returns the largest value among all values produced by [selector] function applied to each entry in the map or `null` if there are no entries. If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

```

*\/@SinceKotlin("1.4")\/@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\/@OverloadResolution
ByLambdaReturnType\/@kotlin.internal.InlineOnly\/public inline fun <K, V> Map<out K,
V>.maxOfOrNull(selector: (Map.Entry<K, V>) -> Float): Float? {
    return
entries.maxOfOrNull(selector)
}

```

`entries.maxOfOrNull(selector)` Returns the largest value among all values produced by [selector] function applied to each entry in the map or `null` if there are no entries. Returns the largest value according to the provided [comparator] among all values produced by [selector] function applied to each entry in the map. @throws NoSuchElementException if the map is empty.

```

*\/@SinceKotlin("1.4")\/@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\/@OverloadResolution
ByLambdaReturnType\/@kotlin.internal.InlineOnly\/public inline fun <K, V, R : Comparable<R>> Map<out K,
V>.maxOfOrNull(selector: (Map.Entry<K, V>) -> R): R? {
    return entries.maxOfOrNull(selector)
}

```

`entries.maxOfWith(comparator, selector)` Returns the largest value according to the provided [comparator] among all values produced by [selector] function applied to each entry in the map or `null` if there are no entries.

```

*\/@SinceKotlin("1.4")\/@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\/@OverloadResolution
ByLambdaReturnType\/@kotlin.internal.InlineOnly\/public inline fun <K, V, R> Map<out K,
V>.maxOfWith(comparator: Comparator<in R>, selector: (Map.Entry<K, V>) -> R): R {
    return
entries.maxOfWith(comparator, selector)
}

```

`entries.maxOfWith(comparator, selector)` Returns the largest value according to the provided [comparator] among all values produced by [selector] function applied to each entry in the map or `null` if there are no entries.

```

*\/@SinceKotlin("1.4")\/@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\/@OverloadResolution
ByLambdaReturnType\/@kotlin.internal.InlineOnly\/public inline fun <K, V, R> Map<out K,
V>.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Map.Entry<K, V>) -> R): R? {
    return
entries.maxOfWithOrNull(comparator, selector)
}

```

`entries.maxOfWithOrNull(comparator, selector)` Returns the largest value according to the provided [comparator] among all values produced by [selector] function applied to each entry in the map or `null` if there are no entries. @deprecated Use `maxWithOrNull` instead.

```

ReplaceWith("this.maxWithOrNull(comparator)")
@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5", hiddenSince = "1.6")

```

`entries.maxWith(comparator)` Returns the largest value according to the provided [comparator] among all values produced by [selector] function applied to each entry in the map or `null` if there are no entries.

```

*\/@SinceKotlin("1.4")\/@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\/@OverloadResolution
ByLambdaReturnType\/@kotlin.internal.InlineOnly\/public inline fun <K, V> Map<out K,
V>.maxWith(comparator: Comparator<in Map.Entry<K, V>>): Map.Entry<K, V>? {
    return

```

```

maxWithOrNull(comparator)\n}\n\n/**\n * Returns the first entry having the largest value according to the provided
[comparator] or `null` if there are no entries.\n * \n\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic
inline fun <K, V> Map<out K, V>.maxWithOrNull(comparator: Comparator<in Map.Entry<K, V>>):
Map.Entry<K, V>? {\n    return entries.maxWithOrNull(comparator)\n}\n\n@Deprecated("Use minByOrNull
instead.", ReplaceWith("this.minByOrNull(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.4",
errorSince = "1.5", hiddenSince = "1.6")\npublic inline fun <K, V, R : Comparable<R>> Map<out K,
V>.minBy(selector: (Map.Entry<K, V>) -> R): Map.Entry<K, V>? {\n    return minByOrNull(selector)\n}\n\n/**\n * Returns the first entry yielding the smallest value of the given function or `null` if there are no entries.\n * \n\n *
@sample samples.collections.Collections.Aggregates.minByOrNull\n
*\n\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <K, V, R : Comparable<R>> Map<out
K, V>.minByOrNull(selector: (Map.Entry<K, V>) -> R): Map.Entry<K, V>? {\n    return
entries.minByOrNull(selector)\n}\n\n/**\n * Returns the smallest value among all values produced by [selector]
function\n * applied to each entry in the map.\n * \n\n * If any of values produced by [selector] function is `NaN`, the
returned result is `NaN`.\n * \n\n * @throws NoSuchElementException if the map is empty.\n
*\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<out K, V>.minOf(selector:
(Map.Entry<K, V>) -> Double): Double {\n    return entries.minOf(selector)\n}\n\n/**\n * Returns the smallest
value among all values produced by [selector] function\n * applied to each entry in the map.\n * \n\n * If any of values
produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n\n * @throws NoSuchElementException
if the map is empty.\n
*\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<out K, V>.minOf(selector:
(Map.Entry<K, V>) -> Float): Float {\n    return entries.minOf(selector)\n}\n\n/**\n * Returns the smallest value
among all values produced by [selector] function\n * applied to each entry in the map.\n * \n\n * @throws
NoSuchElementException if the map is empty.\n
*\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <K, V, R : Comparable<R>> Map<out K,
V>.minOf(selector: (Map.Entry<K, V>) -> R): R {\n    return entries.minOf(selector)\n}\n\n/**\n * Returns the
smallest value among all values produced by [selector] function\n * applied to each entry in the map or `null` if
there are no entries.\n * \n\n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<out K,
V>.minOfOrNull(selector: (Map.Entry<K, V>) -> Double): Double? {\n    return
entries.minOfOrNull(selector)\n}\n\n/**\n * Returns the smallest value among all values produced by [selector]
function\n * applied to each entry in the map or `null` if there are no entries.\n * \n\n * If any of values produced by
[selector] function is `NaN`, the returned result is `NaN`.\n
*\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<out K,
V>.minOfOrNull(selector: (Map.Entry<K, V>) -> Float): Float? {\n    return
entries.minOfOrNull(selector)\n}\n\n/**\n * Returns the smallest value among all values produced by [selector]
function\n * applied to each entry in the map or `null` if there are no entries.\n
*\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <K, V, R : Comparable<R>> Map<out K,
V>.minOfOrNull(selector: (Map.Entry<K, V>) -> R): R? {\n    return entries.minOfOrNull(selector)\n}\n\n/**\n *
Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each entry in the map.\n * \n\n * @throws NoSuchElementException if the map is empty.\n
*\n\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <K, V, R> Map<out K,

```

```

V>.minOfWith(comparator: Comparator<in R>, selector: (Map.Entry<K, V>) -> R): R {\n  return
entries.minOfWith(comparator, selector)\n}\n\n/**\n * Returns the smallest value according to the provided
[comparator]\n * among all values produced by [selector] function applied to each entry in the map or `null` if there
are no entries.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <K, V, R> Map<out K,
V>.minOfWithOrNull(comparator: Comparator<in R>, selector: (Map.Entry<K, V>) -> R): R? {\n  return
entries.minOfWithOrNull(comparator, selector)\n}\n\n@Deprecated("Use minWithOrNull instead.",
ReplaceWith("this.minWithOrNull(comparator)"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince
= "1.5", hiddenSince = "1.6")\npublic fun <K, V> Map<out K, V>.minWith(comparator: Comparator<in
Map.Entry<K, V>>): Map.Entry<K, V>? {\n  return minWithOrNull(comparator)\n}\n\n/**\n * Returns the first
entry having the smallest value according to the provided [comparator] or `null` if there are no entries.\n
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<out K,
V>.minWithOrNull(comparator: Comparator<in Map.Entry<K, V>>): Map.Entry<K, V>? {\n  return
entries.minWithOrNull(comparator)\n}\n\n/**\n * Returns `true` if the map has no entries.\n * \n * @sample
samples.collections.Collections.Aggregates.none\n *\npublic fun <K, V> Map<out K, V>.none(): Boolean {\n
return isEmpty()\n}\n\n/**\n * Returns `true` if no entries match the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.noneWithPredicate\n *\npublic inline fun <K, V> Map<out K,
V>.none(predicate: (Map.Entry<K, V>) -> Boolean): Boolean {\n  if (isEmpty()) return true\n  for (element in
this) if (predicate(element)) return false\n  return true\n}\n\n/**\n * Performs the given [action] on each entry and
returns the map itself afterwards.\n *\n@SinceKotlin("1.1")\npublic inline fun <K, V, M : Map<out K, V>>
M.onEach(action: (Map.Entry<K, V>) -> Unit): M {\n  return apply { for (element in this) action(element)
}\n}\n\n/**\n * Performs the given [action] on each entry, providing sequential index with the entry,\n * and returns
the map itself afterwards.\n * @param [action] function that takes the index of an entry and the entry itself\n * and
performs the action on the entry.\n *\n@SinceKotlin("1.4")\npublic inline fun <K, V, M : Map<out K, V>>
M.onEachIndexed(action: (index: Int, Map.Entry<K, V>) -> Unit): M {\n  return apply {
entries.forEachIndexed(action) }\n}\n\n/**\n * Creates an [Iterable] instance that wraps the original map returning
its entries when being iterated.\n *\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<out K,
V>.asIterable(): Iterable<Map.Entry<K, V>> {\n  return entries\n}\n\n/**\n * Creates a [Sequence] instance that
wraps the original map returning its entries when being iterated.\n *\npublic fun <K, V> Map<out K,
V>.asSequence(): Sequence<Map.Entry<K, V>> {\n  return entries.asSequence()\n}\n\n"/\n * Copyright 2010-
2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n *\npackage kotlin.text\n\n// NOTE:
THIS FILE IS AUTO-GENERATED by the GenerateUnicodeData.kt\n// See:
https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\n// 10 mappings totally\ninternal fun
Char.titlecaseImpl(): String {\n  val uppercase = uppercase()\n  if (uppercase.length > 1) {\n    return if (this ==
"\u0149") uppercase else uppercase[0] + uppercase.substring(1).lowercase()\n  }\n  return
titlecaseChar().toString()\n}\n\n"/\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n *\npackage kotlin.text\n\n/**\n * Converts this character to lower case using Unicode
mapping rules of the invariant locale.\n *\n@Deprecated("Use lowercaseChar() instead.",
ReplaceWith("lowercaseChar()"))\n@DeprecatedSinceKotlin(warningSince =
"1.5")\n@kotlin.internal.InlineOnly\npublic actual inline fun Char.toLowerCase(): Char =
lowercaseChar()\n\n/**\n * Converts this character to lower case using Unicode mapping rules of the invariant
locale.\n * \n * This function performs one-to-one character mapping.\n * To support one-to-many character
mapping use the [lowercase] function.\n * If this character has no mapping equivalent, the character itself is
returned.\n * \n * @sample samples.text.Chars.lowercase\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npubli

```

```

c actual inline fun Char.lowercaseChar(): Char = lowercase()[0]\n\n/**\n * Converts this character to lower case
using Unicode mapping rules of the invariant locale.\n *\n * This function supports one-to-many character mapping,
thus the length of the returned string can be greater than one.\n * For example, ``\u0130'.lowercase()`` returns
``\u0069\u0307``,\n * where ``\u0130`` is the LATIN CAPITAL LETTER I WITH DOT ABOVE character
(`\u0130`).\n * If this character has no lower case mapping, the result of `toString()` of this char is returned.\n *\n *
@sample samples.text.Chars.lowercase\n
*/\n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
c actual inline fun Char.lowercase(): String = toString().asDynamic().toLowerCase().unsafeCast<String>()\n\n/**\n
* Converts this character to upper case using Unicode mapping rules of the invariant locale.\n
*\n\n@Deprecated("Use uppercaseChar() instead.",
ReplaceWith("uppercaseChar()"))\n@DeprecatedSinceKotlin(warningSince =
"1.5")\n@kotlin.internal.InlineOnly\npublic actual inline fun Char.toUpperCase(): Char =
uppercaseChar()\n\n/**\n * Converts this character to upper case using Unicode mapping rules of the invariant
locale.\n *\n * This function performs one-to-one character mapping.\n * To support one-to-many character
mapping use the [uppercase] function.\n * If this character has no mapping equivalent, the character itself is
returned.\n *\n * @sample samples.text.Chars.uppercase\n
*/\n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun
Char.uppercaseChar(): Char {\n    val uppercase = uppercase()\n    return if (uppercase.length > 1) this else
uppercase[0]\n}\n\n/**\n * Converts this character to upper case using Unicode mapping rules of the invariant
locale.\n *\n * This function supports one-to-many character mapping, thus the length of the returned string can be
greater than one.\n * For example, ``\uFB00'.uppercase()`` returns ``\u0046\u0046``,\n * where ``\uFB00`` is the
LATIN SMALL LIGATURE FF character (`\ufb00`).\n * If this character has no upper case mapping, the result of
`toString()` of this char is returned.\n *\n * @sample samples.text.Chars.uppercase\n
*/\n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
c actual inline fun Char.uppercase(): String = toString().asDynamic().toUpperCase().unsafeCast<String>()\n\n/**\n
* Converts this character to title case using Unicode mapping rules of the invariant locale.\n *\n * This function
performs one-to-one character mapping.\n * To support one-to-many character mapping use the [titlecase]
function.\n * If this character has no mapping equivalent, the result of calling [uppercaseChar] is returned.\n *\n *
@sample samples.text.Chars.titlecase\n
*/\n\n@SinceKotlin("1.5")\npublic actual fun Char.titlecaseChar(): Char =
titlecaseCharImpl()\n\n/**\n * Returns `true` if this character is a Unicode high-surrogate code unit (also known as
leading-surrogate code unit).\n *\n\n@public actual fun Char.isHighSurrogate(): Boolean = this in
Char.MIN_HIGH_SURROGATE..Char.MAX_HIGH_SURROGATE)\n\n/**\n * Returns `true` if this character is a
Unicode low-surrogate code unit (also known as trailing-surrogate code unit).\n *\n\n@public actual fun
Char.isLowSurrogate(): Boolean = this in
Char.MIN_LOW_SURROGATE..Char.MAX_LOW_SURROGATE)\n\n/**\n * Returns the Unicode general
category of this character.\n *\n\n@SinceKotlin("1.5")\npublic actual val Char.category: CharCategory\n    get() =
CharCategory.valueOf(getCategoryValue())\n\n/**\n * Returns `true` if this character (Unicode code point) is
defined in Unicode.\n *\n * A character is considered to be defined in Unicode if its [category] is not
[CharCategory.UNASSIGNED].\n *\n\n@SinceKotlin("1.5")\npublic actual fun Char.isDefined(): Boolean {\n    if
(this < "\u0080") {\n        return true\n    }\n    return getCategoryValue() !=
CharCategory.UNASSIGNED.value\n}\n\n/**\n * Returns `true` if this character is a letter.\n *\n * A character is
considered to be a letter if its [category] is [CharCategory.UPPERCASE_LETTER],\n *\n [CharCategory.LOWERCASE_LETTER], [CharCategory.TITLECASE_LETTER],
[CharCategory.MODIFIER_LETTER], or [CharCategory.OTHER_LETTER].\n *\n * @sample
samples.text.Chars.isLetter\n
*/\n\n@SinceKotlin("1.5")\npublic actual fun Char.isLetter(): Boolean {\n    if (this in
'a..'z' || this in 'A..'Z') {\n        return true\n    }\n    if (this < "\u0080") {\n        return false\n    }\n    return
isLetterImpl()\n}\n\n/**\n * Returns `true` if this character is a letter or digit.\n *\n * @see isLetter\n * @see
isDigit\n *\n * @sample samples.text.Chars.isLetterOrDigit\n
*/\n\n@SinceKotlin("1.5")\npublic actual fun

```

```

Char.isLetterOrDigit(): Boolean {\n  if (this in 'a..'z' || this in 'A..'Z' || this in '0..'9') {\n    return true\n  }\n  if (this < "\u0080') {\n    return false\n  }\n  return isDigitImpl() || isLetterImpl()\n}\n\n/**\n * Returns `true` if this character is a digit.\n */\n * A character is considered to be a digit if its [category] is [CharCategory.DECIMAL_DIGIT_NUMBER].\n */\n * @sample samples.text.Chars.isDigit\n */\n\n@SinceKotlin("1.5")\npublic actual fun Char.isDigit(): Boolean {\n  if (this in '0..'9') {\n    return true\n  }\n  if (this < "\u0080') {\n    return false\n  }\n  return isDigitImpl()\n}\n\n/**\n * Returns `true` if this character is upper case.\n */\n * A character is considered to be an upper case character if its [category] is [CharCategory.UPPERCASE_LETTER],\n * or it has contributory property `Other_Uppercase` as defined by the Unicode Standard.\n */\n * @sample samples.text.Chars.isUpperCase\n */\n\n@SinceKotlin("1.5")\npublic actual fun Char.isUpperCase(): Boolean {\n  if (this in 'A..'Z') {\n    return true\n  }\n  if (this < "\u0080') {\n    return false\n  }\n  return isUpperCaseImpl()\n}\n\n/**\n * Returns `true` if this character is lower case.\n */\n * A character is considered to be a lower case character if its [category] is [CharCategory.LOWERCASE_LETTER],\n * or it has contributory property `Other_Lowercase` as defined by the Unicode Standard.\n */\n * @sample samples.text.Chars.isLowerCase\n */\n\n@SinceKotlin("1.5")\npublic actual fun Char.isLowerCase(): Boolean {\n  if (this in 'a..'z') {\n    return true\n  }\n  if (this < "\u0080') {\n    return false\n  }\n  return isLowerCaseImpl()\n}\n\n/**\n * Returns `true` if this character is a title case letter.\n */\n * A character is considered to be a title case letter if its [category] is [CharCategory.TITLECASE_LETTER].\n */\n * @sample samples.text.Chars.isTitleCase\n */\n\n@SinceKotlin("1.5")\npublic actual fun Char.isTitleCase(): Boolean {\n  if (this < "\u0080') {\n    return false\n  }\n  return getCategoryValue() == CharCategory.TITLECASE_LETTER.value\n}\n\n/**\n * Returns `true` if this character is an ISO control character.\n */\n * A character is considered to be an ISO control character if its [category] is [CharCategory.CONTROL],\n * meaning the Char is in the range "\u0000'..' \u001F'" or in the range "\u007F'..' \u009F'".\n */\n * @sample samples.text.Chars.isISOControl\n */\n\n@SinceKotlin("1.5")\npublic actual fun Char.isISOControl(): Boolean {\n  return this <= "\u001F' || this in "\u007F'..' \u009F'\n}\n\n/**\n * Determines whether a character is whitespace according to the Unicode standard.\n */\n * Returns `true` if the character is whitespace.\n */\n * @sample samples.text.Chars.isWhitespace\n */\n\npublic actual fun Char.isWhitespace(): Boolean = isWhitespaceImpl()\n\n/**\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.text\n\nimport kotlin.js.RegExp\n\n/**\n * Converts the characters in the specified array to a string.\n */\n\n@SinceKotlin("1.2")\n@Deprecated("Use CharArray.concatToString() instead", ReplaceWith("\nchars.concatToString()"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5")\npublic actual fun String(chars: CharArray): String {\n  var result = ""\n  for (char in chars) {\n    result += char\n  }\n  return result\n}\n\n/**\n * Converts the characters from a portion of the specified array to a string.\n */\n * @throws IndexOutOfBoundsException if either [offset] or [length] are less than zero\n * or `offset + length` is out of [chars] array bounds.\n */\n\n@SinceKotlin("1.2")\n@Deprecated("Use CharArray.concatToString(startIndex, endIndex) instead", ReplaceWith("\nchars.concatToString(offset, offset + length)"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5")\npublic actual fun String(chars: CharArray, offset: Int, length: Int): String {\n  if (offset < 0 || length < 0 || chars.size - offset < length)\n    throw IndexOutOfBoundsException("size: ${chars.size}; offset: $offset; length: $length")\n  var result = ""\n  for (index in offset until offset + length) {\n    result += chars[index]\n  }\n  return result\n}\n\n/**\n * Concatenates characters in this [CharArray] into a String.\n */\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun CharArray.concatToString(): String {\n  var result = ""\n  for (char in this) {\n    result += char\n  }\n  return result\n}\n\n/**\n * Concatenates characters in this [CharArray] or its subrange into a String.\n */\n * @param startIndex the beginning (inclusive) of the subrange of characters, 0 by default.\n * @param endIndex the end (exclusive) of the subrange of characters, size of this array by default.\n */\n * @throws IndexOutOfBoundsException if [startIndex] is less than zero or [endIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [startIndex] is greater than [endIndex].\n */

```

```

*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual fun CharArray.concatToString(startIndex: Int = 0,
endIndex: Int = this.size): String {\n    AbstractList.checkBoundsIndexes(startIndex, endIndex, this.size)\n    var
result = ""\n    for (index in startIndex until endIndex) {\n        result += this[index]\n    }\n    return
result\n}\n\n/**\n * Returns a [CharArray] containing characters of this string.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun
String.toCharArray(): CharArray {\n    return CharArray(length) { get(it) }\n}\n\n/**\n * Returns a [CharArray]
containing characters of this string or its substring.\n * \n * @param startIndex the beginning (inclusive) of the
substring, 0 by default.\n * @param endIndex the end (exclusive) of the substring, length of this string by default.\n
*\n * @throws IndexOutOfBoundsException if [startIndex] is less than zero or [endIndex] is greater than the length
of this string.\n * @throws IllegalArgumentException if [startIndex] is greater than [endIndex].\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual fun String.toCharArray(startIndex: Int = 0, endIndex: Int = this.length): CharArray {\n    AbstractList.checkBoundsIndexes(startIndex, endIndex, length)\n    return CharArray(endIndex - startIndex) { get(startIndex + it) }\n}\n\n/**\n * Decodes a string from the bytes in UTF-8 encoding in this array.\n * \n * Malformed byte sequences are replaced by the replacement char `\\uFFFD`.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun
ByteArray.decodeToString(): String {\n    return decodeUtf8(this, 0, size, false)\n}\n\n/**\n * Decodes a string from
the bytes in UTF-8 encoding in this array or its subrange.\n * \n * @param startIndex the beginning (inclusive) of the
subrange to decode, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to decode, size of this
array by default.\n * @param throwOnInvalidSequence specifies whether to throw an exception on malformed byte
sequence or replace it by the replacement char `\\uFFFD`.\n * \n * @throws IndexOutOfBoundsException if
[startIndex] is less than zero or [endIndex] is greater than the size of this array.\n * @throws
IllegalArgumentException if [startIndex] is greater than [endIndex].\n * @throws CharacterCodingException if the
byte array contains malformed UTF-8 byte sequence and [throwOnInvalidSequence] is true.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual fun ByteArray.decodeToString(\n    startIndex: Int = 0,\n    endIndex: Int = this.size,\n    throwOnInvalidSequence: Boolean = false\n): String {\n
    AbstractList.checkBoundsIndexes(startIndex, endIndex, this.size)\n    return decodeUtf8(this, startIndex, endIndex,
throwOnInvalidSequence)\n}\n\n/**\n * Encodes this string to an array of bytes in UTF-8 encoding.\n * \n * Any
malformed char sequence is replaced by the replacement byte sequence.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun
String.encodeToArray(): ByteArray {\n    return encodeUtf8(this, 0, length, false)\n}\n\n/**\n * Encodes this
string or its substring to an array of bytes in UTF-8 encoding.\n * \n * @param startIndex the beginning (inclusive)
of the substring to encode, 0 by default.\n * @param endIndex the end (exclusive) of the substring to encode, length
of this string by default.\n * @param throwOnInvalidSequence specifies whether to throw an exception on
malformed char sequence or replace.\n * \n * @throws IndexOutOfBoundsException if [startIndex] is less than zero
or [endIndex] is greater than the length of this string.\n * @throws IllegalArgumentException if [startIndex] is
greater than [endIndex].\n * @throws CharacterCodingException if this string contains malformed char sequence
and [throwOnInvalidSequence] is true.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual fun String.encodeToArray(\n    startIndex: Int = 0,\n    endIndex: Int = this.length,\n    throwOnInvalidSequence: Boolean = false\n): ByteArray {\n
    AbstractList.checkBoundsIndexes(startIndex, endIndex, length)\n    return encodeUtf8(this, startIndex, endIndex,
throwOnInvalidSequence)\n}\n\n/**\n * Returns a copy of this string converted to upper case using the rules of the
default locale.\n * \n * @Deprecated("Use uppercase() instead.")\n
ReplaceWith("uppercase()")\n@DeprecatedSinceKotlin(warningSince =
"1.5")\n@kotlin.internal.InlineOnly\npublic actual inline fun String.toUpperCase(): String =

```

asDynamic().toUpperCase()\n\n/**\n * Returns a copy of this string converted to upper case using Unicode mapping rules of the invariant locale.\n * This function supports one-to-many and many-to-one character mapping,\n * thus the length of the returned string can be different from the length of the original string.\n * @sample samples.text.Strings.uppercase\n

```
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic actual inline fun String.toUpperCase(): String = asDynamic().toUpperCase()\n\n/**\n * Returns a copy of this string converted to lower case using the rules of the default locale.\n * \n@Deprecated("Use lowercase() instead.", ReplaceWith("lowercase()"))\n@DeprecatedSinceKotlin(warningSince =
```

```
"1.5")\n@kotlin.internal.InlineOnly\npublic actual inline fun String.toLowerCase(): String = asDynamic().toLowerCase()\n\n/**\n * Returns a copy of this string converted to lower case using Unicode mapping rules of the invariant locale.\n * This function supports one-to-many and many-to-one character mapping,\n * thus the length of the returned string can be different from the length of the original string.\n * @sample samples.text.Strings.lowercase\n
```

```
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic actual inline fun String.toLowerCase(): String = asDynamic().toLowerCase()\n\n@kotlin.internal.InlineOnly\ninternal actual inline fun String.nativeIndexOf(str: String, fromIndex: Int): Int = asDynamic().indexOf(str, fromIndex)\n\n@kotlin.internal.InlineOnly\ninternal actual inline fun String.nativeLastIndexOf(str: String, fromIndex: Int): Int = asDynamic().lastIndexOf(str, fromIndex)\n\n@kotlin.internal.InlineOnly\ninternal inline fun String.nativeStartsWith(s: String, position: Int): Boolean = asDynamic().startsWith(s, position)\n\n@kotlin.internal.InlineOnly\ninternal inline fun String.nativeEndsWith(s: String): Boolean = asDynamic().endsWith(s)\n\n@kotlin.internal.InlineOnly\npublic actual inline fun String.substring(startIndex: Int): String = asDynamic().substring(startIndex)\n\n@kotlin.internal.InlineOnly\npublic actual inline fun String.substring(startIndex: Int, endIndex: Int): String = asDynamic().substring(startIndex, endIndex)\n\n@Deprecated("Use String.plus() instead", ReplaceWith("this + str"))\n@DeprecatedSinceKotlin(warningSince = "1.6")\n@kotlin.internal.InlineOnly\npublic inline fun String.concat(str: String): String = asDynamic().concat(str)\n\n@Deprecated("Use Regex.findAll() instead or invoke matches() on String dynamically:
```

```
this.asDynamic().match(regex)")\n@DeprecatedSinceKotlin(warningSince = "1.6")\n@kotlin.internal.InlineOnly\npublic inline fun String.match(regex: String): Array<String>? = asDynamic().match(regex)\n\n//native public fun String.trim(): String\n\n//TODO: String.replace to implement effective trimLeading and trimTrailing\n\n@kotlin.internal.InlineOnly\ninternal inline fun String.nativeReplace(pattern: RegExp, replacement: String): String = asDynamic().replace(pattern, replacement)\n\n/**\n * Compares two strings lexicographically, optionally ignoring case differences.\n * If [ignoreCase] is true, the result of `Char.toUpperCaseChar().toLowerCaseChar()` on each character is compared.\n
```

```
*\n@SinceKotlin("1.2")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual fun String.compareTo(other: String, ignoreCase: Boolean = false): Int {\n    if (ignoreCase) {\n        val n1 = this.length\n        val n2 = other.length\n        val min = minOf(n1, n2)\n        if (min == 0) return n1 - n2\n        for (index in 0 until min) {\n            var thisChar = this[index]\n            var otherChar = other[index]\n            if (thisChar != otherChar) {\n                thisChar = thisChar.toUpperCaseChar()\n                otherChar = otherChar.toUpperCaseChar()\n                if (thisChar != otherChar) {\n                    thisChar = thisChar.toLowerCaseChar()\n                    otherChar = otherChar.toLowerCaseChar()\n                    if (thisChar != otherChar) {\n                        return thisChar.compareTo(otherChar)\n                    }\n                }\n            }\n            return n1 - n2\n        } else {\n            return compareTo(other)\n        }\n    }\n\n/**\n * Returns `true` if the contents of this char sequence are equal to the contents of the specified [other],\n * i.e. both char sequences contain the same number of the same characters in the same order.\n * @sample samples.text.Strings.contentEquals\n
```

```
*\n@SinceKotlin("1.5")\npublic actual infix fun CharSequence?.contentEquals(other: CharSequence?): Boolean = contentEqualsImpl(other)\n\n/**\n * Returns `true` if the contents of this char sequence are equal to the contents of the specified [other], optionally ignoring case difference.\n * @param ignoreCase `true` to ignore character case
```

```

when comparing contents.\n * \n * @sample samples.text.Strings.contentEquals\n * \n * @SinceKotlin("1.5")\npublic
actual fun CharSequence?.contentEquals(other: CharSequence?, ignoreCase: Boolean): Boolean {\n    return if
(ignoreCase)\n        this.contentEqualsIgnoreCaseImpl(other)\n    else\n
this.contentEqualsImpl(other)\n}\n\nprivate val STRING_CASE_INSENSITIVE_ORDER = Comparator<String>
{ a, b -> a.compareTo(b, ignoreCase = true) }\n\n@SinceKotlin("1.2")\npublic actual val
String.Companion.CASE_INSENSITIVE_ORDER: Comparator<String>\n    get() =
STRING_CASE_INSENSITIVE_ORDER\n", /*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n
*\n * @file:kotlin.jvm.JvmMultifileClass\n * @file:kotlin.jvm.JvmName("CharsKt")\n * @package kotlin.text\n * \n * \n
*\n * Returns the numeric value of the decimal digit that this Char represents.\n * Throws an exception if this Char is
not a valid decimal digit.\n * \n * A Char is considered to represent a decimal digit if [isDigit] is true for the Char.\n
*\n * In this case, the Unicode decimal digit value of the character is returned.\n * \n * @sample
samples.text.Chars.digitToInt\n
*\n * @SinceKotlin("1.5")\n * @WasExperimental(ExperimentalStdlibApi::class)\npublic fun Char.digitToInt(): Int
{\n    return digitOf(this, 10).also {\n        if (it < 0) throw IllegalArgumentException("Char $this is not a decimal
digit")\n    }\n}\n\n*\n * Returns the numeric value of the digit that this Char represents in the specified [radix].\n
*\n * Throws an exception if the [radix] is not in the range `2..36` or if this Char is not a valid digit in the specified
[radix].\n * \n * A Char is considered to represent a digit in the specified [radix] if at least one of the following is
true:\n * - [isDigit] is `true` for the Char and the Unicode decimal digit value of the character is less than the
specified [radix]. In this case the decimal digit value is returned.\n * - The Char is one of the uppercase Latin letters
'A' through 'Z' and its [code] is less than `radix + 'A'.code - 10`. In this case, `this.code - 'A'.code + 10` is
returned.\n * - The Char is one of the lowercase Latin letters 'a' through 'z' and its [code] is less than `radix +
'a'.code - 10`. In this case, `this.code - 'a'.code + 10` is returned.\n * - The Char is one of the fullwidth Latin
capital letters '\uFF21' through '\uFF3A' and its [code] is less than `radix + 0xFF21 - 10`. In this case, `this.code -
0xFF21 + 10` is returned.\n * - The Char is one of the fullwidth Latin small letters '\uFF41' through '\uFF5A' and its
[code] is less than `radix + 0xFF41 - 10`. In this case, `this.code - 0xFF41 + 10` is returned.\n * \n * @sample
samples.text.Chars.digitToInt\n
*\n * @SinceKotlin("1.5")\n * @WasExperimental(ExperimentalStdlibApi::class)\npublic fun Char.digitToInt(radix:
Int): Int? {\n    return digitToIntOrNull(radix) ?: throw IllegalArgumentException("Char $this is not a digit in the
given radix=$radix")\n}\n\n*\n * Returns the numeric value of the decimal digit that this Char represents, or
`null` if this Char is not a valid decimal digit.\n * \n * A Char is considered to represent a decimal digit if [isDigit]
is true for the Char.\n * In this case, the Unicode decimal digit value of the character is returned.\n * \n * @sample
samples.text.Chars.digitToIntOrNull\n
*\n * @SinceKotlin("1.5")\n * @WasExperimental(ExperimentalStdlibApi::class)\npublic fun
Char.digitToIntOrNull(): Int? {\n    return digitOf(this, 10).takeIf { it >= 0 }\n}\n\n*\n * Returns the numeric
value of the digit that this Char represents in the specified [radix], or `null` if this Char is not a valid digit in the
specified [radix].\n * Throws an exception if the [radix] is not in the range `2..36`.\n * \n * A Char is considered to
represent a digit in the specified [radix] if at least one of the following is true:\n * - [isDigit] is `true` for the
Char and the Unicode decimal digit value of the character is less than the specified [radix]. In this case the decimal
digit value is returned.\n * - The Char is one of the uppercase Latin letters 'A' through 'Z' and its [code] is less
than `radix + 'A'.code - 10`. In this case, `this.code - 'A'.code + 10` is returned.\n * - The Char is one of the
lowercase Latin letters 'a' through 'z' and its [code] is less than `radix + 'a'.code - 10`. In this case, `this.code -
'a'.code + 10` is returned.\n * - The Char is one of the fullwidth Latin capital letters '\uFF21' through '\uFF3A' and
its [code] is less than `radix + 0xFF21 - 10`. In this case, `this.code - 0xFF21 + 10` is returned.\n * - The Char is
one of the fullwidth Latin small letters '\uFF41' through '\uFF5A' and its [code] is less than `radix + 0xFF41 - 10`.
In this case, `this.code - 0xFF41 + 10` is returned.\n * \n * @sample samples.text.Chars.digitToIntOrNull\n
*\n * @SinceKotlin("1.5")\n * @WasExperimental(ExperimentalStdlibApi::class)\npublic fun

```

```

Char.digitToIntOrNull(radix: Int): Int? {\n  checkRadix(radix)\n  return digitOf(this, radix).takeIf { it >= 0
}\n}\n\n/**\n * Returns the Char that represents this decimal digit.\n * Throws an exception if this value is not in the
range `0..9`.\n * If this value is in `0..9`, the decimal digit Char with code `0.code + this` is returned.\n *\n *
@sample samples.text.Chars.digitToChar\n
*\n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun Int.digitToChar(): Char
{\n  if (this in 0..9) {\n    return '0' + this\n  }\n  throw IllegalArgumentException("\nInt $this is not a decimal
digit")\n}\n\n/**\n * Returns the Char that represents this numeric digit value in the specified [radix].\n * Throws
an exception if the [radix] is not in the range `2..36` or if this value is not in the range `0 until radix`.\n *\n * If this
value is less than `10`, the decimal digit Char with code `0.code + this` is returned.\n * Otherwise, the uppercase
Latin letter with code `A.code + this - 10` is returned.\n *\n * @sample samples.text.Chars.digitToChar\n
*\n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun Int.digitToChar(radix:
Int): Char {\n  if (radix !in 2..36) {\n    throw IllegalArgumentException("Invalid radix: $radix. Valid radix
values are in range 2..36")\n  }\n  if (this < 0 || this >= radix) {\n    throw IllegalArgumentException("Digit
$this does not represent a valid digit in radix $radix")\n  }\n  return if (this < 10) {\n    '0' + this\n  } else {\n
  'A' + this - 10\n  }\n}\n\n/**\n * Converts this character to lower case using Unicode mapping rules of the
invariant locale.\n *\n * @sample samples.text.Chars.lowercase\n
*\n\n@Deprecated("Use lowercaseChar() instead."),
ReplaceWith("lowercaseChar()")\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic expect fun
Char.toLowerCase(): Char\n\n/**\n * Converts this character to lower case using Unicode mapping rules of the
invariant locale.\n *\n * This function performs one-to-one character mapping.\n * To support one-to-many
character mapping use the [lowercase] function.\n * If this character has no mapping equivalent, the character itself
is returned.\n *\n * @sample samples.text.Chars.lowercase\n
*\n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
Char.lowercaseChar(): Char\n\n/**\n * Converts this character to lower case using Unicode mapping rules of the
invariant locale.\n *\n * This function supports one-to-many character mapping, thus the length of the returned
string can be greater than one.\n * For example, `"\u0130".lowercase()` returns `"\u0069\u0307"`,\n * where
`"\u0130` is the LATIN CAPITAL LETTER I WITH DOT ABOVE character (`\ufffd\u0130`).\n * If this character
has no lower case mapping, the result of `toString()` of this char is returned.\n *\n * @sample
samples.text.Chars.lowercase\n
*\n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
Char.lowercase(): String\n\n/**\n * Converts this character to upper case using Unicode mapping rules of the
invariant locale.\n *\n * @sample samples.text.Chars.uppercase\n
*\n\n@Deprecated("Use uppercaseChar() instead."),
ReplaceWith("uppercaseChar()")\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic expect fun
Char.toUpperCase(): Char\n\n/**\n * Converts this character to upper case using Unicode mapping rules of the
invariant locale.\n *\n * This function performs one-to-one character mapping.\n * To support one-to-many
character mapping use the [uppercase] function.\n * If this character has no mapping equivalent, the character itself
is returned.\n *\n * @sample samples.text.Chars.uppercase\n
*\n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
Char.uppercaseChar(): Char\n\n/**\n * Converts this character to upper case using Unicode mapping rules of the
invariant locale.\n *\n * This function supports one-to-many character mapping, thus the length of the returned
string can be greater than one.\n * For example, `"\uFB00".uppercase()` returns `"\u0046\u0046"`,\n * where
`"\uFB00` is the LATIN SMALL LIGATURE FF character (`\ufffd\u0046\u0046`).\n * If this character has no upper
case mapping, the result of `toString()` of this char is returned.\n *\n * @sample samples.text.Chars.uppercase\n
*\n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
Char.uppercase(): String\n\n/**\n * Converts this character to title case using Unicode mapping rules of the
invariant locale.\n *\n * This function performs one-to-one character mapping.\n * To support one-to-many
character mapping use the [titlecase] function.\n * If this character has no mapping equivalent, the result of calling
[uppercaseChar] is returned.\n *\n * @sample samples.text.Chars.titlecase\n
*\n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun Char.titlecaseChar(): Char\n\n/**\n * Converts this character to title case using Unicode mapping rules of

```



```

[NoSuchElementException] if no such element is found.\n * \n * The operation is _terminal_.\n * \n\npublic inline fun
<T> Sequence<T>.first(predicate: (T) -> Boolean): T {\n  for (element in this) if (predicate(element)) return
element\n  throw NoSuchElementException("Sequence contains no element matching the predicate.")\n}\n\n/**\n * Returns the first non-null value produced by [transform] function being applied to elements of this sequence in
iteration order,\n * or throws [NoSuchElementException] if no non-null value was produced.\n * \n * The operation
is _terminal_.\n * \n * @sample samples.collections.Collections.Transformations.firstNotNullOf\n
*\n*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Any>
Sequence<T>.firstNotNullOf(transform: (T) -> R?): R {\n  return firstNotNullOfOrNull(transform) ?: throw
NoSuchElementException("No element of the sequence was transformed to a non-null value.")\n}\n\n/**\n * Returns the first non-null value produced by [transform] function being applied to elements of this sequence in
iteration order,\n * or `null` if no non-null value was produced.\n * \n * The operation is _terminal_.\n * \n *
@sample samples.collections.Collections.Transformations.firstNotNullOf\n
*\n*\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Any>
Sequence<T>.firstNotNullOfOrNull(transform: (T) -> R?): R? {\n  for (element in this) {\n    val result =
transform(element)\n    if (result != null) {\n      return result\n    }\n  }\n  return null\n}\n\n/**\n * Returns the first element, or `null` if the sequence is empty.\n * \n * The operation is _terminal_.\n * \n\npublic fun
<T> Sequence<T>.firstOrNull(): T? {\n  val iterator = iterator()\n  if (!iterator.hasNext())\n    return null\n  return iterator.next()\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if element was
not found.\n * \n * The operation is _terminal_.\n * \n\npublic inline fun <T> Sequence<T>.firstOrNull(predicate: (T)
-> Boolean): T? {\n  for (element in this) if (predicate(element)) return element\n  return null\n}\n\n/**\n * Returns first index of [element], or -1 if the sequence does not contain element.\n * \n * The operation is
_terminal_.\n * \n\npublic fun <@kotlin.internal.OnlyInputTypes T> Sequence<T>.indexOf(element: T): Int {\n  var
index = 0\n  for (item in this) {\n    checkIndexOverflow(index)\n    if (element == item)\n      return
index\n    index++\n  }\n  return -1\n}\n\n/**\n * Returns index of the first element matching the given
[predicate], or -1 if the sequence does not contain such element.\n * \n * The operation is _terminal_.\n * \n\npublic
inline fun <T> Sequence<T>.indexOfFirst(predicate: (T) -> Boolean): Int {\n  var index = 0\n  for (item in this)
{\n    checkIndexOverflow(index)\n    if (predicate(item))\n      return index\n    index++\n  }\n  return -
1\n}\n\n/**\n * Returns index of the last element matching the given [predicate], or -1 if the sequence does not
contain such element.\n * \n * The operation is _terminal_.\n * \n\npublic inline fun <T>
Sequence<T>.indexOfLast(predicate: (T) -> Boolean): Int {\n  var lastIndex = -1\n  var index = 0\n  for (item in
this) {\n    checkIndexOverflow(index)\n    if (predicate(item))\n      lastIndex = index\n    index++\n  }\n  return
lastIndex\n}\n\n/**\n * Returns the last element.\n * \n * The operation is _terminal_.\n * \n * @throws
NoSuchElementException if the sequence is empty.\n * \n * @sample
samples.collections.Collections.Elements.last\n*\n*\npublic fun <T> Sequence<T>.last(): T {\n  val iterator =
iterator()\n  if (!iterator.hasNext())\n    throw NoSuchElementException("Sequence is empty.")\n  var last =
iterator.next()\n  while (iterator.hasNext())\n    last = iterator.next()\n  return last\n}\n\n/**\n * Returns the last
element matching the given [predicate].\n * \n * The operation is _terminal_.\n * \n * @throws
NoSuchElementException if no such element is found.\n * \n * @sample
samples.collections.Collections.Elements.last\n*\n*\npublic inline fun <T> Sequence<T>.last(predicate: (T) ->
Boolean): T {\n  var last: T? = null\n  var found = false\n  for (element in this) {\n    if (predicate(element))
{\n      last = element\n      found = true\n    }\n  }\n  if (!found) throw
NoSuchElementException("Sequence contains no element matching the predicate.")\n}\n\n@Suppress("UNCHECKED_CAST")\nreturn last as T\n}\n\n/**\n * Returns last index of [element], or -1 if the
sequence does not contain element.\n * \n * The operation is _terminal_.\n * \n\npublic fun
<@kotlin.internal.OnlyInputTypes T> Sequence<T>.lastIndexOf(element: T): Int {\n  var lastIndex = -1\n  var
index = 0\n  for (item in this) {\n    checkIndexOverflow(index)\n    if (element == item)\n      lastIndex =
index\n    index++\n  }\n  return lastIndex\n}\n\n/**\n * Returns the last element, or `null` if the sequence is
empty.\n * \n * The operation is _terminal_.\n * \n * @sample samples.collections.Collections.Elements.last\n

```

```

*^public fun <T> Sequence<T>.lastOrNull(): T? {
    val iterator = iterator()
    if (!iterator.hasNext())
        return null
    var last = iterator.next()
    while (iterator.hasNext())
        last = iterator.next()
    return last
}

* Returns the last element matching the given [predicate], or `null` if no such element was found.
* The operation is _terminal_.
* @sample samples.collections.Collections.Elements.last

inline fun <T> Sequence<T>.lastOrNull(predicate: (T) -> Boolean): T? {
    var last: T? = null
    for (element in this) {
        if (predicate(element))
            last = element
    }
    return last
}

* Returns the single element, or throws an exception if the sequence is empty or has more than one element.
* The operation is _terminal_.
*^public fun <T> Sequence<T>.single(): T {
    val iterator = iterator()
    if (!iterator.hasNext())
        throw NoSuchElementException("Sequence is empty.")
    val single = iterator.next()
    if (iterator.hasNext())
        throw IllegalArgumentException("Sequence has more than one element.")
    return single
}

* Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.
* The operation is _terminal_.
*^public inline fun <T> Sequence<T>.single(predicate: (T) -> Boolean): T {
    var single: T? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw IllegalArgumentException("Sequence contains more than one matching element.")
            single = element
            found = true
        }
    }
    if (!found) throw NoSuchElementException("Sequence contains no element matching the predicate.")
    @Suppress("UNCHECKED_CAST")
    return single as T
}

* Returns single element, or `null` if the sequence is empty or has more than one element.
* The operation is _terminal_.
*^public fun <T> Sequence<T>.singleOrNull(): T? {
    val iterator = iterator()
    if (!iterator.hasNext())
        return null
    val single = iterator.next()
    if (iterator.hasNext())
        return null
    return single
}

* Returns the single element matching the given [predicate], or `null` if element was not found or more than one element was found.
* The operation is _terminal_.
*^public inline fun <T> Sequence<T>.singleOrNull(predicate: (T) -> Boolean): T? {
    var single: T? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) return null
            single = element
            found = true
        }
    }
    if (!found) return null
    return single
}

* Returns a sequence containing all elements except first [n] elements.
* The operation is _intermediate_ and _stateless_.
* @throws IllegalArgumentException if [n] is negative.
* @sample samples.collections.Collections.Transformations.drop
*^public fun <T> Sequence<T>.drop(n: Int): Sequence<T> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return when {
        n == 0 -> this
        this is DropTakeSequence -> this.drop(n)
        else -> DropSequence(this, n)
    }
}

* Returns a sequence containing all elements except first elements that satisfy the given [predicate].
* The operation is _intermediate_ and _stateless_.
* @sample samples.collections.Collections.Transformations.dropWhile
*^public fun <T> Sequence<T>.dropWhile(predicate: (T) -> Boolean): Sequence<T> {
    return DropWhileSequence(this, predicate)
}

* Returns a sequence containing only elements matching the given [predicate].
* The operation is _intermediate_ and _stateless_.
* @sample samples.collections.Collections.Filtering.filter
*^public fun <T> Sequence<T>.filter(predicate: (T) -> Boolean): Sequence<T> {
    return FilteringSequence(this, true, predicate)
}

* Returns a sequence containing only elements matching the given [predicate].
* @param [predicate] function that takes the index of an element and the element itself
* and returns the result of predicate evaluation on the element.
* The operation is _intermediate_ and _stateless_.
* @sample samples.collections.Collections.Filtering.filterIndexed
*^public fun <T> Sequence<T>.filterIndexed(predicate: (index: Int, T) -> Boolean): Sequence<T> {
    // TODO: Rewrite with generalized MapFilterIndexingSequence
    return TransformingSequence(FilteringSequence(IndexingSequence(this), true, { predicate(it.index, it.value) }), { it.value })
}

* Appends all elements matching the given [predicate] to the given [destination].
* @param [predicate] function that takes the index of an element and the element itself
* and returns the result of predicate evaluation on the element.
* The operation is _terminal_.
* @sample samples.collections.Collections.Filtering.filterIndexedTo
*^public inline fun <T, C : MutableCollection<in T>> Sequence<T>.filterIndexedTo(destination: C, predicate: (index: Int, T) -> Boolean): C {
    forEachIndexed { index, element ->
        if (predicate(index, element)) destination.add(element)
    }
    return destination
}

```

```

destination\n}\n\n/**\n * Returns a sequence containing all elements that are instances of specified type parameter
R.\n *\n * The operation is _intermediate_ and _stateless_.\n *\n * @sample
samples.collections.Collections.Filtering.filterIsInstance\n *\npublic inline fun <reified R>
Sequence<*>.filterIsInstance(): Sequence<@kotlin.internal.NoInfer R> {\n
@Suppress("UNCHECKED_CAST")\n    return filter { it is R } as Sequence<R>\n}\n\n/**\n * Appends all
elements that are instances of specified type parameter R to the given [destination].\n *\n * The operation is
_terminal_.\n *\n * @sample samples.collections.Collections.Filtering.filterIsInstanceTo\n *\npublic inline fun
<reified R, C : MutableCollection<in R>> Sequence<*>.filterIsInstanceTo(destination: C): C {\n    for (element in
this) if (element is R) destination.add(element)\n    return destination\n}\n\n/**\n * Returns a sequence containing
all elements not matching the given [predicate].\n *\n * The operation is _intermediate_ and _stateless_.\n *\n *
@sample samples.collections.Collections.Filtering.filter\n *\npublic fun <T> Sequence<T>.filterNot(predicate: (T)
-> Boolean): Sequence<T> {\n    return FilteringSequence(this, false, predicate)\n}\n\n/**\n * Returns a sequence
containing all elements that are not `null`.\n *\n * The operation is _intermediate_ and _stateless_.\n *\n * @sample
samples.collections.Collections.Filtering.filterNotNull\n *\npublic fun <T : Any> Sequence<T?>.filterNotNull():
Sequence<T> {\n    @Suppress("UNCHECKED_CAST")\n    return filterNot { it == null } as
Sequence<T>\n}\n\n/**\n * Appends all elements that are not `null` to the given [destination].\n *\n * The operation
is _terminal_.\n *\n * @sample samples.collections.Collections.Filtering.filterNotNullTo\n *\npublic fun <C :
MutableCollection<in T>, T : Any> Sequence<T?>.filterNotNullTo(destination: C): C {\n    for (element in this) if
(element != null) destination.add(element)\n    return destination\n}\n\n/**\n * Appends all elements not matching
the given [predicate] to the given [destination].\n *\n * The operation is _terminal_.\n *\n * @sample
samples.collections.Collections.Filtering.filterTo\n *\npublic inline fun <T, C : MutableCollection<in T>>
Sequence<T>.filterNotTo(destination: C, predicate: (T) -> Boolean): C {\n    for (element in this) if
(!predicate(element)) destination.add(element)\n    return destination\n}\n\n/**\n * Appends all elements matching
the given [predicate] to the given [destination].\n *\n * The operation is _terminal_.\n *\n * @sample
samples.collections.Collections.Filtering.filterTo\n *\npublic inline fun <T, C : MutableCollection<in T>>
Sequence<T>.filterTo(destination: C, predicate: (T) -> Boolean): C {\n    for (element in this) if (predicate(element))
destination.add(element)\n    return destination\n}\n\n/**\n * Returns a sequence containing first [n] elements.\n *\n *
The operation is _intermediate_ and _stateless_.\n *\n * @throws IllegalArgumentException if [n] is negative.\n *\n *
@sample samples.collections.Collections.Transformations.take\n *\npublic fun <T> Sequence<T>.take(n: Int):
Sequence<T> {\n    require(n >= 0) { "Requested element count $n is less than zero." }\n    return when {\n        n
== 0 -> emptySequence()\n        this is DropTakeSequence -> this.take(n)\n        else -> TakeSequence(this, n)\n    }\n}\n\n/**\n * Returns a sequence containing first elements satisfying the given [predicate].\n *\n * The operation
is _intermediate_ and _stateless_.\n *\n * @sample samples.collections.Collections.Transformations.take\n *\npublic fun <T> Sequence<T>.takeWhile(predicate: (T) -> Boolean): Sequence<T> {\n    return
TakeWhileSequence(this, predicate)\n}\n\n/**\n * Returns a sequence that yields elements of this sequence sorted
according to their natural sort order.\n *\n * The sort is _stable_. It means that equal elements preserve their order
relative to each other after sorting.\n *\n * The operation is _intermediate_ and _stateful_.\n *\npublic fun <T :
Comparable<T>> Sequence<T>.sorted(): Sequence<T> {\n    return object : Sequence<T> {\n        override fun
iterator(): Iterator<T> {\n            val sortedList = this@sorted.toMutableList()\n            sortedList.sort()\n
return sortedList.iterator()\n        }\n    }\n}\n\n/**\n * Returns a sequence that yields elements of this sequence
sorted according to natural sort order of the value returned by specified [selector] function.\n *\n * The sort is
_stable_. It means that equal elements preserve their order relative to each other after sorting.\n *\n * The operation
is _intermediate_ and _stateful_.\n *\n * @sample samples.collections.Collections.Sorting.sortedBy\n *\npublic
inline fun <T, R : Comparable<R>> Sequence<T>.sortedBy(crossinline selector: (T) -> R?): Sequence<T> {\n    return
sortedWith(compareBy(selector))\n}\n\n/**\n * Returns a sequence that yields elements of this sequence
sorted descending according to natural sort order of the value returned by specified [selector] function.\n *\n * The
sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n *\n * The
operation is _intermediate_ and _stateful_.\n *\npublic inline fun <T, R : Comparable<R>>

```

```

Sequence<T>.sortedByDescending(crossinline selector: (T) -> R?): Sequence<T> {\n  return
sortedWith(compareByDescending(selector))\n}\n\n/**\n * Returns a sequence that yields elements of this sequence
sorted descending according to their natural sort order.\n * \n * The sort is _stable_. It means that equal elements
preserve their order relative to each other after sorting.\n * \n * The operation is _intermediate_ and _stateful_.\n
*/\npublic fun <T : Comparable<T>> Sequence<T>.sortedDescending(): Sequence<T> {\n  return
sortedWith(reverseOrder())\n}\n\n/**\n * Returns a sequence that yields elements of this sequence sorted according
to the specified [comparator].\n * \n * The sort is _stable_. It means that equal elements preserve their order relative
to each other after sorting.\n * \n * The operation is _intermediate_ and _stateful_.\n */\npublic fun <T>
Sequence<T>.sortedWith(comparator: Comparator<in T>): Sequence<T> {\n  return object : Sequence<T> {\n
override fun iterator(): Iterator<T> {\n      val sortedList = this@sortedWith.toList()\n
sortedList.sortWith(comparator)\n      return sortedList.iterator()\n    }\n  }\n}\n\n/**\n * Returns a [Map]
containing key-value pairs provided by [transform] function\n * applied to elements of the given sequence.\n * \n *
If any of two pairs would have the same key the last one gets added to the map.\n * \n * The returned map preserves
the entry iteration order of the original sequence.\n * \n * The operation is _terminal_.\n * \n * @sample
samples.collections.Collections.Transformations.associate\n */\npublic inline fun <T, K, V>
Sequence<T>.associate(transform: (T) -> Pair<K, V>): Map<K, V> {\n  return associateTo(LinkedHashMap<K,
V>(), transform)\n}\n\n/**\n * Returns a [Map] containing the elements from the given sequence indexed by the
key\n * returned from [keySelector] function applied to each element.\n * \n * If any two elements would have the
same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry
iteration order of the original sequence.\n * \n * The operation is _terminal_.\n * \n * @sample
samples.collections.Collections.Transformations.associateBy\n */\npublic inline fun <T, K>
Sequence<T>.associateBy(keySelector: (T) -> K): Map<K, T> {\n  return associateByTo(LinkedHashMap<K,
T>(), keySelector)\n}\n\n/**\n * Returns a [Map] containing the values provided by [valueTransform] and indexed
by [keySelector] functions applied to elements of the given sequence.\n * \n * If any two elements would have the
same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry
iteration order of the original sequence.\n * \n * The operation is _terminal_.\n * \n * @sample
samples.collections.Collections.Transformations.associateByWithValueTransform\n */\npublic inline fun <T, K, V>
Sequence<T>.associateBy(keySelector: (T) -> K, valueTransform: (T) -> V): Map<K, V> {\n  return
associateByTo(LinkedHashMap<K, V>(), keySelector, valueTransform)\n}\n\n/**\n * Populates and returns the
[destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function applied to
each element of the given sequence\n * and value is the element itself.\n * \n * If any two elements would have the
same key returned by [keySelector] the last one gets added to the map.\n * \n * The operation is _terminal_.\n * \n *
@sample samples.collections.Collections.Transformations.associateByTo\n */\npublic inline fun <T, K, M :
MutableMap<in K, in T>> Sequence<T>.associateByTo(destination: M, keySelector: (T) -> K): M {\n  for
(element in this) {\n      destination.put(keySelector(element), element)\n    }\n  }\n  return destination\n}\n\n/**\n *
Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the
[keySelector] function and\n * and value is provided by the [valueTransform] function applied to elements of the
given sequence.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets
added to the map.\n * \n * The operation is _terminal_.\n * \n * @sample
samples.collections.Collections.Transformations.associateByToWithValueTransform\n */\npublic inline fun <T, K,
V, M : MutableMap<in K, in V>> Sequence<T>.associateByTo(destination: M, keySelector: (T) -> K,
valueTransform: (T) -> V): M {\n  for (element in this) {\n      destination.put(keySelector(element),
valueTransform(element))\n    }\n  }\n  return destination\n}\n\n/**\n * Populates and returns the [destination] mutable
map with key-value pairs\n * provided by [transform] function applied to each element of the given sequence.\n * \n *
If any of two pairs would have the same key the last one gets added to the map.\n * \n * The operation is
_terminal_.\n * \n * @sample samples.collections.Collections.Transformations.associateTo\n */\npublic inline fun
<T, K, V, M : MutableMap<in K, in V>> Sequence<T>.associateTo(destination: M, transform: (T) -> Pair<K, V>):
M {\n  for (element in this) {\n      destination += transform(element)\n    }\n  }\n  return destination\n}\n\n/**\n *

```

Returns a [Map] where keys are elements from the given sequence and values are produced by the [valueSelector] function applied to each element.

If any two elements are equal, the last one gets added to the map.

The returned map preserves the entry iteration order of the original sequence.

The operation is `_terminal_`.

```

@SinceKotlin("1.3")
public inline fun <K, V> Sequence<K>.associateWith(valueSelector: (K) -> V):
Map<K, V> {
    val result = LinkedHashMap<K, V>()
    return associateWithTo(result,
valueSelector)
}

```

Populates and returns the [destination] mutable map with key-value pairs for each element of the given sequence, where key is the element itself and value is provided by the [valueSelector] function applied to that key.

If any two elements are equal, the last one overwrites the former value in the map.

The operation is `_terminal_`.

```

@sample
samples.collections.Collections.Transformations.associateWithTo
@SinceKotlin("1.3")
public inline fun
<K, V, M : MutableMap<in K, in V>> Sequence<K>.associateWithTo(destination: M, valueSelector: (K) -> V): M
{
    for (element in this) {
        destination.put(element, valueSelector(element))
    }
    return
destination
}

```

Appends all elements to the given [destination] collection.

The operation is `_terminal_`.

```

public fun <T, C : MutableCollection<in T>> Sequence<T>.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

```

Returns a new [HashSet] of all elements.

The operation is `_terminal_`.

```

public fun <T> Sequence<T>.toHashSet(): HashSet<T> {
    return toCollection(HashSet<T>())
}

```

Returns a [List] containing all elements.

The operation is `_terminal_`.

```

public fun <T> Sequence<T>.toList(): List<T> {
    return
this.toMutableList().optimizeReadOnlyList()
}

```

Returns a new [MutableList] filled with all elements of this sequence.

The operation is `_terminal_`.

```

public fun <T> Sequence<T>.toMutableList():
MutableList<T> {
    return toCollection(ArrayList<T>())
}

```

Returns a [Set] of all elements.

The returned set preserves the element iteration order of the original sequence.

The operation is `_terminal_`.

```

public fun <T> Sequence<T>.toSet(): Set<T> {
    return
toCollection(LinkedHashSet<T>()).optimizeReadOnlySet()
}

```

Returns a single sequence of all elements from results of [transform] function being invoked on each element of original sequence.

The operation is `_intermediate_` and `_stateless_`.

```

@sample
samples.collections.Collections.Transformations.flatMap
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("flatMapIterable")
public fun <T, R>
Sequence<T>.flatMap(transform: (T) -> Iterable<R>): Sequence<R> {
    return FlatteningSequence(this,
transform, Iterable<R>::iterator)
}

```

Returns a single sequence of all elements from results of [transform] function being invoked on each element of original sequence.

The operation is `_intermediate_` and `_stateless_`.

```

@sample
samples.collections.Collections.Transformations.flatMap
public fun <T, R>
Sequence<T>.flatMap(transform: (T) -> Sequence<R>): Sequence<R> {
    return FlatteningSequence(this,
transform, Sequence<R>::iterator)
}

```

Returns a single sequence of all elements yielded from results of [transform] function being invoked on each element and its index in the original sequence.

The operation is `_intermediate_` and `_stateless_`.

```

@sample
samples.collections.Collections.Transformations.flatMapIndexed
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("flatMapIndexedIterable")
public fun <T, R>
Sequence<T>.flatMapIndexed(transform: (index: Int, T) -> Iterable<R>): Sequence<R> {
    return
flatMapIndexed(this, transform, Iterable<R>::iterator)
}

```

Returns a single sequence of all elements yielded from results of [transform] function being invoked on each element and its index in the original sequence.

The operation is `_intermediate_` and `_stateless_`.

```

@sample
samples.collections.Collections.Transformations.flatMapIndexed
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("flatMapIndexedSequence")
public fun <T, R>
Sequence<T>.flatMapIndexed(transform: (index: Int, T) -> Sequence<R>): Sequence<R> {
    return

```

```

flatMapIndexed(this, transform, Sequence<R>::iterator)\n\n\n**\n * Appends all elements yielded from results of
[transform] function being invoked on each element\n * and its index in the original sequence, to the given
[destination].\n *\n * The operation is _terminal_.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterableTo")\n@kotlin.internal.InlineOnly\npubli
c inline fun <T, R, C : MutableCollection<in R>> Sequence<T>.flatMapIndexedTo(destination: C, transform:
(index: Int, T) -> Iterable<R>): C {\n    var index = 0\n    for (element in this) {\n        val list =
transform(checkIndexOverflow(index++), element)\n        destination.addAll(list)\n    }\n    return
destination\n}\n\n**\n * Appends all elements yielded from results of [transform] function being invoked on each
element\n * and its index in the original sequence, to the given [destination].\n *\n * The operation is _terminal_.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedSequenceTo")\n@kotlin.internal.InlineOnly\npub
lic inline fun <T, R, C : MutableCollection<in R>> Sequence<T>.flatMapIndexedTo(destination: C, transform:
(index: Int, T) -> Sequence<R>): C {\n    var index = 0\n    for (element in this) {\n        val list =
transform(checkIndexOverflow(index++), element)\n        destination.addAll(list)\n    }\n    return
destination\n}\n\n**\n * Appends all elements yielded from results of [transform] function being invoked on each
element of original sequence, to the given [destination].\n *\n * The operation is _terminal_.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIterableTo")\npublic inline fun <T, R, C :
MutableCollection<in R>> Sequence<T>.flatMapTo(destination: C, transform: (T) -> Iterable<R>): C {\n    for
(element in this) {\n        val list = transform(element)\n        destination.addAll(list)\n    }\n    return
destination\n}\n\n**\n * Appends all elements yielded from results of [transform] function being invoked on each
element of original sequence, to the given [destination].\n *\n * The operation is _terminal_.\n *\npublic inline fun
<T, R, C : MutableCollection<in R>> Sequence<T>.flatMapTo(destination: C, transform: (T) -> Sequence<R>): C
{\n    for (element in this) {\n        val list = transform(element)\n        destination.addAll(list)\n    }\n    return
destination\n}\n\n**\n * Groups elements of the original sequence by the key returned by the given [keySelector]
function\n * applied to each element and returns a map where each group key is associated with a list of
corresponding elements.\n *\n * The returned map preserves the entry iteration order of the keys produced from the
original sequence.\n *\n * The operation is _terminal_.\n *\n * @sample
samples.collections.Collections.Transformations.groupBy\n *\npublic inline fun <T, K>
Sequence<T>.groupBy(keySelector: (T) -> K): Map<K, List<T>> {\n    return groupByTo(LinkedHashMap<K,
MutableList<T>>(), keySelector)\n}\n\n**\n * Groups values returned by the [valueTransform] function applied to
each element of the original sequence\n * by the key returned by the given [keySelector] function applied to the
element\n * and returns a map where each group key is associated with a list of corresponding values.\n *\n * The
returned map preserves the entry iteration order of the keys produced from the original sequence.\n *\n * The
operation is _terminal_.\n *\n * @sample
samples.collections.Collections.Transformations.groupByKeysAndValues\n *\npublic inline fun <T, K, V>
Sequence<T>.groupBy(keySelector: (T) -> K, valueTransform: (T) -> V): Map<K, List<V>> {\n    return
groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n**\n * Groups elements
of the original sequence by the key returned by the given [keySelector] function\n * applied to each element and
puts to the [destination] map each group key associated with a list of corresponding elements.\n *\n * @return The
[destination] map.\n *\n * The operation is _terminal_.\n *\n * @sample
samples.collections.Collections.Transformations.groupBy\n *\npublic inline fun <T, K, M : MutableMap<in K,
MutableList<T>>> Sequence<T>.groupByTo(destination: M, keySelector: (T) -> K): M {\n    for (element in this)
{\n        val key = keySelector(element)\n        val list = destination.getOrPut(key) { ArrayList<T>() }\n
list.add(element)\n    }\n    return destination\n}\n\n**\n * Groups values returned by the [valueTransform] function
applied to each element of the original sequence\n * by the key returned by the given [keySelector] function applied
to the element\n * and puts to the [destination] map each group key associated with a list of corresponding values.\n

```

```

 * \n * @return The [destination] map.\n * \n * The operation is _terminal_.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeyAndValues\n * \n\npublic inline fun <T, K, V, M :
MutableMap<in K, MutableList<V>>> Sequence<T>.groupByTo(destination: M, keySelector: (T) -> K,
valueTransform: (T) -> V): M {\n    for (element in this) {\n        val key = keySelector(element)\n        val list =
destination.getOrPut(key) { ArrayList<V>() }\n        list.add(valueTransform(element))\n    }\n    return
destination\n}\n\n/**\n * Creates a [Grouping] source from a sequence to be used later with one of group-and-fold
operations\n * using the specified [keySelector] function to extract a key from each element.\n * \n * The operation is
_intermediate_ and _stateless_.\n * \n * @sample samples.collections.Grouping.groupingByEachCount\n
*\n\n@SinceKotlin("1.1")\npublic inline fun <T, K> Sequence<T>.groupingBy(crossinline keySelector: (T) -> K):
Grouping<T, K> {\n    return object : Grouping<T, K> {\n        override fun sourceIterator(): Iterator<T> =
this@groupingBy.iterator()\n        override fun keyOf(element: T): K = keySelector(element)\n    }\n}\n\n/**\n *
Returns a sequence containing the results of applying the given [transform] function\n * to each element in the
original sequence.\n * \n * The operation is _intermediate_ and _stateless_.\n * \n * @sample
samples.collections.Collections.Transformations.map\n * \n\npublic fun <T, R> Sequence<T>.map(transform: (T) ->
R): Sequence<R> {\n    return TransformingSequence(this, transform)\n}\n\n/**\n * Returns a sequence containing
the results of applying the given [transform] function\n * to each element and its index in the original sequence.\n *
@param [transform] function that takes the index of an element and the element itself\n * and returns the result of
the transform applied to the element.\n * \n * The operation is _intermediate_ and _stateless_.\n * \n\npublic fun <T,
R> Sequence<T>.mapIndexed(transform: (index: Int, T) -> R): Sequence<R> {\n    return
TransformingIndexedSequence(this, transform)\n}\n\n/**\n * Returns a sequence containing only the non-null
results of applying the given [transform] function\n * to each element and its index in the original sequence.\n *
@param [transform] function that takes the index of an element and the element itself\n * and returns the result of
the transform applied to the element.\n * \n * The operation is _intermediate_ and _stateless_.\n * \n\npublic fun <T, R
: Any> Sequence<T>.mapIndexedNotNull(transform: (index: Int, T) -> R?): Sequence<R> {\n    return
TransformingIndexedSequence(this, transform).filterNotNull()\n}\n\n/**\n * Applies the given [transform] function
to each element and its index in the original sequence\n * and appends only the non-null results to the given
[destination].\n * @param [transform] function that takes the index of an element and the element itself\n * and
returns the result of the transform applied to the element.\n * \n * The operation is _terminal_.\n * \n\npublic inline fun
<T, R : Any, C : MutableCollection<in R>> Sequence<T>.mapIndexedNotNullTo(destination: C, transform: (index:
Int, T) -> R?): C {\n    forEachIndexed { index, element -> transform(index, element)?.let { destination.add(it) } }\n
return destination\n}\n\n/**\n * Applies the given [transform] function to each element and its index in the original
sequence\n * and appends the results to the given [destination].\n * @param [transform] function that takes the
index of an element and the element itself\n * and returns the result of the transform applied to the element.\n *
\n * The operation is _terminal_.\n * \n\npublic inline fun <T, R, C : MutableCollection<in R>>
Sequence<T>.mapIndexedTo(destination: C, transform: (index: Int, T) -> R): C {\n    var index = 0\n    for (item in
this)\n        destination.add(transform(checkIndexOverflow(index++), item))\n    return destination\n}\n\n/**\n * Returns a sequence containing only the non-null results of applying the given [transform] function\n * to each
element in the original sequence.\n * \n * The operation is _intermediate_ and _stateless_.\n * \n * @sample
samples.collections.Collections.Transformations.mapNotNull\n * \n\npublic fun <T, R : Any>
Sequence<T>.mapNotNull(transform: (T) -> R?): Sequence<R> {\n    return TransformingSequence(this,
transform).filterNotNull()\n}\n\n/**\n * Applies the given [transform] function to each element in the original
sequence\n * and appends only the non-null results to the given [destination].\n * \n * The operation is _terminal_.\n *
\n\npublic inline fun <T, R : Any, C : MutableCollection<in R>> Sequence<T>.mapNotNullTo(destination: C,
transform: (T) -> R?): C {\n    forEach { element -> transform(element)?.let { destination.add(it) } }\n    return
destination\n}\n\n/**\n * Applies the given [transform] function to each element of the original sequence\n * and
appends the results to the given [destination].\n * \n * The operation is _terminal_.\n * \n\npublic inline fun <T, R, C :
MutableCollection<in R>> Sequence<T>.mapTo(destination: C, transform: (T) -> R): C {\n    for (item in this)\n
destination.add(transform(item))\n    return destination\n}\n\n/**\n * Returns a sequence that wraps each element of

```

the original sequence into an [IndexedValue] containing the index of that element and the element itself.

The operation is `_intermediate_ and _stateless_`.

```
Sequence<IndexedValue<T>> {
    return IndexingSequence(this)
}
```

Returns a sequence containing only distinct elements from the given sequence. Among equal elements of the given sequence, only the first one will be present in the resulting sequence. The elements in the resulting sequence are in the same order as they were in the source sequence.

The operation is `_intermediate_ and _stateful_`.

```
Sequence<T>.distinct(): Sequence<T> {
    return this.distinctBy { it }
}
```

Returns a sequence containing only elements from the given sequence having distinct keys returned by the given [selector] function. Among elements of the given sequence with equal keys, only the first one will be present in the resulting sequence. The elements in the resulting sequence are in the same order as they were in the source sequence.

The operation is `_intermediate_ and _stateful_`.

```
Sequence<T>.distinctBy(selector: (T) -> K): Sequence<T> {
    return DistinctSequence(this, selector)
}
```

Returns a new [MutableSet] containing all distinct elements from the given sequence. The returned set preserves the element iteration order of the original sequence.

The operation is `_terminal_`.

```
Sequence<T>.toMutableSet(): MutableSet<T> {
    val set = LinkedHashSet<T>()
    for (item in this)
        set.add(item)
    return set
}
```

Returns `true` if all elements match the given [predicate].

The operation is `_terminal_`.

```
Sequence<T>.all(predicate: (T) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}
```

Returns `true` if sequence has at least one element.

The operation is `_terminal_`.

```
Sequence<T>.any(): Boolean {
    return iterator().hasNext()
}
```

Returns `true` if at least one element matches the given [predicate].

The operation is `_terminal_`.

```
Sequence<T>.any(predicate: (T) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return true
    return false
}
```

Returns the number of elements in this sequence.

The operation is `_terminal_`.

```
Sequence<T>.count(): Int {
    var count = 0
    for (element in this)
        checkCountOverflow(++count)
    return count
}
```

Returns the number of elements matching the given [predicate].

The operation is `_terminal_`.

```
Sequence<T>.count(predicate: (T) -> Boolean): Int {
    var count = 0
    for (element in this) if (predicate(element)) checkCountOverflow(++count)
    return count
}
```

Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element. Returns the specified [initial] value if the sequence is empty.

@param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

The operation is `_terminal_`.

```
Sequence<T>.fold(initial: R, operation: (acc: R, T) -> R): R {
    var accumulator = initial
    for (element in this)
        accumulator = operation(accumulator, element)
    return accumulator
}
```

Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element with its index in the original sequence. Returns the specified [initial] value if the sequence is empty.

@param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.

The operation is `_terminal_`.

```
Sequence<T>.foldIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): R {
    var index = 0
    var accumulator = initial
    for (element in this)
        accumulator = operation(checkIndexOverflow(index++), accumulator, element)
    return accumulator
}
```

Performs the given [action] on each element.

The operation is `_terminal_`.

```
Sequence<T>.forEach(action: (T) -> Unit): Unit {
    for (element in this)
        action(element)
}
```

Performs the given [action] on each element, providing sequential index with the element.

@param [action] function that takes the index of an element and the element itself and performs the action on the element.

The operation is `_terminal_`.

```
Sequence<T>.forEachIndexed(action: (index: Int, element: T) -> Unit): Unit {
    for (index, element in this)
        action(index, element)
}
```

```

Sequence<T>.forEachIndexed(action: (index: Int, T) -> Unit): Unit {
    var index = 0
    for (item in this)
        action(checkIndexOverflow(index++), item)
}
@Deprecated("Use maxOrNull instead.",
ReplaceWith("this.maxOrNull()"))
@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5",
hiddenSince = "1.6")
@SinceKotlin("1.1")
public fun Sequence<Double>.max(): Double? {
    return
    maxOrNull()
}
@Deprecated("Use maxOrNull instead.",
ReplaceWith("this.maxOrNull()"))
@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5",
hiddenSince = "1.6")
@SinceKotlin("1.1")
public fun Sequence<Float>.max(): Float? {
    return
    maxOrNull()
}
@Deprecated("Use maxOrNull instead.",
ReplaceWith("this.maxOrNull()"))
@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5",
hiddenSince = "1.6")
public fun <T : Comparable<T>> Sequence<T>.max(): T? {
    return
    maxOrNull()
}
@Deprecated("Use maxByOrNull instead.",
ReplaceWith("this.maxByOrNull(selector)"))
@DeprecatedSinceKotlin(warningSince = "1.4", errorSince =
"1.5", hiddenSince = "1.6")
public inline fun <T, R : Comparable<R>> Sequence<T>.maxBy(selector: (T) ->
R): T? {
    return maxByOrNull(selector)
}
/**
 * Returns the first element yielding the largest value of the
 * given function or `null` if there are no elements.
 *
 * The operation is _terminal_.
 *
 * @sample
 * samples.collections.Collections.Aggregates.maxByOrNull
 *
 * @SinceKotlin("1.4")
 * public inline fun <T, R :
 * Comparable<R>> Sequence<T>.maxByOrNull(selector: (T) -> R): T? {
 *     val iterator = iterator()
 *     if (!iterator.hasNext()) return null
 *     var maxElem = iterator.next()
 *     if (!iterator.hasNext()) return maxElem
 *     var
 *     maxV = selector(maxElem)
 *     do {
 *         val e = iterator.next()
 *         val v = selector(e)
 *         if (maxV <
 * v) {
 *             maxElem = e
 *             maxV = v
 *         }
 *     } while (iterator.hasNext())
 *     return
 *     maxElem
 * }
 *
 * Returns the largest value among all values produced by [selector] function
 * applied to
 * each element in the sequence.
 *
 * If any of values produced by [selector] function is `NaN`, the returned result
 * is `NaN`.
 *
 * @throws NoSuchElementException if the sequence is empty.
 *
 * The operation is _terminal_.
 *
 *
 * @SinceKotlin("1.4")
 * @OptIn(kotlin.experimental.ExperimentalTypeInference::class)
 * @OverloadResolution
 * ByLambdaReturnType
 * @kotlin.internal.InlineOnly
 * public inline fun <T> Sequence<T>.maxOf(selector: (T) ->
 * Double): Double {
 *     val iterator = iterator()
 *     if (!iterator.hasNext()) throw NoSuchElementException()
 *     var
 *     maxV = selector(iterator.next())
 *     while (iterator.hasNext()) {
 *         val v = selector(iterator.next())
 *         maxV = maxOf(maxV, v)
 *     }
 *     return maxV
 * }
 *
 * Returns the largest value among all
 * values produced by [selector] function
 * applied to each element in the sequence.
 *
 * If any of values
 * produced by [selector] function is `NaN`, the returned result is `NaN`.
 *
 * @throws NoSuchElementException
 * if the sequence is empty.
 *
 * The operation is _terminal_.
 *
 *
 * @SinceKotlin("1.4")
 * @OptIn(kotlin.experimental.ExperimentalTypeInference::class)
 * @OverloadResolution
 * ByLambdaReturnType
 * @kotlin.internal.InlineOnly
 * public inline fun <T> Sequence<T>.maxOf(selector: (T) ->
 * Float): Float {
 *     val iterator = iterator()
 *     if (!iterator.hasNext()) throw NoSuchElementException()
 *     var
 *     maxV = selector(iterator.next())
 *     while (iterator.hasNext()) {
 *         val v = selector(iterator.next())
 *         maxV = maxOf(maxV, v)
 *     }
 *     return maxV
 * }
 *
 * Returns the largest value among all
 * values produced by [selector] function
 * applied to each element in the sequence.
 *
 * @throws
 * NoSuchElementException if the sequence is empty.
 *
 * The operation is _terminal_.
 *
 *
 * @SinceKotlin("1.4")
 * @OptIn(kotlin.experimental.ExperimentalTypeInference::class)
 * @OverloadResolution
 * ByLambdaReturnType
 * @kotlin.internal.InlineOnly
 * public inline fun <T, R : Comparable<R>>
 * Sequence<T>.maxOf(selector: (T) -> R): R {
 *     val iterator = iterator()
 *     if (!iterator.hasNext()) throw
 *     NoSuchElementException()
 *     var maxV = selector(iterator.next())
 *     while (iterator.hasNext()) {
 *         val v
 * = selector(iterator.next())
 *         if (maxV < v) {
 *             maxV = v
 *         }
 *     }
 *     return
 *     maxV
 * }
 *
 * Returns the largest value among all values produced by [selector] function
 * applied to
 * each element in the sequence or `null` if there are no elements.
 *
 * If any of values produced by [selector]
 * function is `NaN`, the returned result is `NaN`.
 *
 * The operation is _terminal_.
 *
 *
 * @SinceKotlin("1.4")
 * @OptIn(kotlin.experimental.ExperimentalTypeInference::class)
 * @OverloadResolution

```

```

ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.maxOrNull(selector:
(T) -> Double): Double? {\n    val iterator = iterator()\n    if (!iterator.hasNext()) return null\n    var max Value =
selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v = selector(iterator.next())\n        max Value =
maxOf(max Value, v)\n    }\n    return max Value\n}\n\n/**\n * Returns the largest value among all values produced
by [selector] function\n * applied to each element in the sequence or `null` if there are no elements.\n * \n * If any of
values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * The operation is _terminal_.\n
*\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.maxOrNull(selector:
(T) -> Float): Float? {\n    val iterator = iterator()\n    if (!iterator.hasNext()) return null\n    var max Value =
selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v = selector(iterator.next())\n        max Value =
maxOf(max Value, v)\n    }\n    return max Value\n}\n\n/**\n * Returns the largest value among all values produced
by [selector] function\n * applied to each element in the sequence or `null` if there are no elements.\n * \n * The
operation is _terminal_.\n
*\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Comparable<R>>
Sequence<T>.maxOrNull(selector: (T) -> R): R? {\n    val iterator = iterator()\n    if (!iterator.hasNext()) return
null\n    var max Value = selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v =
selector(iterator.next())\n        if (max Value < v) {\n            max Value = v\n        }\n    }\n    return
max Value\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values
produced by [selector] function applied to each element in the sequence.\n * \n * @throws
NoSuchElementException if the sequence is empty.\n * \n * The operation is _terminal_.\n
*\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R>
Sequence<T>.maxOfWith(comparator: Comparator<in R>, selector: (T) -> R): R {\n    val iterator = iterator()\n    if
(!iterator.hasNext()) throw NoSuchElementException()\n    var max Value = selector(iterator.next())\n    while
(iterator.hasNext()) {\n        val v = selector(iterator.next())\n        if (comparator.compare(max Value, v) < 0) {\n
            max Value = v\n        }\n    }\n    return max Value\n}\n\n/**\n * Returns the largest value according to the provided
[comparator]\n * among all values produced by [selector] function applied to each element in the sequence or `null`
if there are no elements.\n * \n * The operation is _terminal_.\n
*\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R>
Sequence<T>.maxOfWithOrNull(comparator: Comparator<in R>, selector: (T) -> R): R? {\n    val iterator =
iterator()\n    if (!iterator.hasNext()) return null\n    var max Value = selector(iterator.next())\n    while
(iterator.hasNext()) {\n        val v = selector(iterator.next())\n        if (comparator.compare(max Value, v) < 0) {\n
            max Value = v\n        }\n    }\n    return max Value\n}\n\n/**\n * Returns the largest element or `null` if there are no
elements.\n * \n * If any of elements is `NaN` returns `NaN`.\n * \n * The operation is _terminal_.\n
*\n*\n@SinceKotlin("1.4")\npublic fun Sequence<Double>.maxOrNull(): Double? {\n    val iterator = iterator()\n    if
(!iterator.hasNext()) return null\n    var max = iterator.next()\n    while (iterator.hasNext()) {\n        val e =
iterator.next()\n        max = maxOf(max, e)\n    }\n    return max\n}\n\n/**\n * Returns the largest element or `null` if
there are no elements.\n * \n * If any of elements is `NaN` returns `NaN`.\n * \n * The operation is _terminal_.\n
*\n*\n@SinceKotlin("1.4")\npublic fun Sequence<Float>.maxOrNull(): Float? {\n    val iterator = iterator()\n    if
(!iterator.hasNext()) return null\n    var max = iterator.next()\n    while (iterator.hasNext()) {\n        val e =
iterator.next()\n        max = maxOf(max, e)\n    }\n    return max\n}\n\n/**\n * Returns the largest element or `null` if
there are no elements.\n * \n * The operation is _terminal_.\n
*\n*\n@SinceKotlin("1.4")\npublic fun <T :
Comparable<T>> Sequence<T>.maxOrNull(): T? {\n    val iterator = iterator()\n    if (!iterator.hasNext()) return
null\n    var max = iterator.next()\n    while (iterator.hasNext()) {\n        val e = iterator.next()\n        if (max < e) max
= e\n    }\n    return max\n}\n\n@Deprecated("Use maxWithOrNull instead.",
ReplaceWith("this.maxWithOrNull(comparator)"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince

```

```

= \"1.5\", hiddenSince = \"1.6\")\npublic fun <T> Sequence<T>.maxWith(comparator: Comparator<in T>): T? {\n
return maxOrNull(comparator)\n}\n\n/**\n * Returns the first element having the largest value according to the
provided [comparator] or `null` if there are no elements.\n * \n * The operation is _terminal_.\n
*/\n\n@SinceKotlin(\"1.4\")\npublic fun <T> Sequence<T>.maxOrNull(comparator: Comparator<in T>): T? {\n
val iterator = iterator()\n if (!iterator.hasNext()) return null\n var max = iterator.next()\n while
(iterator.hasNext()) {\n val e = iterator.next()\n if (comparator.compare(max, e) < 0) max = e\n }\n
return max\n}\n\n@Deprecated(\"Use minOrNull instead.\",
ReplaceWith(\"this.minOrNull()\"))\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince = \"1.5\",
hiddenSince = \"1.6\")\n@SinceKotlin(\"1.1\")\npublic fun Sequence<Double>.min(): Double? {\n return
minOrNull()\n}\n\n@Deprecated(\"Use minOrNull instead.\",
ReplaceWith(\"this.minOrNull()\"))\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince = \"1.5\",
hiddenSince = \"1.6\")\n@SinceKotlin(\"1.1\")\npublic fun Sequence<Float>.min(): Float? {\n return
minOrNull()\n}\n\n@Deprecated(\"Use minOrNull instead.\",
ReplaceWith(\"this.minOrNull()\"))\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince = \"1.5\",
hiddenSince = \"1.6\")\npublic fun <T : Comparable<T>> Sequence<T>.min(): T? {\n return
minOrNull()\n}\n\n@Deprecated(\"Use minByOrNull instead.\",
ReplaceWith(\"this.minByOrNull(selector)\"))\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince =
\"1.5\", hiddenSince = \"1.6\")\npublic inline fun <T, R : Comparable<R>> Sequence<T>.minBy(selector: (T) ->
R): T? {\n return minByOrNull(selector)\n}\n\n/**\n * Returns the first element yielding the smallest value of the
given function or `null` if there are no elements.\n * \n * The operation is _terminal_.\n * \n * @sample
samples.collections.Collections.Aggregates.minByOrNull\n * \n * @SinceKotlin(\"1.4\")\npublic inline fun <T, R :
Comparable<R>> Sequence<T>.minByOrNull(selector: (T) -> R): T? {\n val iterator = iterator()\n if
(!iterator.hasNext()) return null\n var minElem = iterator.next()\n if (!iterator.hasNext()) return minElem\n var
minValue = selector(minElem)\n do {\n val e = iterator.next()\n val v = selector(e)\n if (minValue >
v) {\n minElem = e\n minValue = v\n }\n } while (iterator.hasNext())\n return
minElem\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to
each element in the sequence.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result
is `NaN`.\n * \n * @throws NoSuchElementException if the sequence is empty.\n * \n * The operation is
_terminal_.\n
*/\n\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.minOf(selector: (T) ->
Double): Double {\n val iterator = iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var
minValue = selector(iterator.next())\n while (iterator.hasNext()) {\n val v = selector(iterator.next())\n
minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all
values produced by [selector] function\n * applied to each element in the sequence.\n * \n * If any of values
produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException
if the sequence is empty.\n * \n * The operation is _terminal_.\n
*/\n\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.minOf(selector: (T) ->
Float): Float {\n val iterator = iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var
minValue = selector(iterator.next())\n while (iterator.hasNext()) {\n val v = selector(iterator.next())\n
minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all
values produced by [selector] function\n * applied to each element in the sequence.\n * \n * @throws
NoSuchElementException if the sequence is empty.\n * \n * The operation is _terminal_.\n
*/\n\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Comparable<R>>
Sequence<T>.minOf(selector: (T) -> R): R {\n val iterator = iterator()\n if (!iterator.hasNext()) throw
NoSuchElementException()\n var minValue = selector(iterator.next())\n while (iterator.hasNext()) {\n val v

```

```

= selector(iterator.next())\n    if (minValue > v) {\n        minValue = v\n    }\n    }\n    return
minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the sequence or `null` if there are no elements.\n * \n * If any of values produced by [selector]
function is `NaN`, the returned result is `NaN`.\n * \n * The operation is _terminal_.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.minOrNull(selector:
(T) -> Double): Double? {\n    val iterator = iterator()\n    if (!iterator.hasNext()) return null\n    var minValue =
selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v = selector(iterator.next())\n        minValue =
minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all values produced
by [selector] function\n * applied to each element in the sequence or `null` if there are no elements.\n * \n * If any of
values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * The operation is _terminal_.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.minOrNull(selector:
(T) -> Float): Float? {\n    val iterator = iterator()\n    if (!iterator.hasNext()) return null\n    var minValue =
selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v = selector(iterator.next())\n        minValue =
minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all values produced
by [selector] function\n * applied to each element in the sequence or `null` if there are no elements.\n * \n * The
operation is _terminal_.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Comparable<R>>
Sequence<T>.minOrNull(selector: (T) -> R): R? {\n    val iterator = iterator()\n    if (!iterator.hasNext()) return
null\n    var minValue = selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v =
selector(iterator.next())\n        if (minValue > v) {\n            minValue = v\n        }\n    }\n    return
minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values
produced by [selector] function applied to each element in the sequence.\n * \n * @throws
NoSuchElementException if the sequence is empty.\n * \n * The operation is _terminal_.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R>
Sequence<T>.minOfWith(comparator: Comparator<in R>, selector: (T) -> R): R {\n    val iterator = iterator()\n    if
(!iterator.hasNext()) throw NoSuchElementException()\n    var minValue = selector(iterator.next())\n    while
(iterator.hasNext()) {\n        val v = selector(iterator.next())\n        if (comparator.compare(minValue, v) > 0) {\n
            minValue = v\n        }\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value according to the
provided [comparator]\n * among all values produced by [selector] function applied to each element in the sequence
or `null` if there are no elements.\n * \n * The operation is _terminal_.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R>
Sequence<T>.minOfWithOrNull(comparator: Comparator<in R>, selector: (T) -> R): R? {\n    val iterator =
iterator()\n    if (!iterator.hasNext()) return null\n    var minValue = selector(iterator.next())\n    while
(iterator.hasNext()) {\n        val v = selector(iterator.next())\n        if (comparator.compare(minValue, v) > 0) {\n
            minValue = v\n        }\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest element or `null` if there are
no elements.\n * \n * If any of elements is `NaN` returns `NaN`.\n * \n * The operation is _terminal_.\n
*\n@SinceKotlin("1.4")\npublic fun Sequence<Double>.minOrNull(): Double? {\n    val iterator = iterator()\n    if
(!iterator.hasNext()) return null\n    var min = iterator.next()\n    while (iterator.hasNext()) {\n        val e =
iterator.next()\n        min = minOf(min, e)\n    }\n    return min\n}\n\n/**\n * Returns the smallest element or `null` if
there are no elements.\n * \n * If any of elements is `NaN` returns `NaN`.\n * \n * The operation is _terminal_.\n
*\n@SinceKotlin("1.4")\npublic fun Sequence<Float>.minOrNull(): Float? {\n    val iterator = iterator()\n    if
(!iterator.hasNext()) return null\n    var min = iterator.next()\n    while (iterator.hasNext()) {\n        val e =
iterator.next()\n        min = minOf(min, e)\n    }\n    return min\n}\n\n/**\n * Returns the smallest element or `null` if

```

there are no elements.

```

 * The operation is _terminal_.
 * Since Kotlin("1.4")
 public fun <T> : Comparable<T>> Sequence<T>.minOrNull(): T? {
     val iterator = iterator()
     if (!iterator.hasNext()) return null
     var min = iterator.next()
     while (iterator.hasNext()) {
         val e = iterator.next()
         if (min > e) min = e
     }
     return min
 }
 @Deprecated("Use minWithOrNull instead.")
 ReplaceWith("this.minWithOrNull(comparator)")
 @DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5", hiddenSince = "1.6")
 public fun <T> Sequence<T>.minWith(comparator: Comparator<in T>): T? {
     return minWithOrNull(comparator)
 }
 * Returns the first element having the smallest value according to the provided [comparator] or `null` if there are no elements.
 * The operation is _terminal_.
 * Since Kotlin("1.4")
 public fun <T> Sequence<T>.minWithOrNull(comparator: Comparator<in T>): T? {
     val iterator = iterator()
     if (!iterator.hasNext()) return null
     var min = iterator.next()
     while (iterator.hasNext()) {
         val e = iterator.next()
         if (comparator.compare(min, e) > 0) min = e
     }
     return min
 }
 * Returns `true` if the sequence has no elements.
 * The operation is _terminal_.
 * @sample samples.collections.Collections.Aggregates.none
 * public fun <T> Sequence<T>.none(): Boolean {
     return iterator().hasNext()
 }
 * Returns `true` if no elements match the given [predicate].
 * The operation is _terminal_.
 * @sample samples.collections.Collections.Aggregates.noneWithPredicate
 * public inline fun <T> Sequence<T>.none(predicate: (T) -> Boolean): Boolean {
     for (element in this) if (predicate(element)) return false
     return true
 }
 * Returns a sequence which performs the given [action] on each element of the original sequence as they pass through it.
 * The operation is _intermediate_ and _stateless_.
 * Since Kotlin("1.1")
 public fun <T> Sequence<T>.onEach(action: (T) -> Unit): Sequence<T> {
     return map {
         action(it)
         it
     }
 }
 * Returns a sequence which performs the given [action] on each element of the original sequence as they pass through it.
 * @param [action] function that takes the index of an element and the element itself
 * and performs the action on the element.
 * The operation is _intermediate_ and _stateless_.
 * Since Kotlin("1.4")
 public fun <T> Sequence<T>.onEachIndexed(action: (index: Int, T) -> Unit): Sequence<T> {
     return mapIndexed { index, element ->
         action(index, element)
         element
     }
 }
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element.
 * Throws an exception if this sequence is empty. If the sequence can be empty in an expected way,
 * please use [reduceOrNull] instead. It returns `null` when its receiver is empty.
 * @param [operation] function that takes current accumulator value and an element,
 * and calculates the next accumulator value.
 * The operation is _terminal_.
 * @sample samples.collections.Collections.Aggregates.reduce
 * public inline fun <S, T : S> Sequence<T>.reduce(operation: (acc: S, T) -> S): S {
     val iterator = this.iterator()
     if (!iterator.hasNext()) throw UnsupportedOperationException("Empty sequence can't be reduced.")
     var accumulator: S = iterator.next()
     while (iterator.hasNext()) {
         accumulator = operation(accumulator, iterator.next())
     }
     return accumulator
 }
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element with its index in the original sequence.
 * Throws an exception if this sequence is empty. If the sequence can be empty in an expected way,
 * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.
 * @param [operation] function that takes the index of an element, current accumulator value and the element itself,
 * and calculates the next accumulator value.
 * The operation is _terminal_.
 * @sample samples.collections.Collections.Aggregates.reduce
 * public inline fun <S, T : S> Sequence<T>.reduceIndexed(operation: (index: Int, acc: S, T) -> S): S {
     val iterator = this.iterator()
     if (!iterator.hasNext()) throw UnsupportedOperationException("Empty sequence can't be reduced.")
     var index = 1
     var accumulator: S = iterator.next()
     while (iterator.hasNext()) {
         accumulator = operation(checkIndexOverflow(index++), accumulator, iterator.next())
     }
     return accumulator
 }
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element with its index in the original sequence.
 * Returns `null` if the sequence is empty.
 * @param [operation] function that takes the index of an element, current accumulator value and the element itself,
 * and calculates the next accumulator value.
 * The operation is _terminal_.
 * @sample

```

```

samples.collections.Collections.Aggregates.reduceOrNull\n * \n @SinceKotlin("1.4")\n public inline fun <S, T : S>  

Sequence<T>.reduceIndexedOrNull(operation: (index: Int, acc: S, T) -> S): S? {\n   val iterator = this.iterator()\n  

if (!iterator.hasNext()) return null\n   var index = 1\n   var accumulator: S = iterator.next()\n   while  

(iterator.hasNext()) {\n     accumulator = operation(checkIndexOverflow(index++), accumulator, iterator.next())\n  

  }\n   return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation]  

from left to right\n * to current accumulator value and each element.\n * \n * Returns `null` if the sequence is  

empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates  

the next accumulator value.\n * \n * The operation is _terminal_.\n * \n * @sample  

samples.collections.Collections.Aggregates.reduceOrNull\n  

*\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n public inline fun <S, T : S>  

Sequence<T>.reduceOrNull(operation: (acc: S, T) -> S): S? {\n   val iterator = this.iterator()\n   if  

(!iterator.hasNext()) return null\n   var accumulator: S = iterator.next()\n   while (iterator.hasNext()) {\n  

accumulator = operation(accumulator, iterator.next())\n   }\n   return accumulator\n}\n\n/**\n * Returns a sequence  

containing successive accumulation values generated by applying [operation] from left to right\n * to each element  

and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation]  

function should not be mutated;\n * otherwise it would affect the previous value in resulting sequence.\n * The  

[initial] value should also be immutable (or should not be mutated)\n * as it may be passed to [operation] function  

later because of sequence's lazy nature.\n * \n * @param [operation] function that takes current accumulator value  

and an element,  

and calculates the next accumulator value.\n * \n * The operation is _intermediate_ and  

_stateless_.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n  

*\n @SinceKotlin("1.4")\n public fun <T, R> Sequence<T>.runningFold(initial: R, operation: (acc: R, T) -> R):  

Sequence<R> {\n   return sequence {\n     yield(initial)\n     var accumulator = initial\n     for (element in  

this@runningFold) {\n       accumulator = operation(accumulator, element)\n       yield(accumulator)\n     }\n  

  }\n}\n\n/**\n * Returns a sequence containing successive accumulation values generated by applying [operation]  

from left to right\n * to each element, its index in the original sequence and current accumulator value that starts  

with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n *  

otherwise it would affect the previous value in resulting sequence.\n * The [initial] value should also be immutable  

(or should not be mutated)\n * as it may be passed to [operation] function later because of sequence's lazy nature.\n  

*\n * @param [operation] function that takes the index of an element, current accumulator value\n * and the  

element itself, and calculates the next accumulator value.\n * \n * The operation is _intermediate_ and _stateless_.\n  

*\n * @sample samples.collections.Collections.Aggregates.runningFold\n *\n @SinceKotlin("1.4")\n public fun  

<T, R> Sequence<T>.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): Sequence<R> {\n  

return sequence {\n   yield(initial)\n   var index = 0\n   var accumulator = initial\n   for (element in  

this@runningFoldIndexed) {\n     accumulator = operation(checkIndexOverflow(index++), accumulator,  

element)\n     yield(accumulator)\n   }\n }\n}\n\n/**\n * Returns a sequence containing successive  

accumulation values generated by applying [operation] from left to right\n * to each element and current  

accumulator value that starts with the first element of this sequence.\n * \n * Note that `acc` value passed to  

[operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting sequence.\n  

*\n * @param [operation] function that takes current accumulator value and the element, and calculates the next  

accumulator value.\n * \n * The operation is _intermediate_ and _stateless_.\n * \n * @sample  

samples.collections.Collections.Aggregates.runningReduce\n  

*\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n public fun <S, T : S>  

Sequence<T>.runningReduce(operation: (acc: S, T) -> S): Sequence<S> {\n   return sequence {\n     val iterator =  

iterator()\n     if (iterator.hasNext()) {\n       var accumulator: S = iterator.next()\n       yield(accumulator)\n  

while (iterator.hasNext()) {\n         accumulator = operation(accumulator, iterator.next())\n  

yield(accumulator)\n       }\n     }\n   }\n}\n\n/**\n * Returns a sequence containing successive accumulation  

values generated by applying [operation] from left to right\n * to each element, its index in the original sequence and  

current accumulator value that starts with the first element of this sequence.\n * \n * Note that `acc` value passed to

```

```

[operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting sequence.\n
 * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the
element itself, and calculates the next accumulator value.\n *\n * The operation is _intermediate_ and _stateless_.\n
 * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n * ^\n@SinceKotlin("1.4")\npublic fun
<S, T : S> Sequence<T>.runningReduceIndexed(operation: (index: Int, acc: S, T) -> S): Sequence<S> {\n  return
sequence {\n    val iterator = iterator()\n    if (iterator.hasNext()) {\n        var accumulator: S =
iterator.next()\n        yield(accumulator)\n        var index = 1\n        while (iterator.hasNext()) {\n
accumulator = operation(checkIndexOverflow(index++), accumulator, iterator.next())\n
yield(accumulator)\n        }\n    }\n}\n}\n}\n}\n\n/**\n * Returns a sequence containing successive accumulation
values generated by applying [operation] from left to right\n * to each element and current accumulator value that
starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n *
otherwise it would affect the previous value in resulting sequence.\n * The [initial] value should also be immutable
(or should not be mutated)\n * as it may be passed to [operation] function later because of sequence's lazy nature.\n
 * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next
accumulator value.\n *\n * The operation is _intermediate_ and _stateless_.\n * \n * @sample
samples.collections.Collections.Aggregates.scan\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun <T, R>
Sequence<T>.scan(initial: R, operation: (acc: R, T) -> R): Sequence<R> {\n  return runningFold(initial,
operation)\n}\n}\n}\n\n/**\n * Returns a sequence containing successive accumulation values generated by applying
[operation] from left to right\n * to each element, its index in the original sequence and current accumulator value
that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n
 * otherwise it would affect the previous value in resulting sequence.\n * The [initial] value should also be immutable
(or should not be mutated)\n * as it may be passed to [operation] function later because of sequence's lazy nature.\n
 * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the
element itself, and calculates the next accumulator value.\n *\n * The operation is _intermediate_ and _stateless_.\n
 * \n * @sample samples.collections.Collections.Aggregates.scan\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun <T, R>
Sequence<T>.scanIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): Sequence<R> {\n  return
runningFoldIndexed(initial, operation)\n}\n}\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the sequence.\n * \n * The operation is _terminal_.\n * ^\n@Deprecated("Use sumOf
instead.", ReplaceWith("this.sumOf(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic inline
fun <T> Sequence<T>.sumBy(selector: (T) -> Int): Int {\n  var sum: Int = 0\n  for (element in this) {\n    sum
+= selector(element)\n  }\n  return sum\n}\n}\n}\n\n/**\n * Returns the sum of all values produced by [selector]
function applied to each element in the sequence.\n * \n * The operation is _terminal_.\n * ^\n@Deprecated("Use
sumOf instead.", ReplaceWith("this.sumOf(selector)"))\n@DeprecatedSinceKotlin(warningSince =
"1.5")\npublic inline fun <T> Sequence<T>.sumByDouble(selector: (T) -> Double): Double {\n  var sum: Double
= 0.0\n  for (element in this) {\n    sum += selector(element)\n  }\n  return sum\n}\n}\n}\n\n/**\n * Returns the sum
of all values produced by [selector] function applied to each element in the sequence.\n * \n * The operation is
_terminal_.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfDouble")\n@kotlin.internal.InlineOnly\npublic inline fun
<T> Sequence<T>.sumOf(selector: (T) -> Double): Double {\n  var sum: Double = 0.toDouble()\n  for (element
in this) {\n    sum += selector(element)\n  }\n  return sum\n}\n}\n}\n\n/**\n * Returns the sum of all values produced
by [selector] function applied to each element in the sequence.\n * \n * The operation is _terminal_.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfInt")\n@kotlin.internal.InlineOnly\npublic inline fun <T>
Sequence<T>.sumOf(selector: (T) -> Int): Int {\n  var sum: Int = 0.toInt()\n  for (element in this) {\n    sum +=
selector(element)\n  }\n  return sum\n}\n}\n}\n\n/**\n * Returns the sum of all values produced by [selector] function

```

applied to each element in the sequence. \n * \n * The operation is `_terminal_`. \n

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfLong")\n@kotlin.internal.InlineOnly\npublic inline fun
<T> Sequence<T>.sumOf(selector: (T) -> Long): Long {\n    var sum: Long = 0.toLong()\n    for (element in this)
{\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the sequence.\n * \n * The operation is _terminal_. \n
```

```
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfUInt")\n@WasExperimental(ExperimentalUnsignedType
s::class)\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.sumOf(selector: (T) -> UInt): UInt {\n
var sum: UInt = 0.toUInt()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return
sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the
sequence.\n * \n * The operation is _terminal_. \n
```

```
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfULong")\n@WasExperimental(ExperimentalUnsignedType
s::class)\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.sumOf(selector: (T) -> ULong):
ULong {\n    var sum: ULong = 0.toULong()\n    for (element in this) {\n        sum += selector(element)\n    }\n
return sum\n}\n\n/**\n * Returns an original collection containing all the non-`null` elements, throwing an
[IllegalArgumentException] if there are any `null` elements.\n * \n * The operation is _intermediate_ and
_stateless_. \n\npublic fun <T : Any> Sequence<T?>.requireNonNulls(): Sequence<T> {\n    return map { it ?\n
throw IllegalArgumentException("null element found in $this.") }\n}\n\n/**\n * Splits this sequence into a
sequence of lists each not exceeding the given [size]. \n * \n * The last list in the resulting sequence may have fewer
elements than the given [size]. \n * \n * @param size the number of elements to take in each list, must be positive
and can be greater than the number of elements in this sequence.\n * \n * The operation is _intermediate_ and
_stateful_. \n * \n * @sample samples.collections.Collections.Transformations.chunked\n
```

```
*\n@SinceKotlin("1.2")\npublic fun <T> Sequence<T>.chunked(size: Int): Sequence<List<T>> {\n    return
windowed(size, size, partialWindows = true)\n}\n\n/**\n * Splits this sequence into several lists each not exceeding
the given [size] \n * and applies the given [transform] function to an each. \n * \n * @return sequence of results of the
[transform] applied to an each list. \n * \n * Note that the list passed to the [transform] function is ephemeral and is
valid only inside that function. \n * You should not store it or allow it to escape in some way, unless you made a
snapshot of it. \n * The last list may have fewer elements than the given [size]. \n * \n * @param size the number of
elements to take in each list, must be positive and can be greater than the number of elements in this sequence. \n * \n
* The operation is _intermediate_ and _stateful_. \n * \n * @sample samples.text.Strings.chunkedTransform\n
```

```
*\n@SinceKotlin("1.2")\npublic fun <T, R> Sequence<T>.chunked(size: Int, transform: (List<T>) -> R):
Sequence<R> {\n    return windowed(size, size, partialWindows = true, transform = transform)\n}\n\n/**\n * Returns a
sequence containing all elements of the original sequence without the first occurrence of the given
[element]. \n * \n * The operation is _intermediate_ and _stateless_. \n\npublic operator fun <T>
Sequence<T>.minus(element: T): Sequence<T> {\n    override fun iterator():
Iterator<T> {\n        var removed = false\n        return this@minus.filter { if (!removed && it == element) {\n
removed = true; false } else true }.iterator()\n    }\n}\n\n/**\n * Returns a sequence containing all elements
of original sequence except the elements contained in the given [elements] array. \n * \n * Note that the source
sequence and the array being subtracted are iterated only when an `iterator` is requested from \n * the resulting
sequence. Changing any of them between successive calls to `iterator` may affect the result. \n * \n * Before Kotlin
1.6, the [elements] array may have been converted to a [HashSet] to speed up the operation, thus the elements were
required to have \n * a correct and stable implementation of `hashCode()` that didn't change between successive
invocations. \n * On JVM, you can enable this behavior back with the system property
`kotlin.collections.convert_arg_to_set_in_removeAll` set to `true`. \n * \n * The operation is _intermediate_ and
_stateful_. \n\npublic operator fun <T> Sequence<T>.minus(elements: Array<out T>): Sequence<T> {\n    if
(elements.isEmpty()) return this\n    return object: Sequence<T> {\n        override fun iterator(): Iterator<T> {\n
```

```

    val other = elements.convertToSetForSetOperation()\n        return this@minus.filterNot { it in other }.iterator()\n
    }\n }\n}\n\n/**\n * Returns a sequence containing all elements of original sequence except the elements
contained in the given [elements] collection.\n * \n * Note that the source sequence and the collection being
subtracted are iterated only when an `iterator` is requested from\n * the resulting sequence. Changing any of them
between successive calls to `iterator` may affect the result.\n * \n * Before Kotlin 1.6, the [elements] collection may
have been converted to a [HashSet] to speed up the operation, thus the elements were required to have\n * a correct
and stable implementation of `hashCode()` that didn't change between successive invocations.\n * On JVM, you can
enable this behavior back with the system property `kotlin.collections.convert_arg_to_set_in_removeAll` set to
`true`.\n * \n * The operation is _intermediate_ and _stateful_.\n */\n\npublic operator fun <T>
Sequence<T>.minus(elements: Iterable<T>): Sequence<T> {\n    return object: Sequence<T> {\n        override fun
iterator(): Iterator<T> {\n            val other = elements.convertToSetForSetOperation()\n            if (other.isEmpty())\n                return this@minus.iterator()\n            else\n                return this@minus.filterNot { it in other }.iterator()\n        }\n    }\n}\n}\n\n/**\n * Returns a sequence containing all elements of original sequence except the elements
contained in the given [elements] sequence.\n * \n * Note that the source sequence and the sequence being
subtracted are iterated only when an `iterator` is requested from\n * the resulting sequence. Changing any of them
between successive calls to `iterator` may affect the result.\n * \n * The operation is _intermediate_ for this sequence
and _terminal_ and _stateful_ for the [elements] sequence.\n * \n * Before Kotlin 1.6, the [elements] sequence may
have been converted to a [HashSet] to speed up the operation, thus the elements were required to have\n * a correct
and stable implementation of `hashCode()` that didn't change between successive invocations.\n * On JVM, you can
enable this behavior back with the system property `kotlin.collections.convert_arg_to_set_in_removeAll` set to
`true`.\n */\n\npublic operator fun <T> Sequence<T>.minus(elements: Sequence<T>): Sequence<T> {\n    return
object: Sequence<T> {\n        override fun iterator(): Iterator<T> {\n            val other =
elements.convertToSetForSetOperation()\n            if (other.isEmpty())\n                return this@minus.iterator()\n            else\n                return this@minus.filterNot { it in other }.iterator()\n        }\n    }\n}\n}\n\n/**\n * Returns a sequence
containing all elements of the original sequence without the first occurrence of the given [element].\n * \n * The
operation is _intermediate_ and _stateless_.\n */\n\n@kotlin.internal.InlineOnly\npublic inline fun <T>
Sequence<T>.minusElement(element: T): Sequence<T> {\n    return minus(element)\n}\n}\n\n/**\n * Splits the
original sequence into pair of lists,\n * where *first* list contains elements for which [predicate] yielded `true`,\n *
while *second* list contains elements for which [predicate] yielded `false`.\n * \n * The operation is _terminal_.\n *
\n * @sample samples.collections.Sequences.Transformations.partition\n */\n\npublic inline fun <T>
Sequence<T>.partition(predicate: (T) -> Boolean): Pair<List<T>, List<T>> {\n    val first = ArrayList<T>()\n    val
second = ArrayList<T>()\n    for (element in this) {\n        if (predicate(element)) {\n            first.add(element)\n        } else {\n            second.add(element)\n        }\n    }\n    return Pair(first, second)\n}\n}\n\n/**\n * Returns a sequence
containing all elements of the original sequence and then the given [element].\n * \n * The operation is
_intermediate_ and _stateless_.\n */\n\npublic operator fun <T> Sequence<T>.plus(element: T): Sequence<T> {\n
return sequenceOf(this, sequenceOf(element)).flatten()\n}\n}\n\n/**\n * Returns a sequence containing all elements of
original sequence and then all elements of the given [elements] array.\n * \n * Note that the source sequence and the
array being added are iterated only when an `iterator` is requested from\n * the resulting sequence. Changing any of
them between successive calls to `iterator` may affect the result.\n * \n * The operation is _intermediate_ and
_stateless_.\n */\n\npublic operator fun <T> Sequence<T>.plus(elements: Array<out T>): Sequence<T> {\n
return this.plus(elements.asList())\n}\n}\n\n/**\n * Returns a sequence containing all elements of original sequence and then
all elements of the given [elements] collection.\n * \n * Note that the source sequence and the collection being added
are iterated only when an `iterator` is requested from\n * the resulting sequence. Changing any of them between
successive calls to `iterator` may affect the result.\n * \n * The operation is _intermediate_ and _stateless_.\n
*/\n\npublic operator fun <T> Sequence<T>.plus(elements: Iterable<T>): Sequence<T> {\n    return sequenceOf(this,
elements.asSequence()).flatten()\n}\n}\n\n/**\n * Returns a sequence containing all elements of original sequence and
then all elements of the given [elements] sequence.\n * \n * Note that the source sequence and the sequence being
added are iterated only when an `iterator` is requested from\n * the resulting sequence. Changing any of them

```

between successive calls to `iterator` may affect the result.

`Sequence<T>.plus(elements: Sequence<T>): Sequence<T>` {
`return sequenceOf(this, elements).flatten()`}
Returns a sequence containing all elements of the original sequence and then the given [element].

`Sequence<T>.plusElement(element: T): Sequence<T>` {
`return plus(element)`}
Returns a sequence of snapshots of the window of the given [size] sliding along this sequence with the given [step], where each snapshot is a list. Several last lists may have fewer elements than the given [size]. Both [size] and [step] must be positive and can be greater than the number of elements in this sequence.

`Sequence<T>.takeWindows(size: Int, step: Int = 1, partialWindows: Boolean = false): Sequence<List<T>>` {
`return windowedSequence(size, step, partialWindows, reuseBuffer = false)`}
Returns a sequence of results of applying the given [transform] function to an each list representing a view over the window of the given [size] sliding along this sequence with the given [step]. Note that the list passed to the [transform] function is ephemeral and is valid only inside that function. You should not store it or allow it to escape in some way, unless you made a snapshot of it. Several last lists may have fewer elements than the given [size]. Both [size] and [step] must be positive and can be greater than the number of elements in this sequence.

`Sequence<T>.averageWindows(size: Int, step: Int = 1, partialWindows: Boolean = false, transform: (List<T>) -> R): Sequence<R>` {
`return windowedSequence(size, step, partialWindows, reuseBuffer = true).map(transform)`}
Returns a sequence of values built from the elements of `this` sequence and the [other] sequence with the same index. The resulting sequence ends as soon as the shortest input sequence ends.

`Sequence<T>.zip(other: Sequence<R>): Sequence<Pair<T, R>>` {
`return MergingSequence(this, other) { t1, t2 -> t1 to t2 }`}
Returns a sequence of values built from the elements of `this` sequence and the [other] sequence with the same index using the provided [transform] function applied to each pair of elements. The resulting sequence ends as soon as the shortest input sequence ends.

`Sequence<T>.zipWithTransform(other: Sequence<R>, transform: (a: T, b: R) -> V): Sequence<V>` {
`return MergingSequence(this, other, transform)`}
Returns a sequence of pairs of each two adjacent elements in this sequence. The returned sequence is empty if this sequence contains less than two elements.

`Sequence<T>.zipWithNext(): Sequence<Pair<T, T>>` {
`return zipWithNext { a, b -> a to b }`}
Returns a sequence containing the results of applying the given [transform] function to an each pair of two adjacent elements in this sequence. The returned sequence is empty if this sequence contains less than two elements.

`Sequence<T>.zipWithNextToFindDeltas(transform: (a: T, b: T) -> R): Sequence<R>` {
`return sequence result@ {
val iterator = iterator()
if (!iterator.hasNext()) return@result
var current = iterator.next()
while (iterator.hasNext()) {
val next = iterator.next()
yield(transform(current, next))
}`

```

current = next\n    }\n    }\n}\n\n/**\n * Appends the string from all the elements separated using [separator] and
using the given [prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-
negative value of [limit], in which case only the first [limit]\n * elements will be appended, followed by the
[truncated] string (which defaults to "...").\n * \n * The operation is _terminal_.\n * \n * @sample
samples.collections.Collections.Transformations.joinTo\n *\npublic fun <T, A : Appendable>
Sequence<T>.joinTo(buffer: A, separator: CharSequence = "\", ", prefix: CharSequence = "\"", postfix:
CharSequence = "\"", limit: Int = -1, truncated: CharSequence = "...", transform: ((T) -> CharSequence)? = null): A
{\n    buffer.append(prefix)\n    var count = 0\n    for (element in this) {\n        if (++count > 1)
buffer.append(separator)\n        if (limit < 0 || count <= limit) {\n            buffer.appendElement(element, transform)\n
        } else break\n    }\n    if (limit >= 0 && count > limit) buffer.append(truncated)\n    buffer.append(postfix)\n
return buffer\n}\n}\n\n/**\n * Creates a string from all the elements separated using [separator] and using the given
[prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value of
[limit], in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string (which
defaults to "...").\n * \n * The operation is _terminal_.\n * \n * @sample
samples.collections.Collections.Transformations.joinToString\n *\npublic fun <T>
Sequence<T>.joinToString(separator: CharSequence = "\", ", prefix: CharSequence = "\"", postfix: CharSequence =
"\", limit: Int = -1, truncated: CharSequence = "...", transform: ((T) -> CharSequence)? = null): String {\n    return
joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()\n}\n}\n\n/**\n * Creates an
[Iterable] instance that wraps the original sequence returning its elements when being iterated.\n *\npublic fun <T>
Sequence<T>.asIterable(): Iterable<T> {\n    return Iterable { this.iterator() }\n}\n}\n\n/**\n * Returns this sequence as
a [Sequence].\n *\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.asSequence(): Sequence<T>
{\n    return this\n}\n}\n\n/**\n * Returns an average value of elements in the sequence.\n * \n * The operation is
_terminal_.\n *\n@kotlin.jvm.JvmName("averageOfByte")\npublic fun Sequence<Byte>.average(): Double {\n
var sum: Double = 0.0\n    var count: Int = 0\n    for (element in this) {\n        sum += element\n
checkCountOverflow(++count)\n    }\n    return if (count == 0) Double.NaN else sum / count\n}\n}\n\n/**\n * Returns
an average value of elements in the sequence.\n * \n * The operation is _terminal_.\n
*\n@kotlin.jvm.JvmName("averageOfShort")\npublic fun Sequence<Short>.average(): Double {\n    var sum:
Double = 0.0\n    var count: Int = 0\n    for (element in this) {\n        sum += element\n
checkCountOverflow(++count)\n    }\n    return if (count == 0) Double.NaN else sum / count\n}\n}\n\n/**\n * Returns
an average value of elements in the sequence.\n * \n * The operation is _terminal_.\n
*\n@kotlin.jvm.JvmName("averageOfInt")\npublic fun Sequence<Int>.average(): Double {\n    var sum: Double
= 0.0\n    var count: Int = 0\n    for (element in this) {\n        sum += element\n        checkCountOverflow(++count)\n
    }\n    return if (count == 0) Double.NaN else sum / count\n}\n}\n\n/**\n * Returns an average value of elements in the
sequence.\n * \n * The operation is _terminal_.\n *\n@kotlin.jvm.JvmName("averageOfLong")\npublic fun
Sequence<Long>.average(): Double {\n    var sum: Double = 0.0\n    var count: Int = 0\n    for (element in this) {\n
sum += element\n        checkCountOverflow(++count)\n    }\n    return if (count == 0) Double.NaN else sum /
count\n}\n}\n\n/**\n * Returns an average value of elements in the sequence.\n * \n * The operation is _terminal_.\n
*\n@kotlin.jvm.JvmName("averageOfFloat")\npublic fun Sequence<Float>.average(): Double {\n    var sum:
Double = 0.0\n    var count: Int = 0\n    for (element in this) {\n        sum += element\n
checkCountOverflow(++count)\n    }\n    return if (count == 0) Double.NaN else sum / count\n}\n}\n\n/**\n * Returns
an average value of elements in the sequence.\n * \n * The operation is _terminal_.\n
*\n@kotlin.jvm.JvmName("averageOfDouble")\npublic fun Sequence<Double>.average(): Double {\n    var sum:
Double = 0.0\n    var count: Int = 0\n    for (element in this) {\n        sum += element\n
checkCountOverflow(++count)\n    }\n    return if (count == 0) Double.NaN else sum / count\n}\n}\n\n/**\n * Returns
the sum of all elements in the sequence.\n * \n * The operation is _terminal_.\n
*\n@kotlin.jvm.JvmName("sumOfByte")\npublic fun Sequence<Byte>.sum(): Int {\n    var sum: Int = 0\n    for
(element in this) {\n        sum += element\n    }\n    return sum\n}\n}\n\n/**\n * Returns the sum of all elements in the
sequence.\n * \n * The operation is _terminal_.\n *\n@kotlin.jvm.JvmName("sumOfShort")\npublic fun

```

```

Sequence<Short>.sum(): Int {\n  var sum: Int = 0\n  for (element in this) {\n    sum += element\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all elements in the sequence.\n *\n * The operation is _terminal_.\n *\n */\n@kotlin.jvm.JvmName("sumOfInt")\npublic fun Sequence<Int>.sum(): Int {\n  var sum: Int = 0\n  for (element in this) {\n    sum += element\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all elements in the sequence.\n *\n * The operation is _terminal_.\n *\n */\n@kotlin.jvm.JvmName("sumOfLong")\npublic fun Sequence<Long>.sum(): Long {\n  var sum: Long = 0L\n  for (element in this) {\n    sum += element\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all elements in the sequence.\n *\n * The operation is _terminal_.\n *\n */\n@kotlin.jvm.JvmName("sumOfFloat")\npublic fun Sequence<Float>.sum(): Float {\n  var sum: Float = 0.0f\n  for (element in this) {\n    sum += element\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all elements in the sequence.\n *\n * The operation is _terminal_.\n *\n */\n@kotlin.jvm.JvmName("sumOfDouble")\npublic fun Sequence<Double>.sum(): Double {\n  var sum: Double = 0.0\n  for (element in this) {\n    sum += element\n  }\n  return sum\n}\n\n", "*/\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n\n\n*\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("SetsKt")\n\npackage kotlin.collections\n\n\n/\nNOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\nSee: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\nimport kotlin.random.*\nimport kotlin.ranges.contains\nimport kotlin.ranges.reversed\n\n/**\n * Returns a set containing all elements of the original set except the given [element].\n *\n * The returned set preserves the element iteration order of the original set.\n *\n */\npublic operator fun <T> Set<T>.minus(element: T): Set<T> {\n  val result = LinkedHashSet<T>(mapCapacity(size))\n  var removed = false\n  return this.filterTo(result) { if (!removed && it == element) { removed = true; false } else true }\n}\n\n/**\n * Returns a set containing all elements of the original set except the elements contained in the given [elements] array.\n *\n * The returned set preserves the element iteration order of the original set.\n *\n * Before Kotlin 1.6, the [elements] array may have been converted to a [HashSet] to speed up the operation, thus the elements were required to have\n * a correct and stable implementation of `hashCode()` that didn't change between successive invocations.\n * On JVM, you can enable this behavior back with the system property `kotlin.collections.convert_arg_to_set_in_removeAll` set to `true`.\n *\n */\npublic operator fun <T> Set<T>.minus(elements: Array<out T>): Set<T> {\n  val result = LinkedHashSet<T>(this)\n  result.removeAll(elements)\n  return result\n}\n\n/**\n * Returns a set containing all elements of the original set except the elements contained in the given [elements] collection.\n *\n * The returned set preserves the element iteration order of the original set.\n *\n * Before Kotlin 1.6, the [elements] collection may have been converted to a [HashSet] to speed up the operation, thus the elements were required to have\n * a correct and stable implementation of `hashCode()` that didn't change between successive invocations.\n * On JVM, you can enable this behavior back with the system property `kotlin.collections.convert_arg_to_set_in_removeAll` set to `true`.\n *\n */\npublic operator fun <T> Set<T>.minus(elements: Iterable<T>): Set<T> {\n  val other = elements.convertToSetForSetOperationWith(this)\n  if (other.isEmpty())\n    return this.toSet()\n  if (other is Set)\n    return this.filterNotTo(LinkedHashSet<T>()) { it in other }\n  val result = LinkedHashSet<T>(this)\n  result.removeAll(other)\n  return result\n}\n\n/**\n * Returns a set containing all elements of the original set except the elements contained in the given [elements] sequence.\n *\n * The returned set preserves the element iteration order of the original set.\n *\n * Before Kotlin 1.6, the [elements] sequence may have been converted to a [HashSet] to speed up the operation, thus the elements were required to have\n * a correct and stable implementation of `hashCode()` that didn't change between successive invocations.\n * On JVM, you can enable this behavior back with the system property `kotlin.collections.convert_arg_to_set_in_removeAll` set to `true`.\n *\n */\npublic operator fun <T> Set<T>.minus(elements: Sequence<T>): Set<T> {\n  val result = LinkedHashSet<T>(this)\n  result.removeAll(elements)\n  return result\n}\n\n/**\n * Returns a set containing all elements of the original set except the given [element].\n *\n * The returned set preserves the element iteration order of the original set.\n *\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T> Set<T>.minusElement(element: T): Set<T> {\n  return minus(element)\n}\n\n/**\n * Returns a set containing all elements of the original set and then the given [element] if

```

it isn't already in this set.\n * \n * The returned set preserves the element iteration order of the original set.\n

```

*\npublic operator fun <T> Set<T>.plus(element: T): Set<T> {\n    val result =
    LinkedHashSet<T>(mapCapacity(size + 1))\n    result.addAll(this)\n    result.add(element)\n    return
    result\n}\n\n/**\n * Returns a set containing all elements of the original set and the given [elements] array,\n * which aren't already in this set.\n * \n * The returned set preserves the element iteration order of the original set.\n
*\npublic operator fun <T> Set<T>.plus(elements: Array<out T>): Set<T> {\n    val result =
    LinkedHashSet<T>(mapCapacity(this.size + elements.size))\n    result.addAll(this)\n    result.addAll(elements)\n
    return result\n}\n\n/**\n * Returns a set containing all elements of the original set and the given [elements]
    collection,\n * which aren't already in this set.\n * \n * The returned set preserves the element iteration order of the
    original set.\n
*\npublic operator fun <T> Set<T>.plus(elements: Iterable<T>): Set<T> {\n    val result =
    LinkedHashSet<T>(mapCapacity(elements.collectionSizeOrNull()?.let { this.size + it } ?: this.size * 2))\n
    result.addAll(this)\n    result.addAll(elements)\n    return result\n}\n\n/**\n * Returns a set containing all elements of
    the original set and the given [elements] sequence,\n * which aren't already in this set.\n * \n * The returned set
    preserves the element iteration order of the original set.\n
*\npublic operator fun <T> Set<T>.plus(elements:
    Sequence<T>): Set<T> {\n    val result = LinkedHashSet<T>(mapCapacity(this.size * 2))\n    result.addAll(this)\n
    result.addAll(elements)\n    return result\n}\n\n/**\n * Returns a set containing all elements of the original set and
    then the given [element] if it isn't already in this set.\n * \n * The returned set preserves the element iteration
    order of the original set.\n
*\n@kotlin.internal.InlineOnly\npublic inline fun <T> Set<T>.plusElement(element: T):
    Set<T> {\n    return plus(element)\n}\n\n"/**\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming
    Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
    license/LICENSE.txt file.\n
*/\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("StringsKt")\n\npackage
    kotlin.text\n\n/\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See:
    https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\n/\n\nimport kotlin.random.*\n\n/**\n * Returns a
    character at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this char
    sequence.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n
*\npublic expect fun
    CharSequence.elementAt(index: Int): Char\n\n/**\n * Returns a character at the given [index] or the result of calling
    the [defaultValue] function if the [index] is out of bounds of this char sequence.\n * \n * @sample
    samples.collections.Collections.Elements.elementAtOrElse\n
*\n@kotlin.internal.InlineOnly\npublic inline fun
    CharSequence.elementAtOrElse(index: Int, defaultValue: (Int) -> Char): Char {\n    return if (index >= 0 && index
    <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns a character at the given [index] or `null` if
    the [index] is out of bounds of this char sequence.\n * \n * @sample
    samples.collections.Collections.Elements.elementAtOrNull\n
*\n@kotlin.internal.InlineOnly\npublic inline fun
    CharSequence.elementAtOrNull(index: Int): Char? {\n    return this.getOrNull(index)\n}\n\n/**\n * Returns the first
    character matching the given [predicate], or `null` if no such character was found.\n * \n * @sample
    samples.collections.Collections.Elements.find\n
*\n@kotlin.internal.InlineOnly\npublic inline fun
    CharSequence.find(predicate: (Char) -> Boolean): Char? {\n    return firstOrNull(predicate)\n}\n\n/**\n * Returns
    the last character matching the given [predicate], or `null` if no such character was found.\n * \n * @sample
    samples.collections.Collections.Elements.find\n
*\n@kotlin.internal.InlineOnly\npublic inline fun
    CharSequence.find(predicate: (Char) -> Boolean): Char? {\n    return lastOrNull(predicate)\n}\n\n/**\n *
    Returns first character.\n * @throws [NoSuchElementException] if the char sequence is empty.\n
*\npublic fun
    CharSequence.first(): Char {\n    if (isEmpty())\n        throw NoSuchElementException("Char sequence is
    empty.")\n    return this[0]\n}\n\n/**\n * Returns the first character matching the given [predicate].\n * @throws
    [NoSuchElementException] if no such character is found.\n
*\npublic inline fun CharSequence.first(predicate:
    (Char) -> Boolean): Char {\n    for (element in this) if (predicate(element)) return element\n    throw
    NoSuchElementException("Char sequence contains no character matching the predicate.")\n}\n\n/**\n * Returns
    the first non-null value produced by [transform] function being applied to characters of this char sequence in
    iteration order,\n * or throws [NoSuchElementException] if no non-null value was produced.\n * \n * @sample

```

```

samples.collections.Collections.Transformations.firstNotNullOf
*^@SinceKotlin("1.5")@kotlin.internal.InlineOnly\npublic inline fun <R : Any>
CharSequence.firstNotNullOf(transform: (Char) -> R?): R {\n    return firstNotNullOfOrNull(transform) ?: throw
NoSuchElementException("No element of the char sequence was transformed to a non-null value.")\n}\n\n/**\n *
Returns the first non-null value produced by [transform] function being applied to characters of this char sequence in
iteration order,\n * or `null` if no non-null value was produced.\n * \n * @sample
samples.collections.Collections.Transformations.firstNotNullOf
*^@SinceKotlin("1.5")@kotlin.internal.InlineOnly\npublic inline fun <R : Any>
CharSequence.firstNotNullOfOrNull(transform: (Char) -> R?): R? {\n    for (element in this) {\n        val result =
transform(element)\n        if (result != null) {\n            return result\n        }\n    }\n    return null\n}\n\n/**\n *
Returns the first character, or `null` if the char sequence is empty.\n *^@public fun CharSequence.firstOrNull():
Char? {\n    return if (isEmpty()) null else this[0]\n}\n\n/**\n * Returns the first character matching the given
[predicate], or `null` if character was not found.\n *^@public inline fun CharSequence.firstOrNull(predicate: (Char) -
> Boolean): Char? {\n    for (element in this) if (predicate(element)) return element\n    return null\n}\n\n/**\n *
Returns a character at the given [index] or the result of calling the [defaultValue] function if the [index] is out of
bounds of this char sequence.\n *^@kotlin.internal.InlineOnly\npublic inline fun CharSequence.getOrElse(index:
Int, defaultValue: (Int) -> Char): Char {\n    return if (index >= 0 && index <= lastIndex) get(index) else
defaultValue(index)\n}\n\n/**\n * Returns a character at the given [index] or `null` if the [index] is out of bounds of
this char sequence.\n * \n * @sample samples.collections.Collections.Elements.getOrElse\n *^@public fun
CharSequence.getOrElse(index: Int): Char? {\n    return if (index >= 0 && index <= lastIndex) get(index) else
null\n}\n\n/**\n * Returns index of the first character matching the given [predicate], or -1 if the char sequence does
not contain such character.\n *^@public inline fun CharSequence.indexOfFirst(predicate: (Char) -> Boolean): Int {\n
for (index in indices) {\n    if (predicate(this[index])) {\n        return index\n    }\n}\n    return -
1\n}\n\n/**\n * Returns index of the last character matching the given [predicate], or -1 if the char sequence does
not contain such character.\n *^@public inline fun CharSequence.indexOfLast(predicate: (Char) -> Boolean): Int {\n
for (index in indices.reversed()) {\n    if (predicate(this[index])) {\n        return index\n    }\n}\n    return -
1\n}\n\n/**\n * Returns the last character.\n * \n * @throws NoSuchElementException if the char sequence is
empty.\n * \n * @sample samples.text.Strings.last\n *^@public fun CharSequence.last(): Char {\n    if (isEmpty())\n        throw NoSuchElementException("Char sequence is empty.")\n    return this[lastIndex]\n}\n\n/**\n * Returns the
last character matching the given [predicate].\n * \n * @throws NoSuchElementException if no such character is
found.\n * \n * @sample samples.text.Strings.last\n *^@public inline fun CharSequence.last(predicate: (Char) ->
Boolean): Char {\n    for (index in this.indices.reversed()) {\n        val element = this[index]\n        if
(predicate(element)) return element\n    }\n    throw NoSuchElementException("Char sequence contains no
character matching the predicate.")\n}\n\n/**\n * Returns the last character, or `null` if the char sequence is
empty.\n * \n * @sample samples.text.Strings.last\n *^@public fun CharSequence.lastOrNull(): Char? {\n    return if
(isEmpty()) null else this[length - 1]\n}\n\n/**\n * Returns the last character matching the given [predicate], or `null`
if no such character was found.\n * \n * @sample samples.text.Strings.last\n *^@public inline fun
CharSequence.lastOrNull(predicate: (Char) -> Boolean): Char? {\n    for (index in this.indices.reversed()) {\n
val element = this[index]\n        if (predicate(element)) return element\n    }\n    return null\n}\n\n/**\n * Returns a
random character from this char sequence.\n * \n * @throws NoSuchElementException if this char sequence is
empty.\n *^@SinceKotlin("1.3")@kotlin.internal.InlineOnly\npublic inline fun CharSequence.random(): Char
{\n    return random(Random)\n}\n\n/**\n * Returns a random character from this char sequence using the specified
source of randomness.\n * \n * @throws NoSuchElementException if this char sequence is empty.\n *
*^@SinceKotlin("1.3")\npublic fun CharSequence.random(random: Random): Char {\n    if (isEmpty())\n        throw
NoSuchElementException("Char sequence is empty.")\n    return get(random.nextInt(length))\n}\n\n/**\n *
Returns a random character from this char sequence, or `null` if this char sequence is empty.\n *
*^@SinceKotlin("1.4")@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
inline fun CharSequence.randomOrNull(): Char? {\n    return randomOrNull(Random)\n}\n\n/**\n * Returns a

```

random character from this char sequence using the specified source of randomness, or `null` if this char sequence is empty.

```

@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public fun CharSequence.randomOrNull(random: Random): Char? {
    if (isEmpty()) return null
    return get(random.nextInt(length))
}

```

Returns the single character, or throws an exception if the char sequence is empty or has more than one character.

```

public fun CharSequence.single(): Char {
    return when (length) {
        0 -> throw NoSuchElementException("Char sequence is empty.")
        1 -> this[0]
        else -> throw IllegalArgumentException("Char sequence has more than one element.")
    }
}

```

Returns the single character matching the given [predicate], or throws exception if there is no or more than one matching character.

```

public inline fun CharSequence.single(predicate: (Char) -> Boolean): Char {
    var single: Char? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw IllegalArgumentException("Char sequence contains more than one matching element.")
            single = element
            found = true
        }
    }
    if (!found) throw NoSuchElementException("Char sequence contains no character matching the predicate.")
    @SuppressWarnings("UNCHECKED_CAST") return single as Char
}

```

Returns single character, or `null` if the char sequence is empty or has more than one character.

```

public fun CharSequence.singleOrNull(): Char? {
    return if (length == 1) this[0] else null
}

```

Returns the single character matching the given [predicate], or `null` if character was not found or more than one character was found.

```

public inline fun CharSequence.singleOrNull(predicate: (Char) -> Boolean): Char? {
    var single: Char? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) return null
            single = element
            found = true
        }
    }
    if (!found) return null
    return single
}

```

Returns a subsequence of this char sequence with the first [n] characters removed.

```

@throws IllegalArgumentException if [n] is negative.
@sample samples.text.Strings.drop
public fun CharSequence.drop(n: Int): CharSequence {
    require(n >= 0) { "Requested character count $n is less than zero." }
    return subSequence(n.coerceAtMost(length), length)
}

```

Returns a string with the first [n] characters removed.

```

@throws IllegalArgumentException if [n] is negative.
@sample samples.text.Strings.drop
public fun String.drop(n: Int): String {
    require(n >= 0) { "Requested character count $n is less than zero." }
    return substring(n.coerceAtMost(length))
}

```

Returns a subsequence of this char sequence with the last [n] characters removed.

```

@throws IllegalArgumentException if [n] is negative.
@sample samples.text.Strings.drop
public fun CharSequence.dropLast(n: Int): CharSequence {
    require(n >= 0) { "Requested character count $n is less than zero." }
    return take((length - n).coerceAtLeast(0))
}

```

Returns a string with the last [n] characters removed.

```

@throws IllegalArgumentException if [n] is negative.
@sample samples.text.Strings.drop
public fun String.dropLast(n: Int): String {
    require(n >= 0) { "Requested character count $n is less than zero." }
    return take((length - n).coerceAtLeast(0))
}

```

Returns a subsequence of this char sequence containing all characters except last characters that satisfy the given [predicate].

```

@sample samples.text.Strings.drop
public inline fun CharSequence.dropLastWhile(predicate: (Char) -> Boolean): CharSequence {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) return subSequence(0, index + 1)
    }
    return ""
}

```

Returns a string containing all characters except last characters that satisfy the given [predicate].

```

@sample samples.text.Strings.drop
public inline fun String.dropLastWhile(predicate: (Char) -> Boolean): String {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) return substring(0, index + 1)
    }
    return ""
}

```

Returns a subsequence of this char sequence containing all characters except first characters that satisfy the given [predicate].

```

@sample samples.text.Strings.drop
public inline fun CharSequence.dropWhile(predicate: (Char) -> Boolean): CharSequence {
    for (index in this.indices) {
        if (!predicate(this[index])) return subSequence(index, length)
    }
    return ""
}

```

Returns a string containing all characters except first characters that satisfy the given [predicate].

```

@sample samples.text.Strings.drop
public inline fun String.dropWhile(predicate: (Char) -> Boolean): String {
    for (index in this.indices) {
        if (!predicate(this[index])) return substring(index)
    }
    return ""
}

```

Returns a char sequence containing only those characters from the original char sequence that match the given [predicate].

```

@sample samples.text.Strings.filter
public inline fun

```

```

CharSequence.filter(predicate: (Char) -> Boolean): CharSequence {
    return filterTo(StringBuilder(),
predicate)\n}\n\n/**\n * Returns a string containing only those characters from the original string that match the
given [predicate].\n * \n * @sample samples.text.Strings.filter\n */\npublic inline fun String.filter(predicate: (Char) -
> Boolean): String {
    return filterTo(StringBuilder(), predicate).toString()\n}\n\n/**\n * Returns a char sequence
containing only those characters from the original char sequence that match the given [predicate].\n * @param
[predicate] function that takes the index of a character and the character itself\n * and returns the result of predicate
evaluation on the character.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexed\n */\npublic
inline fun CharSequence.filterIndexed(predicate: (index: Int, Char) -> Boolean): CharSequence {
    return
filterIndexedTo(StringBuilder(), predicate)\n}\n\n/**\n * Returns a string containing only those characters from the
original string that match the given [predicate].\n * @param [predicate] function that takes the index of a character
and the character itself\n * and returns the result of predicate evaluation on the character.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexed\n */\npublic inline fun String.filterIndexed(predicate: (index:
Int, Char) -> Boolean): String {
    return filterIndexedTo(StringBuilder(), predicate).toString()\n}\n\n/**\n *
Appends all characters matching the given [predicate] to the given [destination].\n * @param [predicate] function
that takes the index of a character and the character itself\n * and returns the result of predicate evaluation on the
character.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexedTo\n */\npublic inline fun <C :
Appendable> CharSequence.filterIndexedTo(destination: C, predicate: (index: Int, Char) -> Boolean): C {
    for (index, element) in {
        if (predicate(index, element)) destination.append(element)\n    }\n    return destination\n}\n\n/**\n * Returns a char sequence containing only those characters from the original char
sequence that do not match the given [predicate].\n * \n * @sample samples.text.Strings.filterNot\n */\npublic inline
fun CharSequence.filterNot(predicate: (Char) -> Boolean): CharSequence {
    return filterNotTo(StringBuilder(),
predicate)\n}\n\n/**\n * Returns a string containing only those characters from the original string that do not match
the given [predicate].\n * \n * @sample samples.text.Strings.filterNot\n */\npublic inline fun
String.filterNot(predicate: (Char) -> Boolean): String {
    return filterNotTo(StringBuilder(),
predicate).toString()\n}\n\n/**\n * Appends all characters not matching the given [predicate] to the given
[destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n */\npublic inline fun <C :
Appendable> CharSequence.filterNotTo(destination: C, predicate: (Char) -> Boolean): C {
    for (element in this)
if (!predicate(element)) destination.append(element)\n    return destination\n}\n\n/**\n * Appends all characters
matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n */\npublic inline fun <C : Appendable>
CharSequence.filterTo(destination: C, predicate: (Char) -> Boolean): C {
    for (index in 0 until length) {
        val
element = get(index)\n        if (predicate(element)) destination.append(element)\n    }\n    return
destination\n}\n\n/**\n * Returns a char sequence containing characters of the original char sequence at the
specified range of [indices].\n */\npublic fun CharSequence.slice(indices: IntRange): CharSequence {
    if
(indices.isEmpty()) return ""\n    return subSequence(indices)\n}\n\n/**\n * Returns a string containing characters
of the original string at the specified range of [indices].\n */\npublic fun String.slice(indices: IntRange): String {
    if
(indices.isEmpty()) return ""\n    return substring(indices)\n}\n\n/**\n * Returns a char sequence containing
characters of the original char sequence at specified [indices].\n */\npublic fun CharSequence.slice(indices:
Iterable<Int>): CharSequence {
    val size = indices.collectionSizeOrDefault(10)\n    if (size == 0) return ""\n    val
result = StringBuilder(size)\n    for (i in indices) {
        result.append(get(i))\n    }\n    return result\n}\n\n/**\n * Returns a string containing characters of the original string at specified [indices].\n
*/\n\n@kotlin.internal.InlineOnly\npublic inline fun String.slice(indices: Iterable<Int>): String {
    return (this as
CharSequence).slice(indices).toString()\n}\n\n/**\n * Returns a subsequence of this char sequence containing the
first [n] characters from this char sequence, or the entire char sequence if this char sequence is shorter.\n * \n *
@throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.text.Strings.take\n */\npublic fun
CharSequence.take(n: Int): CharSequence {
    require(n >= 0) { "Requested character count $n is less than zero."
}\n    return subSequence(0, n.coerceAtMost(length))\n}\n\n/**\n * Returns a string containing the first [n]
characters from this string, or the entire string if this string is shorter.\n * \n * @throws IllegalArgumentException if

```

```

[n] is negative.\n * \n * @sample samples.text.Strings.take\n */\npublic fun String.take(n: Int): String {\n    require(n
>= 0) { \"Requested character count $n is less than zero.\" }\n    return substring(0,
n.coerceAtMost(length))\n}\n\n/**\n * Returns a subsequence of this char sequence containing the last [n]
characters from this char sequence, or the entire char sequence if this char sequence is shorter.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample samples.text.Strings.take\n */\npublic fun
CharSequence.takeLast(n: Int): CharSequence {\n    require(n >= 0) { \"Requested character count $n is less than
zero.\" }\n    val length = length\n    return subSequence(length - n.coerceAtMost(length), length)\n}\n\n/**\n *
Returns a string containing the last [n] characters from this string, or the entire string if this string is shorter.\n * \n *
@throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.text.Strings.take\n */\npublic fun
String.takeLast(n: Int): String {\n    require(n >= 0) { \"Requested character count $n is less than zero.\" }\n    val
length = length\n    return substring(length - n.coerceAtMost(length))\n}\n\n/**\n * Returns a subsequence of this
char sequence containing last characters that satisfy the given [predicate].\n * \n * @sample
samples.text.Strings.take\n */\npublic inline fun CharSequence.takeLastWhile(predicate: (Char) -> Boolean):
CharSequence {\n    for (index in lastIndex downTo 0) {\n        if (!predicate(this[index])) {\n            return
subSequence(index + 1, length)\n        }\n    }\n    return subSequence(0, length)\n}\n\n/**\n * Returns a string
containing last characters that satisfy the given [predicate].\n * \n * @sample samples.text.Strings.take\n */\npublic
inline fun String.takeLastWhile(predicate: (Char) -> Boolean): String {\n    for (index in lastIndex downTo 0) {\n
        if (!predicate(this[index])) {\n            return substring(index + 1)\n        }\n    }\n    return this\n}\n\n/**\n * Returns
a subsequence of this char sequence containing the first characters that satisfy the given [predicate].\n * \n *
@sample samples.text.Strings.take\n */\npublic inline fun CharSequence.takeWhile(predicate: (Char) -> Boolean):
CharSequence {\n    for (index in 0 until length)\n        if (!predicate(get(index))) {\n            return subSequence(0,
index)\n        }\n    return subSequence(0, length)\n}\n\n/**\n * Returns a string containing the first characters that
satisfy the given [predicate].\n * \n * @sample samples.text.Strings.take\n */\npublic inline fun
String.takeWhile(predicate: (Char) -> Boolean): String {\n    for (index in 0 until length)\n        if
(!predicate(get(index))) {\n            return substring(0, index)\n        }\n    return this\n}\n\n/**\n * Returns a char
sequence with characters in reversed order.\n */\npublic fun CharSequence.reversed(): CharSequence {\n    return
StringBuilder(this).reverse()\n}\n\n/**\n * Returns a string with characters in reversed order.\n */\n@kotlin.internal.InlineOnly\npublic inline fun String.reversed(): String {\n    return (this as
CharSequence).reversed().toString()\n}\n\n/**\n * Returns a [Map] containing key-value pairs provided by
[transform] function\n * applied to characters of the given char sequence.\n * \n * If any of two pairs would have the
same key the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the
original char sequence.\n * \n * @sample samples.text.Strings.associate\n */\npublic inline fun <K, V>
CharSequence.associate(transform: (Char) -> Pair<K, V>): Map<K, V> {\n    val capacity =
mapCapacity(length).coerceAtLeast(16)\n    return associateTo(LinkedHashMap<K, V>(capacity),
transform)\n}\n\n/**\n * Returns a [Map] containing the characters from the given char sequence indexed by the
key\n * returned from [keySelector] function applied to each character.\n * \n * If any two characters would have the
same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry
iteration order of the original char sequence.\n * \n * @sample samples.text.Strings.associateBy\n */\npublic inline
fun <K> CharSequence.associateBy(keySelector: (Char) -> K): Map<K, Char> {\n    val capacity =
mapCapacity(length).coerceAtLeast(16)\n    return associateByTo(LinkedHashMap<K, Char>(capacity),
keySelector)\n}\n\n/**\n * Returns a [Map] containing the values provided by [valueTransform] and indexed by
[keySelector] functions applied to characters of the given char sequence.\n * \n * If any two characters would have
the same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the
entry iteration order of the original char sequence.\n * \n * @sample
samples.text.Strings.associateByWithValueTransform\n */\npublic inline fun <K, V>
CharSequence.associateBy(keySelector: (Char) -> K, valueTransform: (Char) -> V): Map<K, V> {\n    val capacity
= mapCapacity(length).coerceAtLeast(16)\n    return associateByTo(LinkedHashMap<K, V>(capacity),
keySelector, valueTransform)\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value

```

pairs, where key is provided by the [keySelector] function applied to each character of the given char sequence and value is the character itself. If any two characters would have the same key returned by [keySelector] the last one gets added to the map.

```

@sample samples.text.Strings.associateByTo
public inline fun <K, M : MutableMap<in K, in Char>> CharSequence.associateByTo(destination: M, keySelector: (Char) -> K): M {
    for (element in this) {
        destination.put(keySelector(element), element)
    }
    return destination
}

```

Populates and returns the [destination] mutable map with key-value pairs, where key is provided by the [keySelector] function and value is provided by the [valueTransform] function applied to characters of the given char sequence. If any two characters would have the same key returned by [keySelector] the last one gets added to the map.

```

@sample samples.text.Strings.associateByToWithValueTransform
public inline fun <K, V, M : MutableMap<in K, in V>> CharSequence.associateByTo(destination: M, keySelector: (Char) -> K, valueTransform: (Char) -> V): M {
    for (element in this) {
        destination.put(keySelector(element), valueTransform(element))
    }
    return destination
}

```

Populates and returns the [destination] mutable map with key-value pairs provided by [transform] function applied to each character of the given char sequence. If any of two pairs would have the same key the last one gets added to the map.

```

@sample samples.text.Strings.associateTo
public inline fun <K, V, M : MutableMap<in K, in V>> CharSequence.associateTo(destination: M, transform: (Char) -> Pair<K, V>): M {
    for (element in this) {
        destination += transform(element)
    }
    return destination
}

```

Returns a [Map] where keys are characters from the given char sequence and values are produced by the [valueSelector] function applied to each character. If any two characters are equal, the last one gets added to the map. The returned map preserves the entry iteration order of the original char sequence.

```

@sample samples.text.Strings.associateWith
public inline fun <V> CharSequence.associateWith(valueSelector: (Char) -> V): Map<Char, V> {
    val result = LinkedHashMap<Char, V>(mapCapacity(length.coerceAtMost(128)).coerceAtLeast(16))
    return associateWithTo(result, valueSelector)
}

```

Populates and returns the [destination] mutable map with key-value pairs for each character of the given char sequence, where key is the character itself and value is provided by the [valueSelector] function applied to that key. If any two characters are equal, the last one overwrites the former value in the map.

```

@sample samples.text.Strings.associateWithTo
public inline fun <V, M : MutableMap<in Char, in V>> CharSequence.associateWithTo(destination: M, valueSelector: (Char) -> V): M {
    for (element in this) {
        destination.put(element, valueSelector(element))
    }
    return destination
}

```

Appends all characters to the given [destination] collection.

```

public fun <C : MutableCollection<in Char>> CharSequence.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

```

Returns a new [HashSet] of all characters.

```

public fun CharSequence.toHashSet(): HashSet<Char> {
    return toCollection(HashSet<Char>(mapCapacity(length.coerceAtMost(128))))
}

```

Returns a [List] containing all characters.

```

public fun CharSequence.toList(): List<Char> {
    return when (length) {
        0 -> emptyList()
        1 -> listOf(this[0])
        else -> this.toMutableList()
    }
}

```

Returns a new [MutableList] filled with all characters of this char sequence.

```

public fun CharSequence.toMutableList(): MutableList<Char> {
    return toCollection(ArrayList<Char>(length))
}

```

Returns a [Set] of all characters. The returned set preserves the element iteration order of the original char sequence.

```

public fun CharSequence.toSet(): Set<Char> {
    return when (length) {
        0 -> emptySet()
        1 -> setOf(this[0])
        else -> toCollection(LinkedHashSet<Char>(mapCapacity(length.coerceAtMost(128))))
    }
}

```

Returns a single list of all elements yielded from results of [transform] function being invoked on each character of original char sequence.

```

@sample samples.collections.Collections.Transformations.flatMap
public inline fun <R> CharSequence.flatMap(transform: (Char) -> Iterable<R>): List<R> {
    return flatMapTo(ArrayList<R>(), transform)
}

```

Returns a single list of all elements yielded from results of [transform] function being invoked on each character and its index in the original char sequence.

```

@sample samples.collections.Collections.Transformations.flatMapIndexed

```

```

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterable")\n@kotlin.internal.InlineOnly\npublic
inline fun <R> CharSequence.flatMapIndexed(transform: (index: Int, Char) -> Iterable<R>): List<R> {\n  return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Appends all elements yielded from results of
[transform] function being invoked on each character\n * and its index in the original char sequence, to the given
[destination].\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedTo")\n@kotlin.internal.InlineOnly\npublic
inline fun <R, C : MutableCollection<in R>> CharSequence.flatMapIndexedTo(destination: C, transform: (index:
Int, Char) -> Iterable<R>): C {\n  var index = 0\n  for (element in this) {\n    val list = transform(index++,
element)\n    destination.addAll(list)\n  }\n  return destination\n}\n\n/**\n * Appends all elements yielded from
results of [transform] function being invoked on each character of original char sequence, to the given
[destination].\n *\npublic inline fun <R, C : MutableCollection<in R>> CharSequence.flatMapTo(destination: C,
transform: (Char) -> Iterable<R>): C {\n  for (element in this) {\n    val list = transform(element)\n
destination.addAll(list)\n  }\n  return destination\n}\n\n/**\n * Groups characters of the original char sequence by
the key returned by the given [keySelector] function\n * applied to each character and returns a map where each
group key is associated with a list of corresponding characters.\n * \n * The returned map preserves the entry
iteration order of the keys produced from the original char sequence.\n * \n * @sample
samples.collections.Collections.Transformations.groupBy\n *\npublic inline fun <K>
CharSequence.groupBy(keySelector: (Char) -> K): Map<K, List<Char>> {\n  return
groupByTo(LinkedHashMap<K, MutableList<Char>>(), keySelector)\n}\n\n/**\n * Groups values returned by the
[valueTransform] function applied to each character of the original char sequence\n * by the key returned by the
given [keySelector] function applied to the character\n * and returns a map where each group key is associated with
a list of corresponding values.\n * \n * The returned map preserves the entry iteration order of the keys produced
from the original char sequence.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeysAndValues\n *\npublic inline fun <K, V>
CharSequence.groupBy(keySelector: (Char) -> K, valueTransform: (Char) -> V): Map<K, List<V>> {\n  return
groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n * Groups
characters of the original char sequence by the key returned by the given [keySelector] function\n * applied to each
character and puts to the [destination] map each group key associated with a list of corresponding characters.\n * \n
* @return The [destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n
*\npublic inline fun <K, M : MutableMap<in K, MutableList<Char>>> CharSequence.groupByTo(destination: M,
keySelector: (Char) -> K): M {\n  for (element in this) {\n    val key = keySelector(element)\n    val list =
destination.getOrPut(key) { ArrayList<Char>() }\n    list.add(element)\n  }\n  return destination\n}\n\n/**\n *
Groups values returned by the [valueTransform] function applied to each character of the original char sequence\n *
by the key returned by the given [keySelector] function applied to the character\n * and puts to the [destination]
map each group key associated with a list of corresponding values.\n * \n * @return The [destination] map.\n * \n *
@sample samples.collections.Collections.Transformations.groupByKeysAndValues\n *\npublic inline fun <K, V,
M : MutableMap<in K, MutableList<V>>> CharSequence.groupByTo(destination: M, keySelector: (Char) -> K,
valueTransform: (Char) -> V): M {\n  for (element in this) {\n    val key = keySelector(element)\n    val list =
destination.getOrPut(key) { ArrayList<V>() }\n    list.add(valueTransform(element))\n  }\n  return
destination\n}\n\n/**\n * Creates a [Grouping] source from a char sequence to be used later with one of group-and-
fold operations\n * using the specified [keySelector] function to extract a key from each character.\n * \n * @sample
samples.collections.Grouping.groupingByEachCount\n *\n@SinceKotlin("1.1")\npublic inline fun <K>
CharSequence.groupingBy(crossinline keySelector: (Char) -> K): Grouping<Char, K> {\n  return object :
Grouping<Char, K> {\n    override fun sourceIterator(): Iterator<Char> = this@groupingBy.iterator()\n
    override fun keyOf(element: Char): K = keySelector(element)\n  }\n}\n\n/**\n * Returns a list containing the
results of applying the given [transform] function\n * to each character in the original char sequence.\n * \n *

```

```

@sample samples.text.Strings.map\n *\npublic inline fun <R> CharSequence.map(transform: (Char) -> R):
List<R> {\n  return mapTo(ArrayList<R>(length), transform)\n}\n\n/**\n * Returns a list containing the results of
applying the given [transform] function\n * to each character and its index in the original char sequence.\n *
@param [transform] function that takes the index of a character and the character itself\n * and returns the result of
the transform applied to the character.\n *\npublic inline fun <R> CharSequence.mapIndexed(transform: (index:
Int, Char) -> R): List<R> {\n  return mapIndexedTo(ArrayList<R>(length), transform)\n}\n\n/**\n * Returns a list
containing only the non-null results of applying the given [transform] function\n * to each character and its index in
the original char sequence.\n * @param [transform] function that takes the index of a character and the character
itself\n * and returns the result of the transform applied to the character.\n *\npublic inline fun <R : Any>
CharSequence.mapIndexedNotNull(transform: (index: Int, Char) -> R?): List<R> {\n  return
mapIndexedNotNullTo(ArrayList<R>(), transform)\n}\n\n/**\n * Applies the given [transform] function to each
character and its index in the original char sequence\n * and appends only the non-null results to the given
[destination].\n * @param [transform] function that takes the index of a character and the character itself\n * and
returns the result of the transform applied to the character.\n *\npublic inline fun <R : Any, C :
MutableCollection<in R>> CharSequence.mapIndexedNotNullTo(destination: C, transform: (index: Int, Char) ->
R?): C {\n  forEachIndexed { index, element -> transform(index, element)?.let { destination.add(it) } }\n  return
destination\n}\n\n/**\n * Applies the given [transform] function to each character and its index in the original char
sequence\n * and appends the results to the given [destination].\n * @param [transform] function that takes the
index of a character and the character itself\n * and returns the result of the transform applied to the character.\n
*\npublic inline fun <R, C : MutableCollection<in R>> CharSequence.mapIndexedTo(destination: C, transform:
(index: Int, Char) -> R): C {\n  var index = 0\n  for (item in this)\n    destination.add(transform(index++,
item))\n  return destination\n}\n\n/**\n * Returns a list containing only the non-null results of applying the given
[transform] function\n * to each character in the original char sequence.\n * \n * @sample
samples.collections.Collections.Transformations.mapNotNull\n *\npublic inline fun <R : Any>
CharSequence.mapNotNull(transform: (Char) -> R?): List<R> {\n  return mapNotNullTo(ArrayList<R>(),
transform)\n}\n\n/**\n * Applies the given [transform] function to each character in the original char sequence\n *
and appends only the non-null results to the given [destination].\n *\npublic inline fun <R : Any, C :
MutableCollection<in R>> CharSequence.mapNotNullTo(destination: C, transform: (Char) -> R?): C {\n  forEach
{ element -> transform(element)?.let { destination.add(it) } }\n  return destination\n}\n\n/**\n * Applies the given
[transform] function to each character of the original char sequence\n * and appends the results to the given
[destination].\n *\npublic inline fun <R, C : MutableCollection<in R>> CharSequence.mapTo(destination: C,
transform: (Char) -> R): C {\n  for (item in this)\n    destination.add(transform(item))\n  return
destination\n}\n\n/**\n * Returns a lazy [Iterable] that wraps each character of the original char sequence\n * into an
[IndexValue] containing the index of that character and the character itself.\n *\npublic fun
CharSequence.withIndex(): Iterable<IndexedValue<Char>> {\n  return IndexingIterable { iterator() }\n}\n\n/**\n *
Returns `true` if all characters match the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.all\n *\npublic inline fun CharSequence.all(predicate: (Char) ->
Boolean): Boolean {\n  for (element in this) if (!predicate(element)) return false\n  return true\n}\n\n/**\n *
Returns `true` if char sequence has at least one character.\n * \n * @sample
samples.collections.Collections.Aggregates.any\n *\npublic fun CharSequence.any(): Boolean {\n  return
!isEmpty()\n}\n\n/**\n * Returns `true` if at least one character matches the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.anyWithPredicate\n *\npublic inline fun CharSequence.any(predicate:
(Char) -> Boolean): Boolean {\n  for (element in this) if (predicate(element)) return true\n  return
false\n}\n\n/**\n * Returns the length of this char sequence.\n *\npublic inline fun
CharSequence.count(): Int {\n  return length\n}\n\n/**\n * Returns the number of characters matching the given
[predicate].\n *\npublic inline fun CharSequence.count(predicate: (Char) -> Boolean): Int {\n  var count = 0\n
for (element in this) if (predicate(element)) ++count\n  return count\n}\n\n/**\n * Accumulates value starting with
[initial] value and applying [operation] from left to right\n * to current accumulator value and each character.\n * \n

```

* Returns the specified [initial] value if the char sequence is empty.\n * \n * @param [operation] function that takes current accumulator value and a character, and calculates the next accumulator value.\n */\npublic inline fun <R> CharSequence.fold(initial: R, operation: (acc: R, Char) -> R): R {\n var accumulator = initial\n for (element in this) accumulator = operation(accumulator, element)\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and each character with its index in the original char sequence.\n * \n * Returns the specified [initial] value if the char sequence is empty.\n * \n * @param [operation] function that takes the index of a character, current accumulator value\n * and the character itself, and calculates the next accumulator value.\n */\npublic inline fun <R> CharSequence.foldIndexed(initial: R, operation: (index: Int, acc: R, Char) -> R): R {\n var index = 0\n var accumulator = initial\n for (element in this) accumulator = operation(index++, accumulator, element)\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to left\n * to each character and current accumulator value.\n * \n * Returns the specified [initial] value if the char sequence is empty.\n * \n * @param [operation] function that takes a character and current accumulator value, and calculates the next accumulator value.\n */\npublic inline fun <R> CharSequence.foldRight(initial: R, operation: (Char, acc: R) -> R): R {\n var index = lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator = operation(get(index--), accumulator)\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to left\n * to each character with its index in the original char sequence and current accumulator value.\n * \n * Returns the specified [initial] value if the char sequence is empty.\n * \n * @param [operation] function that takes the index of a character, the character itself\n * and current accumulator value, and calculates the next accumulator value.\n */\npublic inline fun <R> CharSequence.foldRightIndexed(initial: R, operation: (index: Int, Char, acc: R) -> R): R {\n var index = lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator = operation(index, get(index), accumulator)\n --index\n }\n return accumulator\n}\n\n/**\n * Performs the given [action] on each character.\n */\npublic inline fun CharSequence.forEach(action: (Char) -> Unit): Unit {\n for (element in this) action(element)\n}\n\n/**\n * Performs the given [action] on each character, providing sequential index with the character.\n * \n * @param [action] function that takes the index of a character and the character itself\n * and performs the action on the character.\n */\npublic inline fun CharSequence.forEachIndexed(action: (index: Int, Char) -> Unit): Unit {\n var index = 0\n for (item in this) action(index++, item)\n}\n\n@Deprecated("Use maxOrNull instead.", ReplaceWith("this.maxOrNull()"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5", hiddenSince = "1.6")\npublic fun CharSequence.max(): Char? {\n return maxOrNull()\n}\n\n@Deprecated("Use maxByOrNull instead.", ReplaceWith("this.maxByOrNull(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5", hiddenSince = "1.6")\npublic inline fun <R : Comparable<R>> CharSequence.maxBy(selector: (Char) -> R): Char? {\n return maxByOrNull(selector)\n}\n\n/**\n * Returns the first character yielding the largest value of the given function or `null` if there are no characters.\n * \n * \n * @sample samples.collections.Collections.Aggregates.maxByOrNull\n */\n@SinceKotlin("1.4")\npublic inline fun <R : Comparable<R>> CharSequence.maxByOrNull(selector: (Char) -> R): Char? {\n if (isEmpty()) return null\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return maxElem\n var maxValue = selector(maxElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n }\n return maxElem\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each character in the char sequence.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the char sequence is empty.\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.maxOf(selector: (Char) -> Double): Double {\n if (isEmpty()) throw NoSuchElementException()\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to

each character in the char sequence.
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.
 * @throws NoSuchElementException if the char sequence is empty.

```

*\/n@SinceKotlin("1.4")n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)n@OverloadResolution
ByLambdaReturnTypen@kotlin.internal.InlineOnlynpublic inline fun CharSequence.maxOf(selector: (Char) ->
Float): Float {n if (isEmpty()) throw NoSuchElementException()n var maxValue = selector(this[0])n for (i
in 1..lastIndex) {n val v = selector(this[i])n maxValue = maxOf(maxValue, v)n }n return
maxValue}n/n/n/**n * Returns the largest value among all values produced by [selector] functionn * applied to
each character in the char sequence.n * @throws NoSuchElementException if the char sequence is empty.n
*/n@SinceKotlin("1.4")n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)n@OverloadResolution
ByLambdaReturnTypen@kotlin.internal.InlineOnlynpublic inline fun <R : Comparable<R>>
CharSequence.maxOf(selector: (Char) -> R): R {n if (isEmpty()) throw NoSuchElementException()n var
maxValue = selector(this[0])n for (i in 1..lastIndex) {n val v = selector(this[i])n if (maxValue < v) {n
maxValue = v\n }n }n return maxValue}n/n/n/**n * Returns the largest value among all values
produced by [selector] functionn * applied to each character in the char sequence or `null` if there are no
characters.n * @throws NoSuchElementException if the char sequence is empty.n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.n
*/n@SinceKotlin("1.4")n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)n@OverloadResolution
ByLambdaReturnTypen@kotlin.internal.InlineOnlynpublic inline fun CharSequence.maxOfOrNull(selector:
(Char) -> Double): Double? {n if (isEmpty()) return nulln var maxValue = selector(this[0])n for (i in
1..lastIndex) {n val v = selector(this[i])n maxValue = maxOf(maxValue, v)n }n return
maxValue}n/n/n/**n * Returns the largest value among all values produced by [selector] functionn * applied to
each character in the char sequence or `null` if there are no characters.n * @throws NoSuchElementException if the char sequence is empty.n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.n
*/n@SinceKotlin("1.4")n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)n@OverloadResolution
ByLambdaReturnTypen@kotlin.internal.InlineOnlynpublic inline fun CharSequence.maxOfOrNull(selector:
(Char) -> Float): Float? {n if (isEmpty()) return nulln var maxValue = selector(this[0])n for (i in
1..lastIndex) {n val v = selector(this[i])n maxValue = maxOf(maxValue, v)n }n return
maxValue}n/n/n/**n * Returns the largest value among all values produced by [selector] functionn * applied to
each character in the char sequence or `null` if there are no characters.n
*/n@SinceKotlin("1.4")n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)n@OverloadResolution
ByLambdaReturnTypen@kotlin.internal.InlineOnlynpublic inline fun <R : Comparable<R>>
CharSequence.maxOfOrNull(selector: (Char) -> R): R? {n if (isEmpty()) return nulln var maxValue =
selector(this[0])n for (i in 1..lastIndex) {n val v = selector(this[i])n if (maxValue < v) {n
maxValue = v\n }n }n return maxValue}n/n/n/**n * Returns the largest value according to the provided
[comparator]n * among all values produced by [selector] function applied to each character in the char sequence.n
* @throws NoSuchElementException if the char sequence is empty.n
*/n@SinceKotlin("1.4")n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)n@OverloadResolution
ByLambdaReturnTypen@kotlin.internal.InlineOnlynpublic inline fun <R> CharSequence.maxOfWith(comparator:
Comparator<in R>, selector: (Char) -> R): R {n if (isEmpty()) throw NoSuchElementException()n var
maxValue = selector(this[0])n for (i in 1..lastIndex) {n val v = selector(this[i])n if
(comparator.compare(maxValue, v) < 0) {n maxValue = v\n }n }n return maxValue}n/n/n/**n *
Returns the largest value according to the provided [comparator]n * among all values produced by [selector]
function applied to each character in the char sequence or `null` if there are no characters.n
*/n@SinceKotlin("1.4")n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)n@OverloadResolution
ByLambdaReturnTypen@kotlin.internal.InlineOnlynpublic inline fun <R>
CharSequence.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Char) -> R): R? {n if (isEmpty())
return nulln var maxValue = selector(this[0])n for (i in 1..lastIndex) {n val v = selector(this[i])n if
(comparator.compare(maxValue, v) < 0) {n maxValue = v\n }n }n return maxValue}n/n/n/**n *
Returns the largest character or `null` if there are no characters.n
*/n@SinceKotlin("1.4")npublic fun

```

```

CharSequence.maxOrNull(): Char? {\n  if (isEmpty()) return null\n  var max = this[0]\n  for (i in 1..lastIndex)
{\n    val e = this[i]\n    if (max < e) max = e\n  }\n  return max\n}\n\n@Deprecated("Use maxWithOrNull
instead.", ReplaceWith("this.maxWithOrNull(comparator)"))\n@DeprecatedSinceKotlin(warningSince = "1.4",
errorSince = "1.5", hiddenSince = "1.6")\npublic fun CharSequence.maxWith(comparator: Comparator<in
Char>): Char? {\n  return maxWithOrNull(comparator)\n}\n\n/**\n * Returns the first character having the largest
value according to the provided [comparator] or `null` if there are no characters.\n
*\n *\n */\n@SinceKotlin("1.4")\npublic fun CharSequence.maxWithOrNull(comparator: Comparator<in Char>): Char?
{\n  if (isEmpty()) return null\n  var max = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if
(comparator.compare(max, e) < 0) max = e\n  }\n  return max\n}\n\n@Deprecated("Use minOrNull instead.",
ReplaceWith("this.minOrNull()"))\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5",
hiddenSince = "1.6")\npublic fun CharSequence.min(): Char? {\n  return minOrNull()\n}\n\n@Deprecated("Use
minByOrNull instead.", ReplaceWith("this.minByOrNull(selector)"))\n@DeprecatedSinceKotlin(warningSince =
"1.4", errorSince = "1.5", hiddenSince = "1.6")\npublic inline fun <R : Comparable<R>>
CharSequence.minBy(selector: (Char) -> R): Char? {\n  return minByOrNull(selector)\n}\n\n/**\n * Returns the
first character yielding the smallest value of the given function or `null` if there are no characters.\n
*\n * @sample
samples.collections.Collections.Aggregates.minByOrNull\n *\n */\n@SinceKotlin("1.4")\npublic inline fun <R :
Comparable<R>> CharSequence.minByOrNull(selector: (Char) -> R): Char? {\n  if (isEmpty()) return null\n  var
minElem = this[0]\n  val lastIndex = this.lastIndex\n  if (lastIndex == 0) return minElem\n  var minValue =
selector(minElem)\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    val v = selector(e)\n    if (minValue > v)
{\n      minElem = e\n      minValue = v\n    }\n  }\n  return minElem\n}\n\n/**\n * Returns the smallest
value among all values produced by [selector] function\n * applied to each character in the char sequence.\n
*\n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*\n * @throws
NoSuchElementException if the char sequence is empty.\n
*\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.minOf(selector: (Char) ->
Double): Double {\n  if (isEmpty()) throw NoSuchElementException()\n  var minValue = selector(this[0])\n  for
(i in 1..lastIndex) {\n    val v = selector(this[i])\n    minValue = minOf(minValue, v)\n  }\n  return
minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to
each character in the char sequence.\n
*\n * If any of values produced by [selector] function is `NaN`, the returned
result is `NaN`.\n
*\n * @throws NoSuchElementException if the char sequence is empty.\n
*\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.minOf(selector: (Char) ->
Float): Float {\n  if (isEmpty()) throw NoSuchElementException()\n  var minValue = selector(this[0])\n  for (i in
1..lastIndex) {\n    val v = selector(this[i])\n    minValue = minOf(minValue, v)\n  }\n  return
minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to
each character in the char sequence.\n
*\n * @throws NoSuchElementException if the char sequence is empty.\n
*\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
CharSequence.minOf(selector: (Char) -> R): R {\n  if (isEmpty()) throw NoSuchElementException()\n  var
minValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if (minValue > v) {\n
    minValue = v\n    }\n  }\n  return minValue\n}\n\n/**\n * Returns the smallest value among all values
produced by [selector] function\n * applied to each character in the char sequence or `null` if there are no
characters.\n
*\n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.minOfOrNull(selector:
(Char) -> Double): Double? {\n  if (isEmpty()) return null\n  var minValue = selector(this[0])\n  for (i in
1..lastIndex) {\n    val v = selector(this[i])\n    minValue = minOf(minValue, v)\n  }\n  return
minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to

```

each character in the char sequence or `null` if there are no characters.

`* If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.`

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.minOrNull(selector:
(Char) -> Float): Float? {\n    if (isEmpty()) return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex)
{\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n *
Returns the smallest value among all values produced by [selector] function\n * applied to each character in the char
sequence or `null` if there are no characters.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
CharSequence.minOrNull(selector: (Char) -> R): R? {\n    if (isEmpty()) return null\n    var minValue =
selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (minValue > v) {\n
minValue = v\n        }\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value according to the provided
[comparator]\n * among all values produced by [selector] function applied to each character in the char sequence.\n
\n * \n * @throws NoSuchElementException if the char sequence is empty.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> CharSequence.minOfWith(comparator:
Comparator<in R>, selector: (Char) -> R): R {\n    if (isEmpty()) throw NoSuchElementException()\n    var
minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if
(comparator.compare(minValue, v) > 0) {\n            minValue = v\n        }\n    }\n    return minValue\n}\n\n/**\n *
Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each character in the char sequence or `null` if there are no characters.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>
CharSequence.minOfWithOrNull(comparator: Comparator<in R>, selector: (Char) -> R): R? {\n    if (isEmpty())
return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n
if (comparator.compare(minValue, v) > 0) {\n            minValue = v\n        }\n    }\n    return minValue\n}\n\n/**\n *
Returns the smallest character or `null` if there are no characters.\n
```

```
*\n@SinceKotlin("1.4")\npublic fun
CharSequence.minOrNull(): Char? {\n    if (isEmpty()) return null\n    var min = this[0]\n    for (i in 1..lastIndex) {\n
val e = this[i]\n    if (min > e) min = e\n    }\n    return min\n}\n\n@Deprecated("Use minWithOrNull
instead.", ReplaceWith("this.minWithOrNull(comparator)"))\n@DeprecatedSinceKotlin(warningSince = "1.4",
errorSince = "1.5", hiddenSince = "1.6")\npublic fun CharSequence.minWith(comparator: Comparator<in
Char>): Char? {\n    return minWithOrNull(comparator)\n}\n\n/**\n * Returns the first character having the smallest
value according to the provided [comparator] or `null` if there are no characters.\n
```

```
*\n@SinceKotlin("1.4")\npublic fun CharSequence.minWithOrNull(comparator: Comparator<in Char>): Char?
{\n    if (isEmpty()) return null\n    var min = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n
if (comparator.compare(min, e) > 0) min = e\n    }\n    return min\n}\n\n/**\n * Returns `true` if the char sequence has
no characters.\n * \n * @sample samples.collections.Collections.Aggregates.none\n
```

```
*\npublic fun
CharSequence.none(): Boolean {\n    return isEmpty()\n}\n\n/**\n * Returns `true` if no characters match the given
[predicate].\n * \n * @sample samples.collections.Collections.Aggregates.noneWithPredicate\n
```

```
*\npublic inline fun
CharSequence.none(predicate: (Char) -> Boolean): Boolean {\n    for (element in this) if (predicate(element)) return
false\n    return true\n}\n\n/**\n * Performs the given [action] on each character and returns the char sequence itself
afterwards.\n
```

```
*\n@SinceKotlin("1.1")\npublic inline fun <S : CharSequence> S.onEach(action: (Char) -> Unit): S
{\n    return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given [action] on each
character, providing sequential index with the character,\n * and returns the char sequence itself afterwards.\n *
\n * @param [action] function that takes the index of a character and the character itself\n * and performs the action on
the character.\n
```

```
*\n@SinceKotlin("1.4")\npublic inline fun <S : CharSequence> S.onEachIndexed(action: (index:
Int, Char) -> Unit): S {\n    return apply { forEachIndexed(action) }\n}\n\n/**\n * Accumulates value starting with
```

the first character and applying [operation] from left to right\n * to current accumulator value and each character.\n *
\n * Throws an exception if this char sequence is empty. If the char sequence can be empty in an expected way,\n *
please use [reduceOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function
that takes current accumulator value and a character,\n * and calculates the next accumulator value.\n * \n *

```
@sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun  

CharSequence.reduce(operation: (acc: Char, Char) -> Char): Char {\n    if (isEmpty())\n        throw  

UnsupportedOperationException("Empty char sequence can't be reduced.")\n    var accumulator = this[0]\n    for  

(index in 1..lastIndex) {\n        accumulator = operation(accumulator, this[index])\n    }\n    return  

accumulator\n}\n\n/**\n * Accumulates value starting with the first character and applying [operation] from left to  

right\n * to current accumulator value and each character with its index in the original char sequence.\n * \n *
```

Throws an exception if this char sequence is empty. If the char sequence can be empty in an expected way,\n *
please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation]
function that takes the index of a character, current accumulator value and the character itself,\n * and calculates the
next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun
CharSequence.reduceIndexed(operation: (index: Int, acc: Char, Char) -> Char): Char {\n if (isEmpty())\n throw
UnsupportedOperationException("Empty char sequence can't be reduced.")\n var accumulator = this[0]\n for
(index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return
accumulator\n}\n\n/**\n * Accumulates value starting with the first character and applying [operation] from left to
right\n * to current accumulator value and each character with its index in the original char sequence.\n * \n *

Returns `null` if the char sequence is empty.\n * \n * @param [operation] function that takes the index of a
character, current accumulator value and the character itself,\n * and calculates the next accumulator value.\n * \n *

```
@sample samples.collections.Collections.Aggregates.reduceOrNull\n */\n@SinceKotlin("1.4")\npublic inline fun  

CharSequence.reduceIndexedOrNull(operation: (index: Int, acc: Char, Char) -> Char): Char? {\n    if (isEmpty())\n        return null\n    var accumulator = this[0]\n    for (index in 1..lastIndex) {\n        accumulator = operation(index,  

accumulator, this[index])\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with the first  

character and applying [operation] from left to right\n * to current accumulator value and each character.\n * \n *
```

Returns `null` if the char sequence is empty.\n * \n * @param [operation] function that takes current accumulator
value and a character,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceOrNull\n

```
*/\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun  

CharSequence.reduceOrNull(operation: (acc: Char, Char) -> Char): Char? {\n    if (isEmpty())\n        return null\n    var  

accumulator = this[0]\n    for (index in 1..lastIndex) {\n        accumulator = operation(accumulator, this[index])\n    }\n    return  

accumulator\n}\n\n/**\n * Accumulates value starting with the last character and applying [operation]  

from right to left\n * to each character and current accumulator value.\n * \n *
```

Throws an exception if this char
sequence is empty. If the char sequence can be empty in an expected way,\n * please use [reduceRightOrNull]
instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes a character and
current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRight\n */\npublic inline fun
CharSequence.reduceRight(operation: (Char, acc: Char) -> Char): Char {\n var index = lastIndex\n if (index < 0)
throw
UnsupportedOperationException("Empty char sequence can't be reduced.")\n var accumulator = get(index--)\n while (index >= 0) {\n accumulator = operation(get(index--), accumulator)\n }\n return
accumulator\n}\n\n/**\n * Accumulates value starting with the last character and applying [operation] from right to
left\n * to each character with its index in the original char sequence and current accumulator value.\n * \n *

Throws an
exception if this char sequence is empty. If the char sequence can be empty in an expected way,\n * please use
[reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation]
function that takes the index of a character, the character itself and current accumulator value,\n * and calculates the
next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n */\npublic
inline fun
CharSequence.reduceRightIndexed(operation: (index: Int, Char, acc: Char) -> Char): Char {\n var index

```

= lastIndex\n  if (index < 0) throw UnsupportedOperationException("Empty char sequence can't be reduced.")\n
var accumulator = get(index--)\n  while (index >= 0) {\n    accumulator = operation(index, get(index),\n
accumulator)\n    --index\n  }\n  return accumulator\n}\n\n/**\n * Accumulates value starting with the last\n
character and applying [operation] from right to left\n * to each character with its index in the original char sequence\n
and current accumulator value.\n * \n * Returns `null` if the char sequence is empty.\n * \n * @param [operation]\n
function that takes the index of a character, the character itself and current accumulator value,\n * and calculates the\n
next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRightOrNull\n
*\n@SinceKotlin("1.4")\npublic inline fun CharSequence.reduceRightIndexedOrNull(operation: (index: Int, Char,\n
acc: Char) -> Char): Char? {\n  var index = lastIndex\n  if (index < 0) return null\n  var accumulator = get(index--)\n
while (index >= 0) {\n    accumulator = operation(index, get(index), accumulator)\n    --index\n  }\n
return accumulator\n}\n\n/**\n * Accumulates value starting with the last character and applying [operation] from\n
right to left\n * to each character and current accumulator value.\n * \n * Returns `null` if the char sequence is\n
empty.\n * \n * @param [operation] function that takes a character and current accumulator value,\n * and calculates the\n
next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRightOrNull\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun\n
CharSequence.reduceRightOrNull(operation: (Char, acc: Char) -> Char): Char? {\n  var index = lastIndex\n  if\n
(index < 0) return null\n  var accumulator = get(index--)\n  while (index >= 0) {\n    accumulator =\n
operation(get(index--), accumulator)\n  }\n  return accumulator\n}\n\n/**\n * Returns a list containing successive\n
accumulation values generated by applying [operation] from left to right\n * to each character and current\n
accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should\n
not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation]\n
function that takes current accumulator value and a character, and calculates the next accumulator value.\n * \n *\n
@sample samples.collections.Collections.Aggregates.runningFold\n *\n@SinceKotlin("1.4")\npublic inline fun\n
<R> CharSequence.runningFold(initial: R, operation: (acc: R, Char) -> R): List<R> {\n  if (isEmpty()) return\n
listOf(initial)\n  val result = ArrayList<R>(length + 1).apply { add(initial) }\n  var accumulator = initial\n  for\n
(element in this) {\n    accumulator = operation(accumulator, element)\n    result.add(accumulator)\n  }\n
return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying\n
[operation] from left to right\n * to each character, its index in the original char sequence and current accumulator\n
value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be\n
mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that\n
takes the index of a character, current accumulator value\n * and the character itself, and calculates the next\n
accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n
*\n@SinceKotlin("1.4")\npublic inline fun <R> CharSequence.runningFoldIndexed(initial: R, operation: (index:\n
Int, acc: R, Char) -> R): List<R> {\n  if (isEmpty()) return listOf(initial)\n  val result = ArrayList<R>(length +\n
1).apply { add(initial) }\n  var accumulator = initial\n  for (index in indices) {\n    accumulator =\n
operation(index, accumulator, this[index])\n    result.add(accumulator)\n  }\n  return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to\n
each character and current accumulator value that starts with the first character of this char sequence.\n * \n * Note\n
that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous\n
value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and a character,\n
and calculates the next accumulator value.\n * \n * @sample\n
samples.collections.Collections.Aggregates.runningReduce\n *\n@SinceKotlin("1.4")\npublic inline fun\n
CharSequence.runningReduce(operation: (acc: Char, Char) -> Char): List<Char> {\n  if (isEmpty()) return\n
emptyList()\n  var accumulator = this[0]\n  val result = ArrayList<Char>(length).apply { add(accumulator) }\n
for (index in 1 until length) {\n    accumulator = operation(accumulator, this[index])\n
result.add(accumulator)\n  }\n  return result\n}\n\n/**\n * Returns a list containing successive accumulation\n
values generated by applying [operation] from left to right\n * to each character, its index in the original char\n
sequence and current accumulator value that starts with the first character of this char sequence.\n * \n * Note that

```

`acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of a character, current accumulator value\n * and the character itself, and calculates the next accumulator value.\n * \n * @sample

```
samples.collections.Collections.Aggregates.runningReduce\n *^\n@SinceKotlin("1.4")\npublic inline fun
CharSequence.runningReduceIndexed(operation: (index: Int, acc: Char, Char) -> Char): List<Char> {\n  if
(isEmpty()) return emptyList()\n  var accumulator = this[0]\n  val result = ArrayList<Char>(length).apply {
add(accumulator) }\n  for (index in 1 until length) {\n    accumulator = operation(index, accumulator,
this[index])\n    result.add(accumulator)\n  }\n  return result\n}\n\n/**\n * Returns a list containing successive
accumulation values generated by applying [operation] from left to right\n * to each character and current
accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should
not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation]
function that takes current accumulator value and a character, and calculates the next accumulator value.\n * \n *
@sample samples.collections.Collections.Aggregates.scan\n
*^\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun <R>
CharSequence.scan(initial: R, operation: (acc: R, Char) -> R): List<R> {\n  return runningFold(initial,
operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation]
from left to right\n * to each character, its index in the original char sequence and current accumulator value that
starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n *
otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the
index of a character, current accumulator value\n * and the character itself, and calculates the next accumulator
value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n
*^\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun <R>
CharSequence.scanIndexed(initial: R, operation: (index: Int, acc: R, Char) -> R): List<R> {\n  return
runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each character in the char sequence.\n *^\n@Deprecated("Use sumOf instead.")
ReplaceWith("this.sumOf(selector)")\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic inline fun
CharSequence.sumBy(selector: (Char) -> Int): Int {\n  var sum: Int = 0\n  for (element in this) {\n    sum +=
selector(element)\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each character in the char sequence.\n *^\n@Deprecated("Use sumOf instead.")
ReplaceWith("this.sumOf(selector)")\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic inline fun
CharSequence.sumByDouble(selector: (Char) -> Double): Double {\n  var sum: Double = 0.0\n  for (element in
this) {\n    sum += selector(element)\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each character in the char sequence.\n
*^\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfDouble")\n@kotlin.internal.InlineOnly\npublic inline fun
CharSequence.sumOf(selector: (Char) -> Double): Double {\n  var sum: Double = 0.toDouble()\n  for (element in
this) {\n    sum += selector(element)\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each character in the char sequence.\n
*^\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfInt")\n@kotlin.internal.InlineOnly\npublic inline fun
CharSequence.sumOf(selector: (Char) -> Int): Int {\n  var sum: Int = 0.toInt()\n  for (element in this) {\n    sum
+= selector(element)\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector]
function applied to each character in the char sequence.\n
*^\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfLong")\n@kotlin.internal.InlineOnly\npublic inline fun
CharSequence.sumOf(selector: (Char) -> Long): Long {\n  var sum: Long = 0.toLong()\n  for (element in this) {\n
sum += selector(element)\n  }\n  return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each character in the char sequence.\n
```

```
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfUInt")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.sumOf(selector: (Char) -> UInt): UInt {\n    var sum: UInt = 0.toUInt()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n * Returns the sum of all values produced by [selector] function applied to each character in the char sequence.
```

```
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfULong")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.sumOf(selector: (Char) -> ULong):
```

```
ULong {\n    var sum: ULong = 0.toULong()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n * Splits this char sequence into a list of strings each not exceeding the given [size].\n * The last string in the resulting list may have fewer characters than the given [size].\n * @param size the number of elements to take in each string, must be positive and can be greater than the number of elements in this char sequence.
```

```
*\n@Sample\nsamples.text.Strings.chunked\n*\n@SinceKotlin("1.2")\npublic fun CharSequence.chunked(size: Int): List<String> {\n    return windowed(size, size, partialWindows = true)\n}\n\n * Splits this char sequence into several char sequences each not exceeding the given [size]\n * and applies the given [transform] function to an each.\n * @return list of results of the [transform] applied to an each char sequence.
```

```
*\n * Note that the char sequence passed to the [transform] function is ephemeral and is valid only inside that function.\n * You should not store it or allow it to escape in some way, unless you made a snapshot of it.\n * The last char sequence may have fewer characters than the given [size].\n * @param size the number of elements to take in each char sequence, must be positive and can be greater than the number of elements in this char sequence.
```

```
*\n@Sample\nsamples.text.Strings.chunkedTransform\n*\n@SinceKotlin("1.2")\npublic fun <R> CharSequence.chunked(size: Int, transform: (CharSequence) -> R): List<R> {\n    return windowed(size, size, partialWindows = true, transform = transform)\n}\n\n * Splits this char sequence into a sequence of strings each not exceeding the given [size].\n * The last string in the resulting sequence may have fewer characters than the given [size].\n * @param size the number of elements to take in each string, must be positive and can be greater than the number of elements in this char sequence.
```

```
*\n@Sample\nsamples.collections.Collections.Transformations.chunked\n*\n@SinceKotlin("1.2")\npublic fun CharSequence.chunkedSequence(size: Int): Sequence<String> {\n    return chunkedSequence(size) { it.toString() }\n}\n\n * Splits this char sequence into several char sequences each not exceeding the given [size]\n * and applies the given [transform] function to an each.\n * @return sequence of results of the [transform] applied to an each char sequence.\n * Note that the char sequence passed to the [transform] function is ephemeral and is valid only inside that function.\n * You should not store it or allow it to escape in some way, unless you made a snapshot of it.\n * The last char sequence may have fewer characters than the given [size].\n * @param size the number of elements to take in each char sequence, must be positive and can be greater than the number of elements in this char sequence.
```

```
*\n@Sample\nsamples.text.Strings.chunkedTransformToSequence\n*\n@SinceKotlin("1.2")\npublic fun <R> CharSequence.chunkedSequence(size: Int, transform: (CharSequence) -> R): Sequence<R> {\n    return windowedSequence(size, size, partialWindows = true, transform =
```

```
transform)\n}\n\n * Splits the original char sequence into pair of char sequences,\n * where *first* char sequence contains characters for which [predicate] yielded `true`,\n * while *second* char sequence contains characters for which [predicate] yielded `false`.\n * @sample samples.text.Strings.partition\n*\npublic inline fun CharSequence.partition(predicate: (Char) -> Boolean): Pair<CharSequence, CharSequence> {\n    val first = StringBuilder()\n    val second = StringBuilder()\n    for (element in this) {\n        if (predicate(element)) {\n            first.append(element)\n        } else {\n            second.append(element)\n        }\n    }\n    return Pair(first,
```

```
second)\n}\n\n * Splits the original string into pair of strings,\n * where *first* string contains characters for which [predicate] yielded `true`,\n * while *second* string contains characters for which [predicate] yielded `false`.\n * @sample samples.text.Strings.partition\n*\npublic inline fun String.partition(predicate: (Char) -> Boolean): Pair<String, String> {\n    val first = StringBuilder()\n    val second = StringBuilder()\n    for (element in
```

```

this) {\n    if (predicate(element)) {\n        first.append(element)\n    } else {\n
second.append(element)\n    }\n }\n return Pair(first.toString(), second.toString())\n}\n\n/**\n * Returns a list
of snapshots of the window of the given [size]\n * sliding along this char sequence with the given [step], where
each\n * snapshot is a string.\n * \n * Several last strings may have fewer characters than the given [size].\n * \n *
Both [size] and [step] must be positive and can be greater than the number of elements in this char sequence.\n *
@param size the number of elements to take in each window\n * @param step the number of elements to move the
window forward by on an each step, by default 1\n * @param partialWindows controls whether or not to keep
partial windows in the end if any,\n * by default `false` which means partial windows won't be preserved\n * \n *
@sample samples.collections.Sequences.Transformations.takeWindows\n *\n@SinceKotlin("1.2")\npublic fun
CharSequence.windowed(size: Int, step: Int = 1, partialWindows: Boolean = false): List<String> {\n    return
windowed(size, step, partialWindows) { it.toString() }\n}\n\n/**\n * Returns a list of results of applying the given
[transform] function to\n * an each char sequence representing a view over the window of the given [size]\n *
sliding along this char sequence with the given [step].\n * \n * Note that the char sequence passed to the [transform]
function is ephemeral and is valid only inside that function.\n * You should not store it or allow it to escape in some
way, unless you made a snapshot of it.\n * Several last char sequences may have fewer characters than the given
[size].\n * \n * Both [size] and [step] must be positive and can be greater than the number of elements in this char
sequence.\n * @param size the number of elements to take in each window\n * @param step the number of
elements to move the window forward by on an each step, by default 1\n * @param partialWindows controls
whether or not to keep partial windows in the end if any,\n * by default `false` which means partial windows won't
be preserved\n * \n * @sample samples.collections.Sequences.Transformations.averageWindows\n
*\n@SinceKotlin("1.2")\npublic fun <R> CharSequence.windowed(size: Int, step: Int = 1, partialWindows:
Boolean = false, transform: (CharSequence) -> R): List<R> {\n    checkWindowSizeStep(size, step)\n    val thisSize
= this.length\n    val resultCapacity = thisSize / step + if (thisSize % step == 0) 0 else 1\n    val result =
ArrayList<R>(resultCapacity)\n    var index = 0\n    while (index in 0 until thisSize) {\n        val end = index + size\n
        val coercedEnd = if (end < 0 || end > thisSize) { if (partialWindows) thisSize else break } else end\n
        result.add(transform(subSequence(index, coercedEnd)))\n        index += step\n    }\n    return result\n}\n\n/**\n *
Returns a sequence of snapshots of the window of the given [size]\n * sliding along this char sequence with the
given [step], where each\n * snapshot is a string.\n * \n * Several last strings may have fewer characters than the
given [size].\n * \n * Both [size] and [step] must be positive and can be greater than the number of elements in this
char sequence.\n * @param size the number of elements to take in each window\n * @param step the number of
elements to move the window forward by on an each step, by default 1\n * @param partialWindows controls
whether or not to keep partial windows in the end if any,\n * by default `false` which means partial windows won't
be preserved\n * \n * @sample samples.collections.Sequences.Transformations.takeWindows\n
*\n@SinceKotlin("1.2")\npublic fun CharSequence.windowedSequence(size: Int, step: Int = 1, partialWindows:
Boolean = false): Sequence<String> {\n    return windowedSequence(size, step, partialWindows) { it.toString()
}\n}\n\n/**\n * Returns a sequence of results of applying the given [transform] function to\n * an each char
sequence representing a view over the window of the given [size]\n * sliding along this char sequence with the given
[step].\n * \n * Note that the char sequence passed to the [transform] function is ephemeral and is valid only inside
that function.\n * You should not store it or allow it to escape in some way, unless you made a snapshot of it.\n *
Several last char sequences may have fewer characters than the given [size].\n * \n * Both [size] and [step] must be
positive and can be greater than the number of elements in this char sequence.\n * @param size the number of
elements to take in each window\n * @param step the number of elements to move the window forward by on an
each step, by default 1\n * @param partialWindows controls whether or not to keep partial windows in the end if
any,\n * by default `false` which means partial windows won't be preserved\n * \n * @sample
samples.collections.Sequences.Transformations.averageWindows\n *\n@SinceKotlin("1.2")\npublic fun <R>
CharSequence.windowedSequence(size: Int, step: Int = 1, partialWindows: Boolean = false, transform:
(CharSequence) -> R): Sequence<R> {\n    checkWindowSizeStep(size, step)\n    val windows = (if
(partialWindows) indices else 0 until length - size + 1) step step\n    return windows.asSequence().map { index ->\n

```

```

    val end = index + size\n    val coercedEnd = if (end < 0 || end > length) length else end\n
transform(subSequence(index, coercedEnd))\n    }\n}\n\n/**\n * Returns a list of pairs built from the characters of
`this` and the [other] char sequences with the same index\n * The returned list has length of the shortest char
sequence.\n * \n * @sample samples.text.Strings.zip\n */\npublic infix fun CharSequence.zip(other: CharSequence):
List<Pair<Char, Char>> {\n    return zip(other) { c1, c2 -> c1 to c2 }\n}\n\n/**\n * Returns a list of values built
from the characters of `this` and the [other] char sequences with the same index\n * using the provided [transform]
function applied to each pair of characters.\n * The returned list has length of the shortest char sequence.\n * \n *
@sample samples.text.Strings.zipWithTransform\n */\npublic inline fun <V> CharSequence.zip(other:
CharSequence, transform: (a: Char, b: Char) -> V): List<V> {\n    val length = minOf(this.length, other.length)\n
val list = ArrayList<V>(length)\n    for (i in 0 until length) {\n        list.add(transform(this[i], other[i]))\n    }\n
return list\n}\n\n/**\n * Returns a list of pairs of each two adjacent characters in this char sequence.\n * \n * The
returned list is empty if this char sequence contains less than two characters.\n * \n * @sample
samples.collections.Collections.Transformations.zipWithNext\n */\n@SinceKotlin("1.2")\npublic fun
CharSequence.zipWithNext(): List<Pair<Char, Char>> {\n    return zipWithNext { a, b -> a to b }\n}\n\n/**\n *
Returns a list containing the results of applying the given [transform] function\n * to an each pair of two adjacent
characters in this char sequence.\n * \n * The returned list is empty if this char sequence contains less than two
characters.\n * \n * @sample samples.collections.Collections.Transformations.zipWithNextToFindDeltas\n
*/\n@SinceKotlin("1.2")\npublic inline fun <R> CharSequence.zipWithNext(transform: (a: Char, b: Char) -> R):
List<R> {\n    val size = length - 1\n    if (size < 1) return emptyList()\n    val result = ArrayList<R>(size)\n    for
(index in 0 until size) {\n        result.add(transform(this[index], this[index + 1]))\n    }\n    return result\n}\n\n/**\n *
Creates an [Iterable] instance that wraps the original char sequence returning its characters when being iterated.\n
*/\npublic fun CharSequence.asIterable(): Iterable<Char> {\n    if (this is String && isEmpty()) return emptyList()\n
return Iterable { this.iterator() }\n}\n\n/**\n * Creates a [Sequence] instance that wraps the original char sequence
returning its characters when being iterated.\n */\npublic fun CharSequence.asSequence(): Sequence<Char> {\n    if
(this is String && isEmpty()) return emptySequence()\n    return Sequence { this.iterator() }\n}\n\n"/**\n *
Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is
governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("StringsKt")\n\npackage
kotlin.text\nimport kotlin.contracts.contract\nimport kotlin.jvm.JvmName\n\n/**\n * Returns a copy of this string
converted to upper case using the rules of the default locale.\n */\n@Deprecated("Use uppercase() instead.",
ReplaceWith("uppercase()"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic expect fun
String.toUpperCase(): String\n\n/**\n * Returns a copy of this string converted to upper case using Unicode
mapping rules of the invariant locale.\n * \n * This function supports one-to-many and many-to-one character
mapping,\n * thus the length of the returned string can be different from the length of the original string.\n * \n *
@sample samples.text.Strings.uppercase\n
*/\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
String.toUpperCase(): String\n\n/**\n * Returns a copy of this string converted to lower case using the rules of the
default locale.\n */\n@Deprecated("Use lowercase() instead.",
ReplaceWith("lowercase()"))\n@DeprecatedSinceKotlin(warningSince = "1.5")\npublic expect fun
String.toLowerCase(): String\n\n/**\n * Returns a copy of this string converted to lower case using Unicode
mapping rules of the invariant locale.\n * \n * This function supports one-to-many and many-to-one character
mapping,\n * thus the length of the returned string can be different from the length of the original string.\n * \n *
@sample samples.text.Strings.lowercase\n
*/\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
String.toLowerCase(): String\n\n/**\n * Returns a copy of this string having its first letter titlecased using the rules of
the default locale.\n * or the original string if it's empty or already starts with a title case letter.\n * \n * The title case
of a character is usually the same as its upper case with several exceptions.\n * The particular list of characters with
the special title case form depends on the underlying platform.\n * \n * @sample samples.text.Strings.capitalize\n
*/\n

```

```

*^@Deprecated("Use replaceFirstChar instead.", ReplaceWith("replaceFirstChar { if (it.isLowerCase())
it.titlecase() else it.toString() }"))^@DeprecatedSinceKotlin(warningSince = "1.5")^public expect fun
String.capitalize(): String^n/**^n * Returns a copy of this string having its first letter lowercased using the rules of
the default locale,^n * or the original string if it's empty or already starts with a lower case letter.^n *^n * @sample
samples.text.Strings.decapiitalize^n/^@Deprecated("Use replaceFirstChar instead.",
ReplaceWith("replaceFirstChar { it.lowercase() }"))^@DeprecatedSinceKotlin(warningSince = "1.5")^public
expect fun String.decapiitalize(): String^n/**^n * Returns a sub sequence of this char sequence having leading and
trailing characters matching the [predicate] removed.^n *^npublic inline fun CharSequence.trim(predicate: (Char) ->
Boolean): CharSequence {^n var startIndex = 0^n var endIndex = length - 1^n var startFound = false^n
while (startIndex <= endIndex) {^n val index = if (!startFound) startIndex else endIndex^n val match =
predicate(this[index])^n if (!startFound) {^n if (!match)^n startFound = true^n else^n
startIndex += 1^n } else {^n if (!match)^n break^n else^n endIndex -= 1^n
}^n }^n return subSequence(startIndex, endIndex + 1)^n}^n/**^n * Returns a string having leading and
trailing characters matching the [predicate] removed.^n/^npublic inline fun String.trim(predicate: (Char) ->
Boolean): String =^n (this as CharSequence).trim(predicate).toString()^n/**^n * Returns a sub sequence of this
char sequence having leading characters matching the [predicate] removed.^n/^npublic inline fun
CharSequence.trimStart(predicate: (Char) -> Boolean): CharSequence {^n for (index in this.indices)^n if
(!predicate(this[index]))^n return subSequence(index, length)^n return ""^n}^n/**^n * Returns a string
having leading characters matching the [predicate] removed.^n/^npublic inline fun String.trimStart(predicate:
(Char) -> Boolean): String =^n (this as CharSequence).trimStart(predicate).toString()^n/**^n * Returns a sub
sequence of this char sequence having trailing characters matching the [predicate] removed.^n/^npublic inline fun
CharSequence.trimEnd(predicate: (Char) -> Boolean): CharSequence {^n for (index in this.indices.reversed())^n
if (!predicate(this[index]))^n return subSequence(0, index + 1)^n return ""^n}^n/**^n * Returns a string
having trailing characters matching the [predicate] removed.^n/^npublic inline fun String.trimEnd(predicate: (Char)
-> Boolean): String =^n (this as CharSequence).trimEnd(predicate).toString()^n/**^n * Returns a sub sequence of
this char sequence having leading and trailing characters from the [chars] array removed.^n/^npublic fun
CharSequence.trim(vararg chars: Char): CharSequence = trim { it in chars }^n/**^n * Returns a string having
leading and trailing characters from the [chars] array removed.^n/^npublic fun String.trim(vararg chars: Char):
String = trim { it in chars }^n/**^n * Returns a sub sequence of this char sequence having leading characters from
the [chars] array removed.^n/^npublic fun CharSequence.trimStart(vararg chars: Char): CharSequence = trimStart {
it in chars }^n/**^n * Returns a string having leading characters from the [chars] array removed.^n/^npublic fun
String.trimStart(vararg chars: Char): String = trimStart { it in chars }^n/**^n * Returns a sub sequence of this char
sequence having trailing characters from the [chars] array removed.^n/^npublic fun CharSequence.trimEnd(vararg
chars: Char): CharSequence = trimEnd { it in chars }^n/**^n * Returns a string having trailing characters from the
[chars] array removed.^n/^npublic fun String.trimEnd(vararg chars: Char): String = trimEnd { it in chars }^n/**^n
* Returns a sub sequence of this char sequence having leading and trailing whitespace removed.^n/^npublic fun
CharSequence.trim(): CharSequence = trim(Char::isWhitespace)^n/**^n * Returns a string having leading and
trailing whitespace removed.^n/^n@kotlin.internal.InlineOnly^npublic inline fun String.trim(): String = (this as
CharSequence).trim().toString()^n/**^n * Returns a sub sequence of this char sequence having leading whitespace
removed.^n/^npublic fun CharSequence.trimStart(): CharSequence = trimStart(Char::isWhitespace)^n/**^n *
Returns a string having leading whitespace removed.^n/^n@kotlin.internal.InlineOnly^npublic inline fun
String.trimStart(): String = (this as CharSequence).trimStart().toString()^n/**^n * Returns a sub sequence of this
char sequence having trailing whitespace removed.^n/^npublic fun CharSequence.trimEnd(): CharSequence =
trimEnd(Char::isWhitespace)^n/**^n * Returns a string having trailing whitespace removed.^n
*^n@kotlin.internal.InlineOnly^npublic inline fun String.trimEnd(): String = (this as
CharSequence).trimEnd().toString()^n/**^n * Returns a char sequence with content of this char sequence padded at
the beginning^n * to the specified [length] with the specified character or space.^n *^n * @param length the desired
string length.^n * @param padChar the character to pad string with, if it has length less than the [length] specified.

```

Space is used by default.

```

@return Returns a char sequence of length at least [length] consisting of `this` char
sequence prepended with [padChar] as many times
as are necessary to reach that length.
@sample samples.text.Strings.padStart
public fun CharSequence.padStart(length: Int, padChar: Char = ' '):
CharSequence {
    if (length < 0) throw IllegalArgumentException("Desired length $length is less than
zero.")
    if (length <= this.length) return this.subSequence(0, this.length)
    val sb =
StringBuilder(length)
    for (i in 1..(length - this.length)) sb.append(padChar)
    sb.append(this)
    return
sb
}

```

Pads the string to the specified [length] at the beginning with the specified character or space.

@param length the desired string length.

@param padChar the character to pad string with, if it has length less than the [length] specified. Space is used by default.

@return Returns a string of length at least [length] consisting of `this` string prepended with [padChar] as many times as are necessary to reach that length.

```

@sample samples.text.Strings.padStart
public fun String.padStart(length: Int, padChar: Char = ' '): String =
(this as CharSequence).padStart(length, padChar).toString()

```

Returns a char sequence with content of this char sequence padded at the end to the specified [length] with the specified character or space.

@param length the desired string length.

@param padChar the character to pad string with, if it has length less than the [length] specified. Space is used by default.

@return Returns a char sequence of length at least [length] consisting of `this` char sequence appended with [padChar] as many times as are necessary to reach that length.

@sample samples.text.Strings.padEnd

```

public fun CharSequence.padEnd(length: Int, padChar: Char = ' '): CharSequence {
    if (length < 0) throw IllegalArgumentException("Desired length $length is less
than zero.")
    if (length <= this.length) return this.subSequence(0, this.length)
    val sb =
StringBuilder(length)
    sb.append(this)
    for (i in 1..(length - this.length)) sb.append(padChar)
    return
sb
}

```

Pads the string to the specified [length] at the end with the specified character or space.

@param length the desired string length.

@param padChar the character to pad string with, if it has length less than the [length] specified. Space is used by default.

@return Returns a string of length at least [length] consisting of `this` string appended with [padChar] as many times as are necessary to reach that length.

@sample samples.text.Strings.padEnd

```

public fun String.padEnd(length: Int, padChar: Char = ' '): String =
(this as CharSequence).padEnd(length, padChar).toString()

```

Returns `true` if this nullable char sequence is either `null` or empty.

@sample samples.text.Strings.stringIsNullOrEmpty

```

@kotlin.internal.InlineOnly
public inline fun CharSequence?.isNullOrEmpty(): Boolean {
    contract {
        returns(false) implies (this@isNullOrEmpty != null)
    }
    return this == null || this.length == 0
}

```

Returns `true` if this char sequence is empty (contains no characters).

@sample samples.text.Strings.stringIsEmpty

```

@kotlin.internal.InlineOnly
public inline fun CharSequence.isEmpty(): Boolean = length == 0

```

Returns `true` if this char sequence is not empty.

@sample samples.text.Strings.stringIsNotEmpty

```

@kotlin.internal.InlineOnly
public inline fun CharSequence.isNotEmpty(): Boolean = length > 0

```

implemented differently in JVM and JS

```

public fun String.isBlank(): Boolean = length() == 0 || all { it.isWhitespace() }

```

Returns `true` if this char sequence is not empty and contains some characters except of whitespace characters.

@sample samples.text.Strings.stringIsNotBlank

```

@kotlin.internal.InlineOnly
public inline fun CharSequence.isNotBlank(): Boolean = !isBlank()

```

Returns `true` if this nullable char sequence is either `null` or empty or consists solely of whitespace characters.

@sample samples.text.Strings.stringOrNullOrBlank

```

@kotlin.internal.InlineOnly
public inline fun CharSequence?.isNullOrBlank(): Boolean {
    contract {
        returns(false) implies (this@isNullOrBlank != null)
    }
    return this == null || this.isBlank()
}

```

Iterator for characters of the given char sequence.

```

public operator fun CharSequence.iterator(): CharIterator = object : CharIterator() {
    private var index = 0
    public override fun nextChar(): Char = get(index++)
    public override fun hasNext(): Boolean = index < length
}

```

Returns the string if it is not `null`, or the empty string otherwise.

```

@kotlin.internal.InlineOnly
public inline fun String?.orEmpty(): String = this ?: ""

```

Returns this char sequence if it's not empty or the result of calling [defaultValue] function if the char sequence is empty.

@sample samples.text.Strings.stringIfEmpty

```

*^@SinceKotlin("1.3")^@kotlin.internal.InlineOnly^public inline fun <C, R> C.isEmpty(defaultValue: () -> R): R where C : CharSequence, C : R =\n    if (isEmpty()) defaultValue() else this\n/**\n * Returns this char sequence if it is not empty and doesn't consist solely of whitespace characters,\n * or the result of calling [defaultValue] function otherwise.\n */\n * @sample samples.text.Strings.stringfBlank\n
*^@SinceKotlin("1.3")^@kotlin.internal.InlineOnly^public inline fun <C, R> C.isBlank(defaultValue: () -> R): R where C : CharSequence, C : R =\n    if (isBlank()) defaultValue() else this\n/**\n * Returns the range of valid character indices for this char sequence.\n */\n^public val CharSequence.indices: IntRange\n    get() = 0..length - 1\n/**\n * Returns the index of the last character in the char sequence or -1 if it is empty.\n */\n^public val CharSequence.lastIndex: Int\n    get() = this.length - 1\n/**\n * Returns `true` if this CharSequence has Unicode surrogate pair at the specified [index].\n */\n^public fun CharSequence.hasSurrogatePairAt(index: Int): Boolean {\n    return index in 0..length - 2\n        && this[index].isHighSurrogate()\n        && this[index + 1].isLowSurrogate()\n}\n/**\n * Returns a substring specified by the given [range] of indices.\n */\n^public fun String.substring(range: IntRange): String = substring(range.start, range.endInclusive + 1)\n/**\n * Returns a subsequence of this char sequence specified by the given [range] of indices.\n */\n^public fun CharSequence.subSequence(range: IntRange): CharSequence = subSequence(range.start, range.endInclusive + 1)\n/**\n * Returns a subsequence of this char sequence.\n */\n * This extension is chosen only for invocation with old-named parameters.\n * Replace parameter names with the same as those of [CharSequence.subSequence].\n
*^@kotlin.internal.InlineOnly^@Suppress("EXTENSION_SHADOWED_BY_MEMBER") // false\nwarning^@Deprecated("Use parameters named startIndex and endIndex.", ReplaceWith("subSequence(startIndex = start, endIndex = end)"))^public inline fun String.subSequence(start: Int, end: Int): CharSequence = subSequence(start, end)\n/**\n * Returns a substring of chars from a range of this char sequence starting at the [startIndex] and ending right before the [endIndex].\n */\n * @param startIndex the start index (inclusive).\n * @param endIndex the end index (exclusive). If not specified, the length of the char sequence is used.\n
*^@kotlin.internal.InlineOnly^public inline fun CharSequence.substring(startIndex: Int, endIndex: Int = length): String = subSequence(startIndex, endIndex).toString()\n/**\n * Returns a substring of chars at indices from the specified [range] of this char sequence.\n */\n^public fun CharSequence.substring(range: IntRange): String = subSequence(range.start, range.endInclusive + 1).toString()\n/**\n * Returns a substring before the first occurrence of [delimiter].\n * If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.\n */\n^public fun String.substringBefore(delimiter: Char, missingDelimiterValue: String = this): String {\n    val index = indexOf(delimiter)\n    return if (index == -1) missingDelimiterValue else substring(0, index)\n}\n/**\n * Returns a substring before the first occurrence of [delimiter].\n * If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.\n */\n^public fun String.substringBefore(delimiter: String, missingDelimiterValue: String = this): String {\n    val index = indexOf(delimiter)\n    return if (index == -1) missingDelimiterValue else substring(0, index)\n}\n/**\n * Returns a substring after the first occurrence of [delimiter].\n * If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.\n */\n^public fun String.substringAfter(delimiter: Char, missingDelimiterValue: String = this): String {\n    val index = indexOf(delimiter)\n    return if (index == -1) missingDelimiterValue else substring(index + 1, length)\n}\n/**\n * Returns a substring after the first occurrence of [delimiter].\n * If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.\n */\n^public fun String.substringAfter(delimiter: String, missingDelimiterValue: String = this): String {\n    val index = indexOf(delimiter)\n    return if (index == -1) missingDelimiterValue else substring(index + delimiter.length, length)\n}\n/**\n * Returns a substring before the last occurrence of [delimiter].\n * If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.\n */\n^public fun String.substringBeforeLast(delimiter: Char, missingDelimiterValue: String = this): String {\n    val index = lastIndexOf(delimiter)\n    return if (index == -1) missingDelimiterValue else substring(0, index)\n}\n/**\n * Returns a substring before the last occurrence of [delimiter].\n * If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.\n */\n^public fun String.substringBeforeLast(delimiter: String, missingDelimiterValue: String = this): String {\n    val index = lastIndexOf(delimiter)\n    return if (index ==

```

```

-1) missingDelimiterValue else substring(0, index)\n}\n\n/**\n * Returns a substring after the last occurrence of
[delimiter].\n * If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the
original string.\n */\npublic fun String.substringAfterLast(delimiter: Char, missingDelimiterValue: String = this):
String {\n    val index = lastIndexOf(delimiter)\n    return if (index == -1) missingDelimiterValue else
substring(index + 1, length)\n}\n\n/**\n * Returns a substring after the last occurrence of [delimiter].\n * If the
string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.\n
*/\npublic fun String.substringAfterLast(delimiter: String, missingDelimiterValue: String = this): String {\n    val
index = lastIndexOf(delimiter)\n    return if (index == -1) missingDelimiterValue else substring(index +
delimiter.length, length)\n}\n\n/**\n * Returns a char sequence with content of this char sequence where its part at
the given range\n * is replaced with the [replacement] char sequence.\n * @param startIndex the index of the first
character to be replaced.\n * @param endIndex the index of the first character after the replacement to keep in the
string.\n */\npublic fun CharSequence.replaceRange(startIndex: Int, endIndex: Int, replacement: CharSequence):
CharSequence {\n    if (endIndex < startIndex)\n        throw IndexOutOfBoundsException("\u201cEnd index ($endIndex)
is less than start index ($startIndex).\u201c")\n    val sb = StringBuilder()\n    sb.appendRange(this, 0, startIndex)\n
sb.append(replacement)\n    sb.appendRange(this, endIndex, length)\n    return sb\n}\n\n/**\n * Replaces the part of
the string at the given range with the [replacement] char sequence.\n * @param startIndex the index of the first
character to be replaced.\n * @param endIndex the index of the first character after the replacement to keep in the
string.\n */\n@kotlin.internal.InlineOnly\npublic inline fun String.replaceRange(startIndex: Int, endIndex: Int,
replacement: CharSequence): String =\n    (this as CharSequence).replaceRange(startIndex, endIndex,
replacement).toString()\n}\n\n/**\n * Returns a char sequence with content of this char sequence where its part at the
given [range]\n * is replaced with the [replacement] char sequence.\n * \n * The end index of the [range] is included
in the part to be replaced.\n */\npublic fun CharSequence.replaceRange(range: IntRange, replacement:
CharSequence): CharSequence =\n    replaceRange(range.start, range.endInclusive + 1, replacement)\n}\n\n/**\n * Replace the part of string at the given [range] with the [replacement] string.\n * \n * The end index of the [range] is
included in the part to be replaced.\n */\n@kotlin.internal.InlineOnly\npublic inline fun String.replaceRange(range:
IntRange, replacement: CharSequence): String =\n    (this as CharSequence).replaceRange(range,
replacement).toString()\n}\n\n/**\n * Returns a char sequence with content of this char sequence where its part at the
given range is removed.\n * \n * @param startIndex the index of the first character to be removed.\n * @param
endIndex the index of the first character after the removed part to keep in the string.\n * \n * [endIndex] is not
included in the removed part.\n */\npublic fun CharSequence.removeRange(startIndex: Int, endIndex: Int):
CharSequence {\n    if (endIndex < startIndex)\n        throw IndexOutOfBoundsException("\u201cEnd index ($endIndex)
is less than start index ($startIndex).\u201c")\n    if (endIndex == startIndex)\n        return this.subSequence(0,
length)\n    val sb = StringBuilder(length - (endIndex - startIndex))\n    sb.appendRange(this, 0, startIndex)\n
sb.appendRange(this, endIndex, length)\n    return sb\n}\n\n/**\n * Removes the part of a string at a given range.\n *
\n * @param startIndex the index of the first character to be removed.\n * @param endIndex the index of the first
character after the removed part to keep in the string.\n * \n * [endIndex] is not included in the removed part.\n
*/\n@kotlin.internal.InlineOnly\npublic inline fun String.removeRange(startIndex: Int, endIndex: Int): String =\n    (this as CharSequence).removeRange(startIndex, endIndex).toString()\n}\n\n/**\n * Returns a char sequence with
content of this char sequence where its part at the given [range] is removed.\n * \n * The end index of the [range] is
included in the removed part.\n */\npublic fun CharSequence.removeRange(range: IntRange): CharSequence =
removeRange(range.start, range.endInclusive + 1)\n}\n\n/**\n * Removes the part of a string at the given [range].\n * \n
* The end index of the [range] is included in the removed part.\n */\n@kotlin.internal.InlineOnly\npublic inline fun
String.removeRange(range: IntRange): String =\n    (this as CharSequence).removeRange(range).toString()\n}\n\n/**\n * If this char sequence starts with the given [prefix], returns a new char sequence\n * with the prefix removed.
Otherwise, returns a new char sequence with the same characters.\n */\npublic fun
CharSequence.removePrefix(prefix: CharSequence): CharSequence {\n    if (startsWith(prefix)) {\n        return
subSequence(prefix.length, length)\n    }\n    return subSequence(0, length)\n}\n\n/**\n * If this string starts with the
given [prefix], returns a copy of this string\n * with the prefix removed. Otherwise, returns this string.\n */\npublic

```

```

fun String.removePrefix(prefix: CharSequence): String {
    if (startsWith(prefix)) {
        return substring(prefix.length)
    }
    return this
}

/** If this char sequence ends with the given [suffix], returns a new char sequence with the suffix removed. Otherwise, returns a new char sequence with the same characters. */
public fun CharSequence.removeSuffix(suffix: CharSequence): CharSequence {
    if (endsWith(suffix)) {
        return subSequence(0, length - suffix.length)
    }
    return subSequence(0, length)
}

/** If this string ends with the given [suffix], returns a copy of this string with the suffix removed. Otherwise, returns this string. */
public fun String.removeSuffix(suffix: CharSequence): String {
    if (endsWith(suffix)) {
        return substring(0, length - suffix.length)
    }
    return this
}

/** When this char sequence starts with the given [prefix] and ends with the given [suffix], returns a new char sequence having both the given [prefix] and [suffix] removed. Otherwise returns a new char sequence with the same characters. */
public fun CharSequence.removeSurrounding(prefix: CharSequence, suffix: CharSequence): CharSequence {
    if ((length >= prefix.length + suffix.length) && startsWith(prefix) && endsWith(suffix)) {
        return subSequence(prefix.length, length - suffix.length)
    }
    return subSequence(0, length)
}

/** Removes from a string both the given [prefix] and [suffix] if and only if it starts with the [prefix] and ends with the [suffix]. Otherwise returns this string unchanged. */
public fun String.removeSurrounding(prefix: CharSequence, suffix: CharSequence): String {
    if ((length >= prefix.length + suffix.length) && startsWith(prefix) && endsWith(suffix)) {
        return substring(prefix.length, length - suffix.length)
    }
    return this
}

/** When this char sequence starts with and ends with the given [delimiter], returns a new char sequence having this [delimiter] removed both from the start and end. Otherwise returns a new char sequence with the same characters. */
public fun CharSequence.removeSurrounding(delimiter: CharSequence): CharSequence {
    return removeSurrounding(delimiter, delimiter)
}

/** Removes the given [delimiter] string from both the start and the end of this string if and only if it starts with and ends with the [delimiter]. Otherwise returns this string unchanged. */
public fun String.removeSurrounding(delimiter: CharSequence): String {
    return removeSurrounding(delimiter, delimiter)
}

/** Replace part of string before the first occurrence of given delimiter with the [replacement] string. If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string. */
public fun String.replaceBefore(delimiter: Char, replacement: String, missingDelimiterValue: String = this): String {
    val index = indexOf(delimiter)
    return if (index == -1) missingDelimiterValue else replaceRange(0, index, replacement)
}

/** Replace part of string before the first occurrence of given delimiter with the [replacement] string. If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string. */
public fun String.replaceBefore(delimiter: String, replacement: String, missingDelimiterValue: String = this): String {
    val index = indexOf(delimiter)
    return if (index == -1) missingDelimiterValue else replaceRange(0, index, replacement)
}

/** Replace part of string after the first occurrence of given delimiter with the [replacement] string. If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string. */
public fun String.replaceAfter(delimiter: Char, replacement: String, missingDelimiterValue: String = this): String {
    val index = indexOf(delimiter)
    return if (index == -1) missingDelimiterValue else replaceRange(index + 1, length, replacement)
}

/** Replace part of string after the first occurrence of given delimiter with the [replacement] string. If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string. */
public fun String.replaceAfter(delimiter: String, replacement: String, missingDelimiterValue: String = this): String {
    val index = indexOf(delimiter)
    return if (index == -1) missingDelimiterValue else replaceRange(index + delimiter.length, length, replacement)
}

/** Replace part of string after the last occurrence of given delimiter with the [replacement] string. If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string. */
public fun String.replaceAfterLast(delimiter: Char, replacement: String, missingDelimiterValue: String = this): String {
    val index = lastIndexOf(delimiter)
    return if (index == -1) missingDelimiterValue else replaceRange(index + 1, length, replacement)
}

/** Replace part of string after the last occurrence of given delimiter with the [replacement] string. If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string. */
public fun String.replaceAfterLast(delimiter: String, replacement: String, missingDelimiterValue: String = this): String {
    val index = lastIndexOf(delimiter)
    return if (index == -1) missingDelimiterValue else replaceRange(index + delimiter.length, length, replacement)
}

```

```

Char, replacement: String, missingDelimiterValue: String = this): String {
    val index = lastIndexOf(delimiter)
    return if (index == -1) missingDelimiterValue else replaceRange(index + 1, length, replacement)
}

Replace part of string before the last occurrence of given delimiter with the [replacement] string.
If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.

public fun String.replaceBeforeLast(delimiter: Char, replacement: String, missingDelimiterValue: String = this): String {
    val index = lastIndexOf(delimiter)
    return if (index == -1) missingDelimiterValue else replaceRange(0, index, replacement)
}

Replace part of string before the last occurrence of given delimiter with the [replacement] string.
If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.

public fun String.replaceBeforeLast(delimiter: String, replacement: String, missingDelimiterValue: String = this): String {
    val index = lastIndexOf(delimiter)
    return if (index == -1) missingDelimiterValue else replaceRange(0, index, replacement)
}

// public fun String.replace(oldChar: Char, newChar: Char, ignoreCase: Boolean): String // JVM- and JS-specific
// public fun String.replace(oldValue: String, newValue: String, ignoreCase: Boolean): String // JVM- and JS-specific

Returns a new string obtained by replacing each substring of this char sequence that matches the given regular expression with the given [replacement].

The [replacement] can consist of any combination of literal text and $-substitutions. To treat the replacement string literally escape it with the [kotlin.text.Regex.Companion.escapeReplacement] method.

@kotlin.internal.InlineOnly
public inline fun CharSequence.replace(regex: Regex, replacement: String): String = regex.replace(this, replacement)

Returns a new string obtained by replacing each substring of this char sequence that matches the given regular expression with the result of the given function [transform] that takes [MatchResult] and returns a string to be used as a replacement for that match.

@kotlin.internal.InlineOnly
public inline fun CharSequence.replace(regex: Regex, noinline transform: (MatchResult) -> CharSequence): String = regex.replace(this, transform)

Replaces the first occurrence of the given regular expression [regex] in this char sequence with specified [replacement] expression.

@param replacement A replacement expression that can include substitutions. See [Regex.replaceFirst] for details.

@kotlin.internal.InlineOnly
public inline fun CharSequence.replaceFirst(regex: Regex, replacement: String): String = regex.replaceFirst(this, replacement)

Returns a copy of this string having its first character replaced with the result of the specified [transform], or the original string if it's empty.

@param transform function that takes the first character and returns the result of the transform applied to the character.

@sample samples.text.Strings.replaceFirstChar

@SinceKotlin("1.5")
@WasExperimental(ExperimentalStdlibApi::class)
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@JvmName("replaceFirstCharWithChar")
@kotlin.internal.InlineOnly
public inline fun String.replaceFirstChar(transform: (Char) -> Char): String {
    return if (isNotEmpty()) transform(this[0]) + substring(1) else this
}

Returns a copy of this string having its first character replaced with the result of the specified [transform], or the original string if it's empty.

@param transform function that takes the first character and returns the result of the transform applied to the character.

@sample samples.text.Strings.replaceFirstChar

@SinceKotlin("1.5")
@WasExperimental(ExperimentalStdlibApi::class)
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolutionByLambdaReturnType
@JvmName("replaceFirstCharWithCharSequence")
@kotlin.internal.InlineOnly
public inline fun String.replaceFirstChar(transform: (Char) -> CharSequence): String {
    return if (isNotEmpty()) transform(this[0]).toString() + substring(1) else this
}

Returns `true` if this char sequence matches the given regular expression.

@kotlin.internal.InlineOnly
public inline infix fun CharSequence.matches(regex: Regex): Boolean = regex.matches(this)

Implementation of [regionMatches] for CharSequences.

Invoked when it's already known that arguments are not Strings, so that no additional type checks are performed.

internal fun CharSequence.regionMatchesImpl(thisOffset: Int, other: CharSequence, otherOffset: Int, length: Int, ignoreCase: Boolean): Boolean {
    if ((otherOffset < 0) || (thisOffset < 0) || (thisOffset > this.length - length) || (otherOffset > other.length - length)) {
        return false
    }
    for (index in 0 until length) {
        if (!this[thisOffset + index].equals(other[otherOffset + index], ignoreCase)) {
            return false
        }
    }
    return true
}

```

```

Returns `true` if this char sequence starts with the specified character.\n */\npublic fun
CharSequence.startsWith(char: Char, ignoreCase: Boolean = false): Boolean =\n    this.length > 0 &&
this[0].equals(char, ignoreCase)\n\n/**\n * Returns `true` if this char sequence ends with the specified character.\n
*/\npublic fun CharSequence.endsWith(char: Char, ignoreCase: Boolean = false): Boolean =\n    this.length > 0 &&
this[lastIndex].equals(char, ignoreCase)\n\n/**\n * Returns `true` if this char sequence starts with the specified
prefix.\n */\npublic fun CharSequence.startsWith(prefix: CharSequence, ignoreCase: Boolean = false): Boolean {\n
    if (!ignoreCase && this is String && prefix is String)\n        return this.startsWith(prefix)\n    else\n        return
regionMatchesImpl(0, prefix, 0, prefix.length, ignoreCase)\n}\n\n/**\n * Returns `true` if a substring of this char
sequence starting at the specified offset [startIndex] starts with the specified prefix.\n */\npublic fun
CharSequence.startsWith(prefix: CharSequence, startIndex: Int, ignoreCase: Boolean = false): Boolean {\n    if
(!ignoreCase && this is String && prefix is String)\n        return this.startsWith(prefix, startIndex)\n    else\n
return regionMatchesImpl(startIndex, prefix, 0, prefix.length, ignoreCase)\n}\n\n/**\n * Returns `true` if this char
sequence ends with the specified suffix.\n */\npublic fun CharSequence.endsWith(suffix: CharSequence,
ignoreCase: Boolean = false): Boolean {\n    if (!ignoreCase && this is String && suffix is String)\n        return
this.endsWith(suffix)\n    else\n        return regionMatchesImpl(length - suffix.length, suffix, 0, suffix.length,
ignoreCase)\n}\n\n// common prefix and suffix\n\n/**\n * Returns the longest string `prefix` such that this char
sequence and [other] char sequence both start with this prefix,\n * taking care not to split surrogate pairs.\n * If this
and [other] have no common prefix, returns the empty string.\n */\n * @param ignoreCase `true` to ignore character
case when matching a character. By default `false`.\n * @sample samples.text.Strings.commonPrefixWith\n
*/\npublic fun CharSequence.commonPrefixWith(other: CharSequence, ignoreCase: Boolean = false): String {\n
    val shortestLength = minOf(this.length, other.length)\n    var i = 0\n    while (i < shortestLength &&
this[i].equals(other[i], ignoreCase = ignoreCase)) {\n        i++\n    }\n    if (this.hasSurrogatePairAt(i - 1) ||
other.hasSurrogatePairAt(i - 1)) {\n        i--\n    }\n    return subSequence(0, i).toString()\n}\n\n/**\n * Returns the
longest string `suffix` such that this char sequence and [other] char sequence both end with this suffix,\n * taking
care not to split surrogate pairs.\n * If this and [other] have no common suffix, returns the empty string.\n */\n *
@param ignoreCase `true` to ignore character case when matching a character. By default `false`.\n * @sample
samples.text.Strings.commonSuffixWith\n
*/\npublic fun CharSequence.commonSuffixWith(other: CharSequence,
ignoreCase: Boolean = false): String {\n    val thisLength = this.length\n    val otherLength = other.length\n    val
shortestLength = minOf(thisLength, otherLength)\n    var i = 0\n    while (i < shortestLength && this[thisLength -
i - 1].equals(other[otherLength - i - 1], ignoreCase = ignoreCase)) {\n        i++\n    }\n    if
(this.hasSurrogatePairAt(thisLength - i - 1) || other.hasSurrogatePairAt(otherLength - i - 1)) {\n        i--\n    }\n
return subSequence(thisLength - i, thisLength).toString()\n}\n\n// indexOfAny()\n\n/**\n * Finds the index of the
first occurrence of any of the specified [chars] in this char sequence,\n * starting from the specified [startIndex] and
optionally ignoring the case.\n */\n * @param ignoreCase `true` to ignore character case when matching a character.
By default `false`.\n * @return An index of the first occurrence of matched character from [chars] or -1 if none of
[chars] are found.\n */\n * \n\npublic fun CharSequence.indexOfAny(chars: CharArray, startIndex: Int = 0, ignoreCase:
Boolean = false): Int {\n    if (!ignoreCase && chars.size == 1 && this is String) {\n        val char = chars.single()\n
return nativeIndexOf(char, startIndex)\n    }\n    for (index in startIndex.coerceAtLeast(0)..lastIndex) {\n        val
charAtIndex = get(index)\n        if (chars.any { it.equals(charAtIndex, ignoreCase) })\n            return index\n    }\n
return -1\n}\n\n/**\n * Finds the index of the last occurrence of any of the specified [chars] in this char sequence,\n
* starting from the specified [startIndex] and optionally ignoring the case.\n */\n * @param startIndex The index of
character to start searching at. The search proceeds backward toward the beginning of the string.\n * @param
ignoreCase `true` to ignore character case when matching a character. By default `false`.\n * @return An index of
the last occurrence of matched character from [chars] or -1 if none of [chars] are found.\n */\n * \n\npublic fun
CharSequence.lastIndexOfAny(chars: CharArray, startIndex: Int = lastIndex, ignoreCase: Boolean = false): Int {\n
    if (!ignoreCase && chars.size == 1 && this is String) {\n        val char = chars.single()\n        return
nativeLastIndexOf(char, startIndex)\n    }\n    for (index in startIndex.coerceAtMost(lastIndex) downTo 0) {\n
        val charAtIndex = get(index)\n        if (chars.any { it.equals(charAtIndex, ignoreCase) })\n            return index\n
}

```

```

}\n\n return -1\n}\n\nprivate fun CharSequence.indexOf(other: CharSequence, startIndex: Int, endIndex: Int,
ignoreCase: Boolean, last: Boolean = false): Int {\n    val indices = if (!last)\n        startIndex.coerceAtLeast(0)..endIndex.coerceAtMost(length)\n    else\n        startIndex.coerceAtMost(lastIndex)\n        downTo endIndex.coerceAtLeast(0)\n\n    if (this is String && other is String) { // smart cast\n        for (index in\n            indices) {\n                if (other.regionMatches(0, this, index, other.length, ignoreCase))\n                    return index\n            }\n    } else {\n        for (index in indices) {\n            if (other.regionMatchesImpl(0, this, index, other.length,\n                ignoreCase))\n                return index\n        }\n    }\n    return -1\n}\n\nprivate fun CharSequence.findAnyOf(strings:
Collection<String>, startIndex: Int, ignoreCase: Boolean, last: Boolean): Pair<Int, String>? {\n    if (!ignoreCase
&& strings.size == 1) {\n        val string = strings.single()\n        val index = if (!last) indexOf(string, startIndex) else
lastIndexOf(string, startIndex)\n        return if (index < 0) null else index to string\n    }\n\n    val indices = if (!last)
startIndex.coerceAtLeast(0)..length else startIndex.coerceAtMost(lastIndex) downTo 0\n\n    if (this is String) {\n        for (index in indices) {\n            val matchingString = strings.firstOrNull { it.regionMatches(0, this, index, it.length, ignoreCase) }\n            if (matchingString != null)\n                return index to matchingString\n        }\n    } else {\n        for (index in indices) {\n            val matchingString = strings.firstOrNull { it.regionMatchesImpl(0, this, index, it.length, ignoreCase) }\n            if (matchingString != null)\n                return index to matchingString\n        }\n    }\n\n    return null\n}\n\n/**\n * Finds the first occurrence of any of the specified [strings] in this char sequence,\n * starting from the specified [startIndex] and optionally ignoring the case.\n * @param ignoreCase `true` to ignore character case when matching a string. By default `false`.\n * @return A pair of an index of the first occurrence of matched string from [strings] and the string matched\n * or `null` if none of [strings] are found.\n * To avoid ambiguous results when strings in [strings] have characters in common, this method proceeds from\n * the beginning to the end of this string, and finds at each position the first element in [strings]\n * that matches this string at that position.\n */\npublic fun CharSequence.findAnyOf(strings: Collection<String>, startIndex: Int = 0, ignoreCase: Boolean = false): Pair<Int, String>? =\n    findAnyOf(strings, startIndex, ignoreCase, last = false)\n\n/**\n * Finds the last occurrence of any of the specified [strings] in this char sequence,\n * starting from the specified [startIndex] and optionally ignoring the case.\n * @param startIndex The index of character to start searching at. The search proceeds backward toward the beginning of the string.\n * @param ignoreCase `true` to ignore character case when matching a string. By default `false`.\n * @return A pair of an index of the last occurrence of matched string from [strings] and the string matched or `null` if none of [strings] are found.\n * To avoid ambiguous results when strings in [strings] have characters in common, this method proceeds from\n * the end toward the beginning of this string, and finds at each position the first element in [strings]\n * that matches this string at that position.\n */\npublic fun CharSequence.findLastAnyOf(strings: Collection<String>, startIndex: Int = lastIndex, ignoreCase: Boolean = false): Pair<Int, String>? =\n    findAnyOf(strings, startIndex, ignoreCase, last = true)\n\n/**\n * Finds the index of the first occurrence of any of the specified [strings] in this char sequence,\n * starting from the specified [startIndex] and optionally ignoring the case.\n * @param ignoreCase `true` to ignore character case when matching a string. By default `false`.\n * @return An index of the first occurrence of matched string from [strings] or -1 if none of [strings] are found.\n * To avoid ambiguous results when strings in [strings] have characters in common, this method proceeds from\n * the beginning to the end of this string, and finds at each position the first element in [strings]\n * that matches this string at that position.\n */\npublic fun CharSequence.indexOfAny(strings: Collection<String>, startIndex: Int = 0, ignoreCase: Boolean = false): Int =\n    findAnyOf(strings, startIndex, ignoreCase, last = false)?.first ?: -1\n\n/**\n * Finds the index of the last occurrence of any of the specified [strings] in this char sequence,\n * starting from the specified [startIndex] and optionally ignoring the case.\n * @param startIndex The index of character to start searching at. The search proceeds backward toward the beginning of the string.\n * @param ignoreCase `true` to ignore character case when matching a string. By default `false`.\n * @return An index of the last occurrence of matched string from [strings] or -1 if none of [strings] are found.\n * To avoid ambiguous results when strings in [strings] have characters in common, this method proceeds from\n * the end toward the beginning of this string, and finds at each position the first element in [strings]\n * that matches this string at that position.\n */\npublic fun CharSequence.lastIndexOfAny(strings: Collection<String>, startIndex: Int = lastIndex, ignoreCase: Boolean =

```

```

false): Int = \n    findAnyOf(strings, startIndex, ignoreCase, last = true)?.first ?: -1\n\n\n// indexOf\n\n/**\n * Returns the index within this string of the first occurrence of the specified character, starting from the specified [startIndex].\n * @param ignoreCase `true` to ignore character case when matching a character. By default `false`.\n * @return An index of the first occurrence of [char] or -1 if none is found.\n */\npublic fun CharSequence.indexOf(char: Char, startIndex: Int = 0, ignoreCase: Boolean = false): Int {\n    return if (ignoreCase || this !is String)\n        indexOfAny(charArrayOf(char), startIndex, ignoreCase)\n    else\n        nativeIndexOf(char, startIndex)\n}\n\n/**\n * Returns the index within this char sequence of the first occurrence of the specified [string],\n * starting from the specified [startIndex].\n * @param ignoreCase `true` to ignore character case when matching a string. By default `false`.\n * @return An index of the first occurrence of [string] or -1 if none is found.\n * @sample samples.text.Strings.indexOf\n */\npublic fun CharSequence.indexOf(string: String, startIndex: Int = 0, ignoreCase: Boolean = false): Int {\n    return if (ignoreCase || this !is String)\n        indexOf(string, startIndex, length, ignoreCase)\n    else\n        nativeIndexOf(string, startIndex)\n}\n\n/**\n * Returns the index within this char sequence of the last occurrence of the specified character,\n * starting from the specified [startIndex].\n * @param startIndex The index of character to start searching at. The search proceeds backward toward the beginning of the string.\n * @param ignoreCase `true` to ignore character case when matching a character. By default `false`.\n * @return An index of the last occurrence of [char] or -1 if none is found.\n */\npublic fun CharSequence.lastIndexOf(char: Char, startIndex: Int = lastIndex, ignoreCase: Boolean = false): Int {\n    return if (ignoreCase || this !is String)\n        lastIndexOfAny(charArrayOf(char), startIndex, ignoreCase)\n    else\n        nativeLastIndexOf(char, startIndex)\n}\n\n/**\n * Returns the index within this char sequence of the last occurrence of the specified [string],\n * starting from the specified [startIndex].\n * @param startIndex The index of character to start searching at. The search proceeds backward toward the beginning of the string.\n * @param ignoreCase `true` to ignore character case when matching a string. By default `false`.\n * @return An index of the last occurrence of [string] or -1 if none is found.\n */\npublic fun CharSequence.lastIndexOf(string: String, startIndex: Int = lastIndex, ignoreCase: Boolean = false): Int {\n    return if (ignoreCase || this !is String)\n        indexOf(string, startIndex, 0, ignoreCase, last = true)\n    else\n        nativeLastIndexOf(string, startIndex)\n}\n\n/**\n * Returns `true` if this char sequence contains the specified [other] sequence of characters as a substring.\n * @param ignoreCase `true` to ignore character case when comparing strings. By default `false`.\n */\n@Suppress("INAPPLICABLE_OPERATOR_MODIFIER")\npublic operator fun CharSequence.contains(other: CharSequence, ignoreCase: Boolean = false): Boolean =\n    if (other is String)\n        indexOf(other, ignoreCase = ignoreCase) >= 0\n    else\n        indexOf(other, 0, length, ignoreCase) >= 0\n\n/**\n * Returns `true` if this char sequence contains the specified character [char].\n * @param ignoreCase `true` to ignore character case when comparing characters. By default `false`.\n */\n@Suppress("INAPPLICABLE_OPERATOR_MODIFIER")\npublic operator fun CharSequence.contains(char: Char, ignoreCase: Boolean = false): Boolean =\n    indexOf(char, ignoreCase = ignoreCase) >= 0\n\n/**\n * Returns `true` if this char sequence contains at least one match of the specified regular expression [regex].\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun CharSequence.contains(regex: Regex): Boolean =\n    regex.containsMatchIn(this)\n\n// rangesDelimitedBy\n\nprivate class DelimitedRangesSequence(\n    private val input: CharSequence,\n    private val startIndex: Int,\n    private val limit: Int,\n    private val getNextMatch: CharSequence.(currentIndex: Int) -> Pair<Int, Int>?) : Sequence<IntRange> {\n    override fun iterator(): Iterator<IntRange> = object : Iterator<IntRange> {\n        var nextState: Int = -1 // -1 for unknown, 0 for done, 1 for continue\n        var currentStartIndex: Int = startIndex.coerceIn(0, input.length)\n        var nextSearchIndex: Int = currentStartIndex\n        var nextItem: IntRange? = null\n        var counter: Int = 0\n        private fun calcNext() {\n            if (nextSearchIndex < 0) {\n                nextState = 0\n                nextItem = null\n            } else {\n                if (limit > 0 && ++counter >= limit || nextSearchIndex > input.length) {\n                    nextItem =\n                        currentStartIndex..input.lastIndex\n                    nextSearchIndex = -1\n                } else {\n                    val match =\n                        input.getNextMatch(nextSearchIndex)\n                    if (match == null) {\n                        nextItem =\n                            currentStartIndex..input.lastIndex\n                    } else {\n                        val\n                            (index, length) = match\n                            nextItem = currentStartIndex until index\n                            currentStartIndex

```



```

when matching a delimiter. By default `false`.
 * @param limit The maximum number of substrings to return.
 * public fun CharSequence.splitToSequence(vararg delimiters: Char, ignoreCase: Boolean = false, limit: Int = 0):
Sequence<String> = rangesDelimitedBy(delimiters, ignoreCase = ignoreCase, limit = limit).map { substring(it)
}
 * Splits this char sequence to a list of strings around occurrences of the specified [delimiters].
 * @param delimiters One or more characters to be used as delimiters.
 * @param ignoreCase `true` to ignore character case when matching a delimiter. By default `false`.
 * @param limit The maximum number of substrings to return.
 * public fun CharSequence.split(vararg delimiters: Char, ignoreCase: Boolean = false, limit: Int = 0):
List<String> {
    if (delimiters.size == 1) {
        return split(delimiters[0].toString(), ignoreCase, limit)
    }
    return rangesDelimitedBy(delimiters, ignoreCase = ignoreCase, limit = limit).asIterable().map { substring(it)
}
}
 * Splits this char sequence to a list of strings around occurrences of the specified [delimiter].
 * This is specialized version of split which receives single non-empty delimiter and offers better performance
 * @param delimiter String used as delimiter
 * @param ignoreCase `true` to ignore character case when matching a delimiter. By default `false`.
 * @param limit The maximum number of substrings to return.
 * private fun CharSequence.split(delimiter: String, ignoreCase: Boolean, limit: Int): List<String> {
    requireNonNegativeLimit(limit)
    var currentOffset = 0
    var nextIndex = indexOf(delimiter, currentOffset, ignoreCase)
    if (nextIndex == -1 || limit == 1) {
        return listOf(this.toString())
    }
    val isLimited = limit > 0
    val result = ArrayList<String>(if (isLimited) limit.coerceAtMost(10) else 10)
    do {
        result.add(substring(currentOffset, nextIndex))
        currentOffset = nextIndex + delimiter.length
        // Do not search for next occurrence if we're reaching limit
        if (isLimited && result.size == limit - 1) break
        nextIndex = indexOf(delimiter, currentOffset, ignoreCase)
    } while (nextIndex != -1)
    result.add(substring(currentOffset, length))
    return result
}
 * Splits this char sequence to a list of strings around matches of the given regular expression.
 * @param limit Non-negative value specifying the maximum number of substrings to return.
 * Zero by default means no limit is set.
 * @kotlin.internal.InlineOnly
 * public inline fun CharSequence.split(regex: Regex, limit: Int = 0): List<String> =
regex.split(this, limit)
 * Splits this char sequence to a sequence of strings around matches of the given regular expression.
 * @param limit Non-negative value specifying the maximum number of substrings to return.
 * Zero by default means no limit is set.
 * @sample samples.text.Strings.splitToSequence
 * @SinceKotlin("1.6")
 * @WasExperimental(ExperimentalStdlibApi::class)
 * @kotlin.internal.InlineOnly
 * public inline fun CharSequence.splitToSequence(regex: Regex, limit: Int = 0): Sequence<String> =
regex.splitToSequence(this, limit)
 * Splits this char sequence to a sequence of lines delimited by any of the following character sequences: CRLF, LF or CR.
 * The lines returned do not include terminating line separators.
 * public fun CharSequence.lineSequence(): Sequence<String> = splitToSequence("\\r\\n", "\\n", "\\r")
 * Splits this char sequence to a list of lines delimited by any of the following character sequences: CRLF, LF or CR.
 * The lines returned do not include terminating line separators.
 * public fun CharSequence.lines(): List<String> = lineSequence().toList()
 * Returns `true` if the contents of this char sequence are equal to the contents of the specified [other],
 * i.e. both char sequences contain the same number of the same characters in the same order.
 * @sample samples.text.Strings.contentEquals
 * @SinceKotlin("1.5")
 * public expect infix fun CharSequence?.contentEquals(other: CharSequence?): Boolean
 * Returns `true` if the contents of this char sequence are equal to the contents of the specified [other],
 * optionally ignoring case difference.
 * @param ignoreCase `true` to ignore character case when comparing contents.
 * @sample samples.text.Strings.contentEquals
 * @SinceKotlin("1.5")
 * public expect fun CharSequence?.contentEquals(other: CharSequence?, ignoreCase: Boolean): Boolean
 * internal fun CharSequence?.contentEqualsIgnoreCaseImpl(other: CharSequence?): Boolean {
    if (this is String && other is String) {
        return this.equals(other, ignoreCase = true)
    }
    if (this === other) return true
    if (this == null || other == null || this.length != other.length) return false
    for (i in 0 until length) {
        if (!this[i].equals(other[i], ignoreCase = true)) {
            return false
        }
    }
    return true
}
 * internal fun CharSequence?.contentEqualsImpl(other: CharSequence?): Boolean {
    if (this is String && other is String) {
        return this == other
    }
    if (this === other) return true
    if (this == null || other == null || this.length !=

```

```

other.length) return false\n\n    for (i in 0 until length) {\n        if (this[i] != other[i]) {\n            return false\n        }\n    }\n    return true\n}\n\n/**\n * Returns `true` if the content of this string is equal to the word `true`, `false` if it is  

equal to `false`,\n * and throws an exception otherwise.\n *\n * There is also a lenient version of the function  

available on nullable String, [String?.toBoolean].\n * Note that this function is case-sensitive.\n *\n * @sample  

samples.text.Strings.toBooleanStrict\n *\n@SinceKotlin("1.5")\npublic fun String.toBooleanStrict(): Boolean =  

when (this) {\n    `true` -> true\n    `false` -> false\n    else -> throw IllegalArgumentException("The string  

doesn't represent a boolean value: $this")\n}\n\n/**\n * Returns `true` if the content of this string is equal to the  

word `true`, `false` if it is equal to `false`,\n * and `null` otherwise.\n *\n * There is also a lenient version of the  

function available on nullable String, [String?.toBoolean].\n * Note that this function is case-sensitive.\n *\n *  

@sample samples.text.Strings.toBooleanStrictOrNull\n *\n@SinceKotlin("1.5")\npublic fun  

String.toBooleanStrictOrNull(): Boolean? = when (this) {\n    `true` -> true\n    `false` -> false\n    else ->  

null\n},"/>\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of  

this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n *\n@n\n//  

Auto-generated file. DO NOT EDIT!\n\npackage kotlin\n\nimport  

kotlin.jvm.*\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@JvmInline\npublic value class  

UByteArray\n@PublishedApi\ninternal constructor(@PublishedApi internal val storage: ByteArray) :  

Collection<UByte> {\n\n    /** Creates a new array of the specified [size], with all elements initialized to zero. *\n\n    public constructor(size: Int) : this(ByteArray(size))\n\n    /**\n     * Returns the array element at the given [index].  

This method can be called using the index operator.\n     *\n     * If the [index] is out of bounds of this array, throws  

an [IndexOutOfBoundsException] except in Kotlin/JS\n     * where the behavior is unspecified.\n     *\n     public  

operator fun get(index: Int): UByte = storage[index].toUByte()\n\n    /**\n     * Sets the element at the given [index]  

to the given [value]. This method can be called using the index operator.\n     *\n     * If the [index] is out of bounds  

of this array, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n     * where the behavior is  

unspecified.\n     *\n     public operator fun set(index: Int, value: UByte) {\n        storage[index] = value.toByte()\n    }\n\n    /** Returns the number of elements in the array. *\n     public override val size: Int get() = storage.size\n\n    /** Creates an iterator over the elements of the array. *\n     public override operator fun iterator():  

kotlin.collections.Iterator<UByte> = Iterator(storage)\n\n    @Suppress("DEPRECATION_ERROR")\n    private  

class Iterator(private val array: ByteArray) : UByteIterator() {\n        private var index = 0\n        override fun  

hasNext() = index < array.size\n        override fun nextUByte() = if (index < array.size) array[index++].toUByte()  

else throw NoSuchElementException(index.toString())\n    }\n\n    override fun contains(element: UByte): Boolean  

{\n        // TODO: Eliminate this check after KT-30016 gets fixed.\n        // Currently JS BE does not generate  

special bridge method for this method.\n        @Suppress("USELESS_CAST")\n        if ((element as Any?) !is  

UByte) return false\n\n        return storage.contains(element.toByte())\n    }\n\n    override fun containsAll(elements:  

Collection<UByte>): Boolean {\n        return (elements as Collection<*>).all { it is UByte &&  

storage.contains(it.toByte()) }\n    }\n\n    override fun isEmpty(): Boolean = this.storage.size == 0\n}\n\n/**\n *  

Creates a new array of the specified [size], where each element is calculated by calling the specified\n * [init]  

function.\n *\n * The function [init] is called for each array element sequentially starting from the first one.\n * It  

should return the value for an array element given its index.\n\n    *\n@n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun  

UByteArray(size: Int, init: (Int) -> UByte): UByteArray {\n    return UByteArray(ByteArray(size) { index ->  

init(index).toByte() })\n}\n\n@n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun  

ubyteArrayOf(vararg elements: UByte): UByteArray = elements\n","/>\n * Copyright 2010-2022 JetBrains s.r.o. and  

Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that  

can be found in the license/LICENSE.txt file.\n *\n@n\n// Auto-generated file. DO NOT EDIT!\n\npackage  

kotlin\n\nimport kotlin.jvm.*\n\n@n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@JvmInline\npublic  

value class UIntArray\n@PublishedApi\ninternal constructor(@PublishedApi internal val storage: IntArray) :  

Collection<UInt> {\n\n    /** Creates a new array of the specified [size], with all elements initialized to zero. *\n\n

```

```

public constructor(size: Int) : this(IntArray(size))\n\n /**\n * Returns the array element at the given [index].
This method can be called using the index operator.\n * \n * If the [index] is out of bounds of this array, throws
an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n * \n public
operator fun get(index: Int): UInt = storage[index].toInt()\n\n /**\n * Sets the element at the given [index] to
the given [value]. This method can be called using the index operator.\n * \n * If the [index] is out of bounds of
this array, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*\n public operator fun set(index: Int, value: UInt) {\n    storage[index] = value.toInt()\n } \n\n /** Returns
the number of elements in the array. *\n public override val size: Int get() = storage.size\n\n /** Creates an
iterator over the elements of the array. *\n public override operator fun iterator(): kotlin.collections.Iterator<UInt>
= Iterator(storage)\n\n @Suppress("DEPRECATION_ERROR")\n private class Iterator(private val array:
IntArray) : UIntIterator() {\n    private var index = 0\n    override fun hasNext() = index < array.size\n
override fun nextUInt() = if (index < array.size) array[index++].toInt() else throw
NoSuchElementException(index.toString())\n } \n\n override fun contains(element: UInt): Boolean {\n    //
TODO: Eliminate this check after KT-30016 gets fixed.\n    // Currently JS BE does not generate special bridge
method for this method.\n    @Suppress("USELESS_CAST")\n    if ((element as Any?) !is UInt) return
false\n\n    return storage.contains(element.toInt())\n } \n\n override fun containsAll(elements:
Collection<UInt>): Boolean {\n    return (elements as Collection<*>).all { it is UInt &&
storage.contains(it.toInt()) }\n } \n\n override fun isEmpty(): Boolean = this.storage.size == 0\n}\n\n/**\n *
Creates a new array of the specified [size], where each element is calculated by calling the specified\n * [init]
function.\n * \n * The function [init] is called for each array element sequentially starting from the first one.\n * It
should return the value for an array element given its index.\n
*\n*\n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline fun
UIntArray(size: Int, init: (Int) -> UInt): UIntArray {\n    return UIntArray(IntArray(size) { index ->
init(index).toInt()
})\n}\n\n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline fun
uintArrayOf(vararg elements: UInt): UIntArray = elements\n", "/*\n * Copyright 2010-2022 JetBrains s.r.o. and
Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file.\n */\n\n// Auto-generated file. DO NOT EDIT!\n\npackage
kotlin\n\nimport kotlin.jvm.*\n\n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n @JvmInline\n public
value class ULongArray\n @PublishedApi\n internal constructor(@PublishedApi internal val storage: LongArray) :
Collection<ULong> {\n\n    /** Creates a new array of the specified [size], with all elements initialized to zero. *\n
public constructor(size: Int) : this(LongArray(size))\n\n    /**\n * Returns the array element at the given [index].
This method can be called using the index operator.\n * \n * If the [index] is out of bounds of this array, throws
an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n * \n public
operator fun get(index: Int): ULong = storage[index].toULong()\n\n    /**\n * Sets the element at the given
[index] to the given [value]. This method can be called using the index operator.\n * \n * If the [index] is out of
bounds of this array, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is
unspecified.\n * \n public operator fun set(index: Int, value: ULong) {\n    storage[index] = value.toLong()\n
}\n\n    /** Returns the number of elements in the array. *\n public override val size: Int get() = storage.size\n\n
/** Creates an iterator over the elements of the array. *\n public override operator fun iterator():
kotlin.collections.Iterator<ULong> = Iterator(storage)\n\n    @Suppress("DEPRECATION_ERROR")\n private
class Iterator(private val array: LongArray) : ULongIterator() {\n    private var index = 0\n    override fun
hasNext() = index < array.size\n    override fun nextULong() = if (index < array.size) array[index++].toULong()
else throw NoSuchElementException(index.toString())\n } \n\n    override fun contains(element: ULong): Boolean
{\n    // TODO: Eliminate this check after KT-30016 gets fixed.\n    // Currently JS BE does not generate
special bridge method for this method.\n    @Suppress("USELESS_CAST")\n    if ((element as Any?) !is
ULong) return false\n\n    return storage.contains(element.toLong())\n } \n\n    override fun
containsAll(elements: Collection<ULong>): Boolean {\n    return (elements as Collection<*>).all { it is ULong

```


is unspecified.

```

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline operator fun UIntArray.component1(): UInt {
    return get(0)
}

```

Returns 1st element from the array. If the size of this array is less than 1, throws an [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.

```

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline operator fun ULongArray.component1(): ULong {
    return get(0)
}

```

Returns 1st element from the array. If the size of this array is less than 1, throws an [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.

```

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline operator fun UByteArray.component1(): UByte {
    return get(0)
}

```

Returns 1st element from the array. If the size of this array is less than 1, throws an [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.

```

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline operator fun UShortArray.component1(): UShort {
    return get(0)
}

```

Returns 2nd element from the array. If the size of this array is less than 2, throws an [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.

```

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline operator fun UIntArray.component2(): UInt {
    return get(1)
}

```

Returns 2nd element from the array. If the size of this array is less than 2, throws an [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.

```

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline operator fun ULongArray.component2(): ULong {
    return get(1)
}

```

Returns 2nd element from the array. If the size of this array is less than 2, throws an [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.

```

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline operator fun UByteArray.component2(): UByte {
    return get(1)
}

```

Returns 2nd element from the array. If the size of this array is less than 2, throws an [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.

```

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline operator fun UShortArray.component2(): UShort {
    return get(1)
}

```

Returns 3rd element from the array. If the size of this array is less than 3, throws an [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.

```

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline operator fun UIntArray.component3(): UInt {
    return get(2)
}

```

Returns 3rd element from the array. If the size of this array is less than 3, throws an [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.

```

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline operator fun ULongArray.component3(): ULong {
    return get(2)
}

```

Returns 3rd element from the array. If the size of this array is less than 3, throws an [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.

```

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline operator fun UByteArray.component3(): UByte {
    return get(2)
}

```

Returns 3rd element from the array. If the size of this array is less than 3, throws an [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.

```

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline operator fun UShortArray.component3(): UShort {
    return get(2)
}

```

Returns 4th element from the array. If the size of this array is less than 4, throws an [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun UIntArray.component4(): UInt {\n return get(3)\n}\n\n/**\n * Returns 4th *element* from the array.\n * \n * If the size of this array is less than 4, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun ULongArray.component4(): ULong {\n return get(3)\n}\n\n/**\n * Returns 4th *element* from the array.\n * \n * If the size of this array is less than 4, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun UByteArray.component4(): UByte {\n return get(3)\n}\n\n/**\n * Returns 4th *element* from the array.\n * \n * If the size of this array is less than 4, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun UShortArray.component4(): UShort {\n return get(3)\n}\n\n/**\n * Returns 5th *element* from the array.\n * \n * If the size of this array is less than 5, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun UIntArray.component5(): UInt {\n return get(4)\n}\n\n/**\n * Returns 5th *element* from the array.\n * \n * If the size of this array is less than 5, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun ULongArray.component5(): ULong {\n return get(4)\n}\n\n/**\n * Returns 5th *element* from the array.\n * \n * If the size of this array is less than 5, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun UByteArray.component5(): UByte {\n return get(4)\n}\n\n/**\n * Returns 5th *element* from the array.\n * \n * If the size of this array is less than 5, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun UShortArray.component5(): UShort {\n return get(4)\n}\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n */\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic expect fun UIntArray.elementAt(index: Int): UInt\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n */\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic expect fun ULongArray.elementAt(index: Int): ULong\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n */\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic expect fun UByteArray.elementAt(index: Int): UByte\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrElse\n */\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.elementAtOrElse(index: Int, defaultValue: (Int) -> UInt): UInt {\n return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrElse\n */\n

```

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.elementAtOrElse(index: Int, defaultValue: (Int) -> ULong): ULong {\n    return if (index >= 0 &&
index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the
result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrElse\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.elementAtOrElse(index: Int, defaultValue: (Int) -> UByte): UByte {\n    return if (index >= 0 && index
<= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result
of calling the [defaultValue] function if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrElse\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.elementAtOrElse(index: Int, defaultValue: (Int) -> UShort): UShort {\n    return if (index >= 0 &&
index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or
`null` if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrNull\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.elementAtOrNull(index: Int): UInt? {\n    return this.getOrNull(index)\n}\n\n/**\n * Returns an element
at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrNull\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.elementAtOrNull(index: Int): ULong? {\n    return this.getOrNull(index)\n}\n\n/**\n * Returns an
element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrNull\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.elementAtOrNull(index: Int): UByte? {\n    return this.getOrNull(index)\n}\n\n/**\n * Returns an
element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrNull\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.elementAtOrNull(index: Int): UShort? {\n    return this.getOrNull(index)\n}\n\n/**\n * Returns the
first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.find(predicate: (UInt) -> Boolean): UInt? {\n    return firstOrNull(predicate)\n}\n\n/**\n * Returns the
first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.find(predicate: (ULong) -> Boolean): ULong? {\n    return firstOrNull(predicate)\n}\n\n/**\n *
Returns the first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.find(predicate: (UByte) -> Boolean): UByte? {\n    return firstOrNull(predicate)\n}\n\n/**\n * Returns
the first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.find(predicate: (UShort) -> Boolean): UShort? {\n    return firstOrNull(predicate)\n}\n\n/**\n *
Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample
samples.collections.Collections.Elements.find\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun

```

```

UIntArray.findLast(predicate: (UInt) -> Boolean): UInt? {\n  return lastOrNull(predicate)\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.findLast(predicate: (UByte) -> Boolean): UByte? {\n  return lastOrNull(predicate)\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.findLast(predicate: (UShort) -> Boolean): UShort? {\n  return lastOrNull(predicate)\n}\n\n/**\n * Returns first element.\n * @throws [NoSuchElementException] if the array is empty.\n */\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.first(): UInt {\n  return storage.first().toUInt()\n}\n\n/**\n * Returns first element.\n * @throws [NoSuchElementException] if the array is empty.\n */\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.first(): UByte {\n  return storage.first().toUByte()\n}\n\n/**\n * Returns first element.\n * @throws [NoSuchElementException] if the array is empty.\n */\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.first(): UShort {\n  return storage.first().toUShort()\n}\n\n/**\n * Returns the first element matching the given [predicate].\n * @throws [NoSuchElementException] if no such element is found.\n */\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.first(predicate: (UInt) -> Boolean): UInt {\n  for (element in this) if (predicate(element)) return element\n  throw NoSuchElementException("Array contains no element matching the predicate.")\n}\n\n/**\n * Returns the first element matching the given [predicate].\n * @throws [NoSuchElementException] if no such element is found.\n */\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.first(predicate: (UByte) -> Boolean): UByte {\n  for (element in this) if (predicate(element)) return element\n  throw NoSuchElementException("Array contains no element matching the predicate.")\n}\n\n/**\n * Returns the first element matching the given [predicate].\n * @throws [NoSuchElementException] if no such element is found.\n */\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.first(predicate: (UShort) -> Boolean): UShort {\n  for (element in this) if (predicate(element)) return element\n  throw NoSuchElementException("Array contains no element matching the predicate.")\n}\n\n/**\n * Returns the first element, or `null` if the array is empty.\n */\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.firstOrNull(): UInt? {\n  return if (isEmpty()) null else this[0]\n}\n\n/**\n * Returns the first element, or `null` if the array is empty.\n */\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.firstOrNull(): ULong? {\n

```

```

return if (isEmpty()) null else this[0]\n\n/**\n * Returns the first element, or `null` if the array is empty.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.firstOrNull(): UByte? {\n
return if (isEmpty()) null else this[0]\n\n/**\n * Returns the first element, or `null` if the array is empty.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.firstOrNull(): UShort? {\n
return if (isEmpty()) null else this[0]\n\n/**\n * Returns the first element matching the given [predicate], or `null` if element was not found.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.firstOrNull(predicate: (UInt) -> Boolean): UInt? {\n    for (element in this) if (predicate(element)) return
element\n    return null\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if element
was not found.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun ULongArray.firstOrNull(predicate: (ULong) -> Boolean): ULong? {\n    for (element in this) if
(predicate(element)) return element\n    return null\n}\n\n/**\n * Returns the first element matching the given
[predicate], or `null` if element was not found.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.firstOrNull(predicate: (UByte) -> Boolean): UByte? {\n    for (element in this) if (predicate(element))
return element\n    return null\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if
element was not found.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.firstOrNull(predicate: (UShort) -> Boolean): UShort? {\n    for (element in this) if (predicate(element))
return element\n    return null\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the
[defaultValue] function if the [index] is out of bounds of this array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.getOrNull(index: Int, default: (Int) -> UInt): UInt? {\n    return if (index >= 0 && index <=
lastIndex) get(index) else default(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of
calling the [defaultValue] function if the [index] is out of bounds of this array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.getOrNull(index: Int, default: (Int) -> ULong): ULong? {\n    return if (index >= 0 && index <=
lastIndex) get(index) else default(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of
calling the [defaultValue] function if the [index] is out of bounds of this array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.getOrNull(index: Int, default: (Int) -> UByte): UByte? {\n    return if (index >= 0 && index <=
lastIndex) get(index) else default(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of
calling the [defaultValue] function if the [index] is out of bounds of this array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.getOrNull(index: Int, default: (Int) -> UShort): UShort? {\n    return if (index >= 0 && index <=
lastIndex) get(index) else default(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if the
[index] is out of bounds of this array.\n
*\n * @sample samples.collections.Collections.Elements.getOrNull\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.getOrNull(index: Int): UInt? {\n
return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns an element at the given
[index] or `null` if the [index] is out of bounds of this array.\n
*\n * @sample
samples.collections.Collections.Elements.getOrNull\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.getOrNull(index: Int):
ULong? {\n    return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns an element at
the given [index] or `null` if the [index] is out of bounds of this array.\n
*\n * @sample
samples.collections.Collections.Elements.getOrNull\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.getOrNull(index: Int): UByte?
{\n    return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns an element at the
given [index] or `null` if the [index] is out of bounds of this array.\n
*\n * @sample

```

```

samples.collections.Collections.Elements.getOrNull\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.getOrNull(index: Int):
UShort? {\n    return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns first index of
[element], or -1 if the array does not contain element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.indexOf(element: UInt): Int {\n    return storage.indexOf(element.toInt())\n}\n\n/**\n * Returns first
index of [element], or -1 if the array does not contain element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.indexOf(element: ULong): Int {\n    return storage.indexOf(element.toLong())\n}\n\n/**\n * Returns
first index of [element], or -1 if the array does not contain element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.indexOf(element: UByte): Int {\n    return storage.indexOf(element.toByte())\n}\n\n/**\n * Returns
first index of [element], or -1 if the array does not contain element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.indexOf(element: UShort): Int {\n    return storage.indexOf(element.toShort())\n}\n\n/**\n * Returns
index of the first element matching the given [predicate], or -1 if the array does not contain such element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.indexOfFirst(predicate: (UInt) -> Boolean): Int {\n    return storage.indexOfFirst { predicate(it.toUInt())
}\n}\n\n/**\n * Returns index of the first element matching the given [predicate], or -1 if the array does not contain
such element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun ULongArray.indexOfFirst(predicate: (ULong) -> Boolean): Int {\n    return storage.indexOfFirst {
predicate(it.toULong()) }\n}\n\n/**\n * Returns index of the first element matching the given [predicate], or -1 if the
array does not contain such element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.indexOfFirst(predicate: (UByte) -> Boolean): Int {\n    return storage.indexOfFirst {
predicate(it.toUByte()) }\n}\n\n/**\n * Returns index of the first element matching the given [predicate], or -1 if the
array does not contain such element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.indexOfFirst(predicate: (UShort) -> Boolean): Int {\n    return storage.indexOfFirst {
predicate(it.toUShort()) }\n}\n\n/**\n * Returns index of the last element matching the given [predicate], or -1 if the
array does not contain such element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.indexOfLast(predicate: (UInt) -> Boolean): Int {\n    return storage.indexOfLast { predicate(it.toUInt())
}\n}\n\n/**\n * Returns index of the last element matching the given [predicate], or -1 if the array does not contain
such element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun ULongArray.indexOfLast(predicate: (ULong) -> Boolean): Int {\n    return storage.indexOfLast {
predicate(it.toULong()) }\n}\n\n/**\n * Returns index of the last element matching the given [predicate], or -1 if the
array does not contain such element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.indexOfLast(predicate: (UByte) -> Boolean): Int {\n    return storage.indexOfLast {
predicate(it.toUByte()) }\n}\n\n/**\n * Returns index of the last element matching the given [predicate], or -1 if the
array does not contain such element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.indexOfLast(predicate: (UShort) -> Boolean): Int {\n    return storage.indexOfLast {
predicate(it.toUShort()) }\n}\n\n/**\n * Returns the last element.\n * \n * @throws NoSuchElementException if the
array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.last(): UInt {\n    return storage.last().toUInt()\n}\n\n/**\n * Returns the last element.\n * \n * @throws

```

```

NoSuchElementException if the array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.last(): UByte {\n    return storage.last().toUByte()\n}\n\n/**\n * Returns the last element.\n * \n *
@throws NoSuchElementException if the array is empty.\n * \n * @sample
samples.collections.Collections.Elements.last\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.last(): UShort {\n    return storage.last().toUShort()\n}\n\n/**\n * Returns the last element matching
the given [predicate].\n * \n * @throws NoSuchElementException if no such element is found.\n * \n * @sample
samples.collections.Collections.Elements.last\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.last(predicate: (UInt) -> Boolean): UInt {\n    for (index in this.indices.reversed()) {\n        val element =
this[index]\n        if (predicate(element)) return element\n    }\n    throw NoSuchElementException("Array contains
no element matching the predicate.")\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n *
@throws NoSuchElementException if no such element is found.\n * \n * @sample
samples.collections.Collections.Elements.last\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.last(predicate: (UByte) -> Boolean): UByte {\n    for (index in this.indices.reversed()) {\n        val
element = this[index]\n        if (predicate(element)) return element\n    }\n    throw
NoSuchElementException("Array contains no element matching the predicate.")\n}\n\n/**\n * Returns the last
element matching the given [predicate].\n * \n * @throws NoSuchElementException if no such element is found.\n
*\n * @sample samples.collections.Collections.Elements.last\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.last(predicate: (UShort) -> Boolean): UShort {\n    for (index in this.indices.reversed()) {\n        val
element = this[index]\n        if (predicate(element)) return element\n    }\n    throw
NoSuchElementException("Array contains no element matching the predicate.")\n}\n\n/**\n * Returns last index
of [element], or -1 if the array does not contain element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.lastIndexOf(element: UInt): Int {\n    return storage.lastIndexOf(element.toInt())\n}\n\n/**\n * Returns
last index of [element], or -1 if the array does not contain element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.lastIndexOf(element: UByte): Int {\n    return storage.lastIndexOf(element.toByte())\n}\n\n/**\n *
Returns last index of [element], or -1 if the array does not contain element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.lastIndexOf(element: UShort): Int {\n    return storage.lastIndexOf(element.toShort())\n}\n\n/**\n *
Returns the last element, or `null` if the array is empty.\n * \n * @sample

```

```

samples.collections.Collections.Elements.last\n * \n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n public fun UIntArray.lastOrNull(): UInt? {\n     return if (isEmpty()) null else this[size - 1]\n }\n\n /**\n * Returns the last element, or `null` if the array is empty.\n */\n * \n * @sample samples.collections.Collections.Elements.last\n *\n *\n *\n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n public fun ULongArray.lastOrNull(): ULong? {\n     return if (isEmpty()) null else this[size - 1]\n }\n\n /**\n * Returns the last element, or `null` if the array is empty.\n */\n * \n * @sample samples.collections.Collections.Elements.last\n *\n *\n *\n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n public fun UByteArray.lastOrNull(): UByte? {\n     return if (isEmpty()) null else this[size - 1]\n }\n\n /**\n * Returns the last element, or `null` if the array is empty.\n */\n * \n * @sample samples.collections.Collections.Elements.last\n *\n *\n *\n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n public fun UShortArray.lastOrNull(): UShort? {\n     return if (isEmpty()) null else this[size - 1]\n }\n\n /**\n * Returns the last element matching the given [predicate], or `null` if no such element was found.\n */\n * \n * @sample samples.collections.Collections.Elements.last\n *\n *\n *\n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline fun UIntArray.lastOrNull(predicate: (UInt) -> Boolean): UInt? {\n     for (index in this.indices.reversed()) {\n         val element = this[index]\n         if (predicate(element)) return element\n     }\n     return null\n }\n\n /**\n * Returns the last element matching the given [predicate], or `null` if no such element was found.\n */\n * \n * @sample samples.collections.Collections.Elements.last\n *\n *\n *\n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline fun ULongArray.lastOrNull(predicate: (ULong) -> Boolean): ULong? {\n     for (index in this.indices.reversed()) {\n         val element = this[index]\n         if (predicate(element)) return element\n     }\n     return null\n }\n\n /**\n * Returns the last element matching the given [predicate], or `null` if no such element was found.\n */\n * \n * @sample samples.collections.Collections.Elements.last\n *\n *\n *\n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline fun UByteArray.lastOrNull(predicate: (UByte) -> Boolean): UByte? {\n     for (index in this.indices.reversed()) {\n         val element = this[index]\n         if (predicate(element)) return element\n     }\n     return null\n }\n\n /**\n * Returns the last element matching the given [predicate], or `null` if no such element was found.\n */\n * \n * @sample samples.collections.Collections.Elements.last\n *\n *\n *\n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline fun UShortArray.lastOrNull(predicate: (UShort) -> Boolean): UShort? {\n     for (index in this.indices.reversed()) {\n         val element = this[index]\n         if (predicate(element)) return element\n     }\n     return null\n }\n\n /**\n * Returns a random element from this array.\n * \n * @throws NoSuchElementException if this array is empty.\n */\n * \n * @sample samples.collections.Collections.Elements.last\n *\n *\n *\n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline fun UIntArray.random(): UInt {\n     return random(Random)\n }\n\n /**\n * Returns a random element from this array.\n * \n * @throws NoSuchElementException if this array is empty.\n */\n * \n * @sample samples.collections.Collections.Elements.last\n *\n *\n *\n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline fun ULongArray.random(): ULong {\n     return random(Random)\n }\n\n /**\n * Returns a random element from this array.\n * \n * @throws NoSuchElementException if this array is empty.\n */\n * \n * @sample samples.collections.Collections.Elements.last\n *\n *\n *\n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline fun UByteArray.random(): UByte {\n     return random(Random)\n }\n\n /**\n * Returns a random element from this array.\n * \n * @throws NoSuchElementException if this array is empty.\n */\n * \n * @sample samples.collections.Collections.Elements.last\n *\n *\n *\n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline fun UShortArray.random(): UShort {\n     return random(Random)\n }\n\n /**\n * Returns a random element from this array using the specified source of randomness.\n * \n * @throws NoSuchElementException if this array is empty.\n */\n * \n * @sample samples.collections.Collections.Elements.last\n *\n *\n *\n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n public fun UIntArray.random(random: Random): UInt {\n     if (isEmpty())\n         throw NoSuchElementException("Array is empty.")\n     return get(random.nextInt(size))\n }\n\n /**\n * Returns a random element from this array using the specified source of randomness.\n * \n * @throws NoSuchElementException if this array is empty.\n */\n * \n * @sample samples.collections.Collections.Elements.last\n *\n *\n *\n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n public fun ULongArray.random(random: Random):

```

```

ULong { \n  if (isEmpty())\n    throw NoSuchElementException("Array is empty.")\n  return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness.\n * \n * @throws NoSuchElementException if this array is empty.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.random(random: Random):
UByte { \n  if (isEmpty())\n    throw NoSuchElementException("Array is empty.")\n  return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness.\n * \n * @throws NoSuchElementException if this array is empty.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.random(random: Random):
UShort { \n  if (isEmpty())\n    throw NoSuchElementException("Array is empty.")\n  return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun UIntArray.randomOrNull(): UInt? { \n  return
randomOrNull(Random)\n}\n\n/**\n * Returns a random element from this array, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun ULongArray.randomOrNull(): ULong? { \n  return
randomOrNull(Random)\n}\n\n/**\n * Returns a random element from this array, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun UByteArray.randomOrNull(): UByte? { \n  return
randomOrNull(Random)\n}\n\n/**\n * Returns a random element from this array, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun UShortArray.randomOrNull(): UShort? { \n  return
randomOrNull(Random)\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
public fun UIntArray.randomOrNull(random: Random): UInt? { \n  if (isEmpty())\n    return null\n  return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
public fun ULongArray.randomOrNull(random: Random): ULong? { \n  if (isEmpty())\n    return null\n  return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
public fun UByteArray.randomOrNull(random: Random): UByte? { \n  if (isEmpty())\n    return null\n  return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness, or `null` if this array is empty.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
public fun UShortArray.randomOrNull(random: Random): UShort? { \n  if (isEmpty())\n    return null\n  return
get(random.nextInt(size))\n}\n\n/**\n * Returns the single element, or throws an exception if the array is empty or
has more than one element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.single(): UInt { \n  return storage.single().toUInt()\n}\n\n/**\n * Returns the single element, or throws an
exception if the array is empty or has more than one element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.single(): ULong { \n  return storage.single().toULong()\n}\n\n/**\n * Returns the single element, or
throws an exception if the array is empty or has more than one element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.single(): UByte { \n  return storage.single().toUByte()\n}\n\n/**\n * Returns the single element, or
throws an exception if the array is empty or has more than one element.\n

```

```

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.single(): UShort {\n    return storage.single().toUShort()\n}\n\n/**\n * Returns the single element
matching the given [predicate], or throws exception if there is no or more than one matching element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.single(predicate: (UInt) -> Boolean): UInt {\n    var single: UInt? = null\n    var found = false\n    for
(element in this) {\n        if (predicate(element)) {\n            if (found) throw IllegalArgumentException("Array
contains more than one matching element.")\n            single = element\n            found = true\n        }\n    }\n    if
(!found) throw NoSuchElementException("Array contains no element matching the predicate.")\n    @Suppress("UNCHECKED_CAST")\n    return single as UInt\n}\n\n/**\n * Returns the single element matching
the given [predicate], or throws exception if there is no or more than one matching element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.single(predicate: (ULong) -> Boolean): ULong {\n    var single: ULong? = null\n    var found = false\n    for
(element in this) {\n        if (predicate(element)) {\n            if (found) throw IllegalArgumentException("Array
contains more than one matching element.")\n            single = element\n            found = true\n        }\n    }\n    if
(!found) throw NoSuchElementException("Array contains no element matching the predicate.")\n    @Suppress("UNCHECKED_CAST")\n    return single as ULong\n}\n\n/**\n * Returns the single element
matching the given [predicate], or throws exception if there is no or more than one matching element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.single(predicate: (UByte) -> Boolean): UByte {\n    var single: UByte? = null\n    var found = false\n    for
(element in this) {\n        if (predicate(element)) {\n            if (found) throw IllegalArgumentException("Array
contains more than one matching element.")\n            single = element\n            found = true\n        }\n    }\n    if
(!found) throw NoSuchElementException("Array contains no element matching the predicate.")\n    @Suppress("UNCHECKED_CAST")\n    return single as UByte\n}\n\n/**\n * Returns the single element
matching the given [predicate], or throws exception if there is no or more than one matching element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.single(predicate: (UShort) -> Boolean): UShort {\n    var single: UShort? = null\n    var found = false\n    for
(element in this) {\n        if (predicate(element)) {\n            if (found) throw IllegalArgumentException("Array
contains more than one matching element.")\n            single = element\n            found = true\n        }\n    }\n    if
(!found) throw NoSuchElementException("Array contains no element matching the predicate.")\n    @Suppress("UNCHECKED_CAST")\n    return single as UShort\n}\n\n/**\n * Returns single element, or `null` if
the array is empty or has more than one element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.singleOrNull(): UInt? {\n    return if (size == 1) this[0] else null\n}\n\n/**\n * Returns single element, or `null` if the array is empty or has more
than one element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
ULongArray.singleOrNull(): ULong? {\n    return if (size == 1) this[0] else null\n}\n\n/**\n * Returns single
element, or `null` if the array is empty or has more than one element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
UByteArray.singleOrNull(): UByte? {\n    return if (size == 1) this[0] else null\n}\n\n/**\n * Returns single
element, or `null` if the array is empty or has more than one element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
UShortArray.singleOrNull(): UShort? {\n    return if (size == 1) this[0] else null\n}\n\n/**\n * Returns the single
element matching the given [predicate], or `null` if element was not found or more than one element was found.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.singleOrNull(predicate: (UInt) -> Boolean): UInt? {\n    var single: UInt? = null\n    var found = false\n    for
(element in this) {\n        if (predicate(element)) {\n            if (found) return null\n            single = element\n            found = true\n        }\n    }\n    if (!found) return null\n    return single\n}\n\n/**\n * Returns the single element
matching the given [predicate], or `null` if element was not found or more than one element was found.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.singleOrNull(predicate: (ULong) -> Boolean): ULong? {\n    var single: ULong? = null\n    var found

```

```

= false\n for (element in this) {\n if (predicate(element)) {\n if (found) return null\n single =
element\n found = true\n }\n }\n if (!found) return null\n return single\n}\n\n/**\n * Returns the
single element matching the given [predicate], or `null` if element was not found or more than one element was
found.\n *\n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline
fun UByteArray.singleOrNull(predicate: (UByte) -> Boolean): UByte? {\n var single: UByte? = null\n var found
= false\n for (element in this) {\n if (predicate(element)) {\n if (found) return null\n single =
element\n found = true\n }\n }\n if (!found) return null\n return single\n}\n\n/**\n * Returns the
single element matching the given [predicate], or `null` if element was not found or more than one element was
found.\n *\n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline
fun UShortArray.singleOrNull(predicate: (UShort) -> Boolean): UShort? {\n var single: UShort? = null\n var
found = false\n for (element in this) {\n if (predicate(element)) {\n if (found) return null\n single
= element\n found = true\n }\n }\n if (!found) return null\n return single\n}\n\n/**\n * Returns a list
containing all elements except first [n] elements.\n *\n * @throws IllegalArgumentException if [n] is negative.\n *
\n *\n @sample samples.collections.Collections.Transformations.drop\n
*\n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n public fun UIntArray.drop(n: Int): List<UInt> {\n
require(n >= 0) { "Requested element count $n is less than zero." }\n return takeLast((size -
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except first [n] elements.\n *\n * @throws
IllegalArgumentException if [n] is negative.\n *\n * @sample
samples.collections.Collections.Transformations.drop\n
*\n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n public fun ULongArray.drop(n: Int): List<ULong> {\n
require(n >= 0) { "Requested element count $n is less than zero." }\n return takeLast((size -
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except first [n] elements.\n *\n * @throws
IllegalArgumentException if [n] is negative.\n *\n * @sample
samples.collections.Collections.Transformations.drop\n
*\n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n public fun UByteArray.drop(n: Int): List<UByte> {\n
require(n >= 0) { "Requested element count $n is less than zero." }\n return takeLast((size -
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except first [n] elements.\n *\n * @throws
IllegalArgumentException if [n] is negative.\n *\n * @sample
samples.collections.Collections.Transformations.drop\n
*\n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n public fun UShortArray.drop(n: Int): List<UShort> {\n
require(n >= 0) { "Requested element count $n is less than zero." }\n return takeLast((size -
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n *\n * @throws
IllegalArgumentException if [n] is negative.\n *\n * @sample
samples.collections.Collections.Transformations.drop\n
*\n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n public fun UIntArray.dropLast(n: Int): List<UInt> {\n
require(n >= 0) { "Requested element count $n is less than zero." }\n return take((size -
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n *\n * @throws
IllegalArgumentException if [n] is negative.\n *\n * @sample
samples.collections.Collections.Transformations.drop\n
*\n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n public fun ULongArray.dropLast(n: Int):
List<ULong> {\n require(n >= 0) { "Requested element count $n is less than zero." }\n return take((size -
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n *\n * @throws
IllegalArgumentException if [n] is negative.\n *\n * @sample
samples.collections.Collections.Transformations.drop\n
*\n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n public fun UByteArray.dropLast(n: Int): List<UByte>
{\n require(n >= 0) { "Requested element count $n is less than zero." }\n return take((size -
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n *\n * @throws
IllegalArgumentException if [n] is negative.\n *\n * @sample

```

```

samples.collections.Collections.Transformations.drop\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.dropLast(n: Int):
List<UShort> {\n    require(n >= 0) { "Requested element count $n is less than zero." }\n    return take((size -
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last elements that satisfy the given
[predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.dropLastWhile(predicate: (UInt) -> Boolean): List<UInt> {\n    for (index in lastIndex downTo 0) {\n
if (!predicate(this[index])) {\n        return take(index + 1)\n    }\n}\n    return emptyList()\n}\n\n/**\n *
Returns a list containing all elements except last elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.dropLastWhile(predicate: (ULong) -> Boolean): List<ULong> {\n    for (index in lastIndex downTo 0)
{\n        if (!predicate(this[index])) {\n            return take(index + 1)\n        }\n    }\n    return emptyList()\n}\n\n/**\n *
Returns a list containing all elements except last elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.dropLastWhile(predicate: (UByte) -> Boolean): List<UByte> {\n    for (index in lastIndex downTo 0)
{\n        if (!predicate(this[index])) {\n            return take(index + 1)\n        }\n    }\n    return emptyList()\n}\n\n/**\n *
Returns a list containing all elements except last elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.dropLastWhile(predicate: (UShort) -> Boolean): List<UShort> {\n    for (index in lastIndex downTo
0) {\n        if (!predicate(this[index])) {\n            return take(index + 1)\n        }\n    }\n    return
emptyList()\n}\n\n/**\n * Returns a list containing all elements except first elements that satisfy the given
[predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.dropWhile(predicate: (UInt) -> Boolean): List<UInt> {\n    var yielding = false\n    val list =
ArrayList<UInt>()\n    for (item in this)\n        if (yielding)\n            list.add(item)\n        else if (!predicate(item)) {\n
            list.add(item)\n            yielding = true\n        }\n    return list\n}\n\n/**\n * Returns a list containing all
elements except first elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.dropWhile(predicate: (ULong) -> Boolean): List<ULong> {\n    var yielding = false\n    val list =
ArrayList<ULong>()\n    for (item in this)\n        if (yielding)\n            list.add(item)\n        else if (!predicate(item))
{\n            list.add(item)\n            yielding = true\n        }\n    return list\n}\n\n/**\n * Returns a list containing all
elements except first elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.dropWhile(predicate: (UByte) -> Boolean): List<UByte> {\n    var yielding = false\n    val list =
ArrayList<UByte>()\n    for (item in this)\n        if (yielding)\n            list.add(item)\n        else if (!predicate(item))
{\n            list.add(item)\n            yielding = true\n        }\n    return list\n}\n\n/**\n * Returns a list containing all
elements except first elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.dropWhile(predicate: (UShort) -> Boolean): List<UShort> {\n    var yielding = false\n    val list =
ArrayList<UShort>()\n    for (item in this)\n        if (yielding)\n            list.add(item)\n        else if (!predicate(item))
{\n            list.add(item)\n            yielding = true\n        }\n    return list\n}\n\n/**\n * Returns a list containing only
elements matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n

```

```

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.filter(predicate: (UInt) -> Boolean): List<UInt> {\n  return filterTo(ArrayList<UInt>(),
predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.filter(predicate: (ULong) -> Boolean): List<ULong> {\n  return filterTo(ArrayList<ULong>(),
predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.filter(predicate: (UByte) -> Boolean): List<UByte> {\n  return filterTo(ArrayList<UByte>(),
predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * \n * @sample
samples.collections.Collections.Filtering.filter\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.filter(predicate: (UShort) -> Boolean): List<UShort> {\n  return filterTo(ArrayList<UShort>(),
predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * @param
[predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate
evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexed\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.filterIndexed(predicate: (index: Int, UInt) -> Boolean): List<UInt> {\n  return
filterIndexedTo(ArrayList<UInt>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the
given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself\n * and
returns the result of predicate evaluation on the element.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexed\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.filterIndexed(predicate: (index: Int, ULong) -> Boolean): List<ULong> {\n  return
filterIndexedTo(ArrayList<ULong>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the
given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself\n * and
returns the result of predicate evaluation on the element.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexed\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.filterIndexed(predicate: (index: Int, UByte) -> Boolean): List<UByte> {\n  return
filterIndexedTo(ArrayList<UByte>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the
given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself\n * and
returns the result of predicate evaluation on the element.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexed\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.filterIndexed(predicate: (index: Int, UShort) -> Boolean): List<UShort> {\n  return
filterIndexedTo(ArrayList<UShort>(), predicate)\n}\n\n/**\n * Appends all elements matching the given [predicate]
to the given [destination].\n * @param [predicate] function that takes the index of an element and the element
itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample
samples.collections.Collections.Filtering.filterIndexedTo\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <C :
MutableCollection<in UInt>> UIntArray.filterIndexedTo(destination: C, predicate: (index: Int, UInt) -> Boolean): C
{\n  forEachIndexed { index, element ->\n    if (predicate(index, element)) destination.add(element)\n  }\n  return destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n *
@param [predicate] function that takes the index of an element and the element itself\n * and returns the result of
predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexedTo\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <C :

```

```

MutableCollection<in ULong>> ULongArray.filterIndexedTo(destination: C, predicate: (index: Int, ULong) ->
Boolean): C {
    forEachIndexed { index, element ->
        if (predicate(index, element))
            destination.add(element)
    }
    return destination
}
Append all elements matching the given [predicate] to the given [destination].
@param [predicate] function that takes the index of an element and the element itself
and returns the result of predicate evaluation on the element.
@sample samples.collections.Collections.Filtering.filterIndexedTo

*SinceKotlin("1.3")ExperimentalUnsignedTypes@kotlin.internal.InlineOnlypublic inline fun <C :
MutableCollection<in UByte>> UByteArray.filterIndexedTo(destination: C, predicate: (index: Int, UByte) ->
Boolean): C {
    forEachIndexed { index, element ->
        if (predicate(index, element))
            destination.add(element)
    }
    return destination
}
Append all elements matching the given [predicate] to the given [destination].
@param [predicate] function that takes the index of an element and the element itself
and returns the result of predicate evaluation on the element.
@sample samples.collections.Collections.Filtering.filterIndexedTo

*SinceKotlin("1.3")ExperimentalUnsignedTypes@kotlin.internal.InlineOnlypublic inline fun <C :
MutableCollection<in UShort>> UShortArray.filterIndexedTo(destination: C, predicate: (index: Int, UShort) ->
Boolean): C {
    forEachIndexed { index, element ->
        if (predicate(index, element))
            destination.add(element)
    }
    return destination
}
Returns a list containing all elements not matching the given [predicate].
@sample samples.collections.Collections.Filtering.filter

*SinceKotlin("1.3")ExperimentalUnsignedTypes@kotlin.internal.InlineOnlypublic inline fun
UIntArray.filterNot(predicate: (UInt) -> Boolean): List<UInt> {
    return filterNotTo(ArrayList<UInt>(), predicate)
}
Returns a list containing all elements not matching the given [predicate].
@sample samples.collections.Collections.Filtering.filter

*SinceKotlin("1.3")ExperimentalUnsignedTypes@kotlin.internal.InlineOnlypublic inline fun
ULongArray.filterNot(predicate: (ULong) -> Boolean): List<ULong> {
    return filterNotTo(ArrayList<ULong>(), predicate)
}
Returns a list containing all elements not matching the given [predicate].
@sample samples.collections.Collections.Filtering.filter

*SinceKotlin("1.3")ExperimentalUnsignedTypes@kotlin.internal.InlineOnlypublic inline fun
UByteArray.filterNot(predicate: (UByte) -> Boolean): List<UByte> {
    return filterNotTo(ArrayList<UByte>(), predicate)
}
Returns a list containing all elements not matching the given [predicate].
@sample samples.collections.Collections.Filtering.filter

*SinceKotlin("1.3")ExperimentalUnsignedTypes@kotlin.internal.InlineOnlypublic inline fun
UShortArray.filterNot(predicate: (UShort) -> Boolean): List<UShort> {
    return filterNotTo(ArrayList<UShort>(), predicate)
}
Append all elements not matching the given [predicate] to the given [destination].
@sample samples.collections.Collections.Filtering.filterTo

*SinceKotlin("1.3")ExperimentalUnsignedTypes@kotlin.internal.InlineOnlypublic inline fun <C :
MutableCollection<in UInt>> UIntArray.filterNotTo(destination: C, predicate: (UInt) -> Boolean): C {
    for (element in this) if (!predicate(element)) destination.add(element)
    return destination
}
Append all elements not matching the given [predicate] to the given [destination].
@sample samples.collections.Collections.Filtering.filterTo

*SinceKotlin("1.3")ExperimentalUnsignedTypes@kotlin.internal.InlineOnlypublic inline fun <C :
MutableCollection<in ULong>> ULongArray.filterNotTo(destination: C, predicate: (ULong) -> Boolean): C {
    for (element in this) if (!predicate(element)) destination.add(element)
    return destination
}
Append all elements not matching the given [predicate] to the given [destination].
@sample samples.collections.Collections.Filtering.filterTo

*SinceKotlin("1.3")ExperimentalUnsignedTypes@kotlin.internal.InlineOnlypublic inline fun <C :
MutableCollection<in UByte>> UByteArray.filterNotTo(destination: C, predicate: (UByte) -> Boolean): C {
    for (element in this) if (!predicate(element)) destination.add(element)
    return destination
}
Append all elements not matching the given [predicate] to the given [destination].
@sample samples.collections.Collections.Filtering.filterTo

```

```

samples.collections.Collections.Filtering.filterTo\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <C :
MutableCollection<in UShort>> UShortArray.filterNotTo(destination: C, predicate: (UShort) -> Boolean): C {\n
for (element in this) if (!predicate(element)) destination.add(element)\n return destination\n}\n\n/**\n * Appends
all elements matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <C :
MutableCollection<in UInt>> UIntArray.filterTo(destination: C, predicate: (UInt) -> Boolean): C {\n
for (element
in this) if (predicate(element)) destination.add(element)\n return destination\n}\n\n/**\n * Appends all elements
matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <C :
MutableCollection<in ULong>> ULongArray.filterTo(destination: C, predicate: (ULong) -> Boolean): C {\n
for
(element in this) if (predicate(element)) destination.add(element)\n return destination\n}\n\n/**\n * Appends all
elements matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <C :
MutableCollection<in UByte>> UByteArray.filterTo(destination: C, predicate: (UByte) -> Boolean): C {\n
for
(element in this) if (predicate(element)) destination.add(element)\n return destination\n}\n\n/**\n * Appends all
elements matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <C :
MutableCollection<in UShort>> UShortArray.filterTo(destination: C, predicate: (UShort) -> Boolean): C {\n
for
(element in this) if (predicate(element)) destination.add(element)\n return destination\n}\n\n/**\n * Returns a list
containing elements at indices in the specified [indices] range.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.slice(indices: IntRange):
List<UInt> {\n
if (indices.isEmpty()) return listOf()\n return copyOfRange(indices.start, indices.endInclusive +
1).asList()\n}\n\n/**\n * Returns a list containing elements at indices in the specified [indices] range.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.slice(indices: IntRange):
List<ULong> {\n
if (indices.isEmpty()) return listOf()\n return copyOfRange(indices.start, indices.endInclusive
+ 1).asList()\n}\n\n/**\n * Returns a list containing elements at indices in the specified [indices] range.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.slice(indices: IntRange):
List<UByte> {\n
if (indices.isEmpty()) return listOf()\n return copyOfRange(indices.start, indices.endInclusive
+ 1).asList()\n}\n\n/**\n * Returns a list containing elements at indices in the specified [indices] range.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.slice(indices: IntRange):
List<UShort> {\n
if (indices.isEmpty()) return listOf()\n return copyOfRange(indices.start, indices.endInclusive
+ 1).asList()\n}\n\n/**\n * Returns a list containing elements at specified [indices].\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.slice(indices: Iterable<Int>):
List<UInt> {\n
val size = indices.collectionSizeOrDefault(10)\n if (size == 0) return emptyList()\n val list =
ArrayList<UInt>(size)\n for (index in indices) {\n list.add(get(index))\n }\n return list\n}\n\n/**\n *
Returns a list containing elements at specified [indices].\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.slice(indices: Iterable<Int>):
List<ULong> {\n
val size = indices.collectionSizeOrDefault(10)\n if (size == 0) return emptyList()\n val list =
ArrayList<ULong>(size)\n for (index in indices) {\n list.add(get(index))\n }\n return list\n}\n\n/**\n *
Returns a list containing elements at specified [indices].\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.slice(indices: Iterable<Int>):
List<UByte> {\n
val size = indices.collectionSizeOrDefault(10)\n if (size == 0) return emptyList()\n val list =
ArrayList<UByte>(size)\n for (index in indices) {\n list.add(get(index))\n }\n return list\n}\n\n/**\n *
Returns a list containing elements at specified [indices].\n

```

Returns a list containing elements at specified [indices].\n

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.slice(indices: Iterable<Int>):  
List<UShort> {\n    val size = indices.collectionSizeOrDefault(10)\n    if (size == 0) return emptyList()\n    val list =  
ArrayList<UShort>(size)\n    for (index in indices) {\n        list.add(get(index))\n    }\n    return list\n}\n\n/**\n *
```

Returns an array containing elements of this array at specified [indices].\n

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.sliceArray(indices:  
Collection<Int>): UIntArray {\n    return UIntArray(storage.sliceArray(indices))\n}\n\n/**\n * Returns an array  
containing elements of this array at specified [indices].\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.sliceArray(indices:  
Collection<Int>): ULongArray {\n    return ULongArray(storage.sliceArray(indices))\n}\n\n/**\n * Returns an array  
containing elements of this array at specified [indices].\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.sliceArray(indices:  
Collection<Int>): UByteArray {\n    return UByteArray(storage.sliceArray(indices))\n}\n\n/**\n * Returns an array  
containing elements of this array at specified [indices].\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.sliceArray(indices:  
Collection<Int>): UShortArray {\n    return UShortArray(storage.sliceArray(indices))\n}\n\n/**\n * Returns an  
array containing elements at indices in the specified [indices] range.\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.sliceArray(indices: IntRange):  
UIntArray {\n    return UIntArray(storage.sliceArray(indices))\n}\n\n/**\n * Returns an array containing elements at  
indices in the specified [indices] range.\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun  
ULongArray.sliceArray(indices: IntRange): ULongArray {\n    return  
ULongArray(storage.sliceArray(indices))\n}\n\n/**\n * Returns an array containing elements at indices in the  
specified [indices] range.\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun  
UByteArray.sliceArray(indices: IntRange): UByteArray {\n    return  
UByteArray(storage.sliceArray(indices))\n}\n\n/**\n * Returns an array containing elements at indices in the  
specified [indices] range.\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun  
UShortArray.sliceArray(indices: IntRange): UShortArray {\n    return  
UShortArray(storage.sliceArray(indices))\n}\n\n/**\n * Returns a list containing first [n] elements.\n
```

@throws IllegalArgumentException if [n] is negative.\n * \n * @sample

samples.collections.Collections.Transformations.take\n

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.take(n: Int): List<UInt> {\n    require(n >= 0) { "Requested element count $n is less than zero." }\n    if (n == 0) return emptyList()\n    if (n >=  
size) return toList()\n    if (n == 1) return listOf(this[0])\n    var count = 0\n    val list = ArrayList<UInt>(n)\n    for  
(item in this) {\n        list.add(item)\n        if (++count == n)\n            break\n    }\n    return list\n}\n\n/**\n * Returns  
a list containing first [n] elements.\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.take(n: Int): List<ULong> {\n    require(n >= 0) { "Requested element count $n is less than zero." }\n    if (n == 0) return emptyList()\n    if (n >=  
size) return toList()\n    if (n == 1) return listOf(this[0])\n    var count = 0\n    val list = ArrayList<ULong>(n)\n    for  
(item in this) {\n        list.add(item)\n        if (++count == n)\n            break\n    }\n    return list\n}\n\n/**\n * Returns  
a list containing first [n] elements.\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.take(n: Int): List<UByte> {\n    require(n >= 0) { "Requested element count $n is less than zero." }\n    if (n == 0) return emptyList()\n    if (n >=  
size) return toList()\n    if (n == 1) return listOf(this[0])\n    var count = 0\n    val list = ArrayList<UByte>(n)\n    for  
(item in this) {\n        list.add(item)\n        if (++count == n)\n            break\n    }\n    return list\n}\n\n/**\n * Returns  
a list containing first [n] elements.\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun  
UShortArray.take(n: Int): List<UShort> {\n    require(n >= 0) { "Requested element count $n is less than zero." }\n    if (n == 0) return emptyList()\n    if (n >=  
size) return toList()\n    if (n == 1) return listOf(this[0])\n    var count = 0\n    val list = ArrayList<UShort>(n)\n    for  
(item in this) {\n        list.add(item)\n        if (++count == n)\n            break\n    }\n    return list\n}\n\n/**\n * Returns  
a list containing first [n] elements.\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.take(n: Int): List<UShort> {\n    require(n >= 0) { \"Requested element count $n is less than zero.\" }\n    if (n == 0) return emptyList()\n    if (n >= size) return toList()\n    if (n == 1) return listOf(this[0])\n    var count = 0\n    val list = ArrayList<UShort>(n)\n    for (item in this) {\n        list.add(item)\n        if (++count == n)\n            break\n    }\n    return list\n}\n\n/**\n * Returns a list containing last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.take\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.takeLast(n: Int): List<UInt> {\n    require(n >= 0) { \"Requested element count $n is less than zero.\" }\n    if (n == 0) return emptyList()\n    val size = size\n    if (n >= size) return toList()\n    if (n == 1) return listOf(this[size - 1])\n    val list = ArrayList<UInt>(n)\n    for (index in size - n until size)\n        list.add(this[index])\n    return list\n}\n\n/**\n * Returns a list containing last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.take\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.takeLast(n: Int): List<ULong> {\n    require(n >= 0) { \"Requested element count $n is less than zero.\" }\n    if (n == 0) return emptyList()\n    val size = size\n    if (n >= size) return toList()\n    if (n == 1) return listOf(this[size - 1])\n    val list = ArrayList<ULong>(n)\n    for (index in size - n until size)\n        list.add(this[index])\n    return list\n}\n\n/**\n * Returns a list containing last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.take\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.takeLast(n: Int): List<UByte> {\n    require(n >= 0) { \"Requested element count $n is less than zero.\" }\n    if (n == 0) return emptyList()\n    val size = size\n    if (n >= size) return toList()\n    if (n == 1) return listOf(this[size - 1])\n    val list = ArrayList<UByte>(n)\n    for (index in size - n until size)\n        list.add(this[index])\n    return list\n}\n\n/**\n * Returns a list containing last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.take\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.takeLastWhile(predicate: (UShort) -> Boolean): List<UShort> {\n    require(n >= 0) { \"Requested element count $n is less than zero.\" }\n    if (n == 0) return emptyList()\n    val size = size\n    if (n >= size) return toList()\n    if (n == 1) return listOf(this[size - 1])\n    val list = ArrayList<UShort>(n)\n    for (index in size - n until size)\n        list.add(this[index])\n    return list\n}\n\n/**\n * Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.takeLastWhile(predicate: (UInt) -> Boolean): List<UInt> {\n    for (index in lastIndex downTo 0) {\n        if (!predicate(this[index]))\n            return drop(index + 1)\n    }\n    return toList()\n}\n\n/**\n * Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.takeLastWhile(predicate: (ULong) -> Boolean): List<ULong> {\n    for (index in lastIndex downTo 0)\n        if (!predicate(this[index]))\n            return drop(index + 1)\n    return toList()\n}\n\n/**\n * Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.takeLastWhile(predicate: (UByte) -> Boolean): List<UByte> {\n    for (index in lastIndex downTo 0)\n        if (!predicate(this[index]))\n            return drop(index + 1)\n    return toList()\n}\n\n/**\n * Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.takeLastWhile(predicate: (UShort) -> Boolean): List<UShort> {\n    for (index in lastIndex downTo 0)\n        if (!predicate(this[index]))\n            return drop(index + 1)\n    return toList()\n}\n\n/**\n * Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n
```

Returns a list containing first elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.takeWhile(predicate: (UInt) -> Boolean): List<UInt> {\n val list = ArrayList<UInt>()\n for (item in
this) {\n if (!predicate(item))\n break\n list.add(item)\n }\n return list\n}\n\n/**\n * Returns a list
containing first elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.takeWhile(predicate: (ULong) -> Boolean): List<ULong> {\n val list = ArrayList<ULong>()\n for
(item in this) {\n if (!predicate(item))\n break\n list.add(item)\n }\n return list\n}\n\n/**\n *
Returns a list containing first elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.takeWhile(predicate: (UByte) -> Boolean): List<UByte> {\n val list = ArrayList<UByte>()\n for
(item in this) {\n if (!predicate(item))\n break\n list.add(item)\n }\n return list\n}\n\n/**\n *
Returns a list containing first elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.takeWhile(predicate: (UShort) -> Boolean): List<UShort> {\n val list = ArrayList<UShort>()\n for
(item in this) {\n if (!predicate(item))\n break\n list.add(item)\n }\n return list\n}\n\n/**\n *
Reverses elements in the array in-place.\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.reverse(): Unit {\n storage.reverse()\n}\n\n/**\n * Reverses elements in the array in-place.\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.reverse(): Unit {\n storage.reverse()\n}\n\n/**\n * Reverses elements in the array in-place.\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.reverse(): Unit {\n storage.reverse()\n}\n\n/**\n * Reverses elements in the array in-place.\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.reverse(): Unit {\n storage.reverse()\n}\n\n/**\n * Reverses elements of the array in the specified
range in-place.\n * \n * @param fromIndex the start of the range (inclusive) to reverse.\n * @param toIndex the end
of the range (exclusive) to reverse.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero
or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater
than [toIndex].\n
```

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun UIntArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n storage.reverse(fromIndex,
toIndex)\n}\n\n/**\n * Reverses elements of the array in the specified range in-place.\n * \n * @param fromIndex
the start of the range (inclusive) to reverse.\n * @param toIndex the end of the range (exclusive) to reverse.\n * \n *
@throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this
array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
```

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n storage.reverse(fromIndex, toIndex)\n}\n\n/**\n *
Reverses elements of the array in the specified range in-place.\n * \n * @param fromIndex the start of the range
(inclusive) to reverse.\n * @param toIndex the end of the range (exclusive) to reverse.\n * \n * @throws
IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n *
@throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
```

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n storage.reverse(fromIndex, toIndex)\n}\n\n/**\n *
Reverses elements of the array in the specified range in-place.\n * \n * @param fromIndex the start of the range
(inclusive) to reverse.\n * @param toIndex the end of the range (exclusive) to reverse.\n * \n * @throws
IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n *
@throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
```

IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n *
 @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n

```

*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n    storage.reverse(fromIndex, toIndex)\n}\n\n/**\n *
Returns a list with elements in reversed order.\n *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic
fun UIntArray.reversed(): List<UInt> {\n    if (isEmpty()) return emptyList()\n    val list = toMutableList()\n
list.reverse()\n    return list\n}\n\n/**\n * Returns a list with elements in reversed order.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.reversed(): List<ULong> {\n
if (isEmpty()) return emptyList()\n    val list = toMutableList()\n    list.reverse()\n    return list\n}\n\n/**\n * Returns
a list with elements in reversed order.\n *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
UByteArray.reversed(): List<UByte> {\n    if (isEmpty()) return emptyList()\n    val list = toMutableList()\n
list.reverse()\n    return list\n}\n\n/**\n * Returns a list with elements in reversed order.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.reversed(): List<UShort> {\n
if (isEmpty()) return emptyList()\n    val list = toMutableList()\n    list.reverse()\n    return list\n}\n\n/**\n * Returns
an array with elements of this array in reversed order.\n *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.reversedArray(): UIntArray {\n    return UIntArray(storage.reversedArray())\n}\n\n/**\n * Returns an
array with elements of this array in reversed order.\n *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.reversedArray(): ULongArray {\n    return ULongArray(storage.reversedArray())\n}\n\n/**\n * Returns
an array with elements of this array in reversed order.\n *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.reversedArray(): UByteArray {\n    return UByteArray(storage.reversedArray())\n}\n\n/**\n * Returns
an array with elements of this array in reversed order.\n *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.reversedArray(): UShortArray {\n    return UShortArray(storage.reversedArray())\n}\n\n/**\n *
Randomly shuffles elements in this array in-place.\n *\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.shuffle(): Unit {\n
shuffle(Random)\n}\n\n/**\n * Randomly shuffles elements in this array in-place.\n *\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.shuffle(): Unit {\n
shuffle(Random)\n}\n\n/**\n * Randomly shuffles elements in this array in-place.\n *\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.shuffle(): Unit {\n
shuffle(Random)\n}\n\n/**\n * Randomly shuffles elements in this array in-place.\n *\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.shuffle(): Unit {\n
shuffle(Random)\n}\n\n/**\n * Randomly shuffles elements in this array in-place using the specified [random]
instance as the source of randomness.\n *\n * See:
https://en.wikipedia.org/wiki/Fisher%20%93Yates\_shuffle#The\_modern\_algorithm\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.shuffle(random: Random): Unit
{\n    for (i in lastIndex downTo 1) {\n        val j = random.nextInt(i + 1)\n        val copy = this[i]\n        this[i] =
this[j]\n        this[j] = copy\n    }\n}\n\n/**\n * Randomly shuffles elements in this array in-place using the specified
[random] instance as the source of randomness.\n *\n * See:
https://en.wikipedia.org/wiki/Fisher%20%93Yates\_shuffle#The\_modern\_algorithm\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.shuffle(random: Random):
Unit {\n    for (i in lastIndex downTo 1) {\n        val j = random.nextInt(i + 1)\n        val copy = this[i]\n        this[i] =
this[j]\n        this[j] = copy\n    }\n}\n\n/**\n * Randomly shuffles elements in this array in-place using the specified
[random] instance as the source of randomness.\n *\n * See:
https://en.wikipedia.org/wiki/Fisher%20%93Yates\_shuffle#The\_modern\_algorithm\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.shuffle(random: Random):

```

```

Unit {
    for (i in lastIndex downTo 1) {
        val j = random.nextInt(i + 1)
        val copy = this[i]
        this[i] =
        this[j]
        this[j] = copy
    }
}

```

* Randomly shuffles elements in this array in-place using the specified [random] instance as the source of randomness.

* See:
https://en.wikipedia.org/wiki/Fisher%20%80%93Yates_shuffle#The_modern_algorithm

```

@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun UShortArray.shuffle(random: Random):
Unit {
    for (i in lastIndex downTo 1) {
        val j = random.nextInt(i + 1)
        val copy = this[i]
        this[i] =
        this[j]
        this[j] = copy
    }
}

```

* Sorts elements in the array in-place descending according to their natural sort order.

```

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun
UIntArray.sortDescending(): Unit {
    if (size > 1) {
        sort()
        reverse()
    }
}

```

* Sorts elements in the array in-place descending according to their natural sort order.

```

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.sortDescending(): Unit {
    if (size > 1) {
        sort()
        reverse()
    }
}

```

* Sorts elements in the array in-place descending according to their natural sort order.

```

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.sortDescending(): Unit {
    if (size > 1) {
        sort()
        reverse()
    }
}

```

* Sorts elements in the array in-place descending according to their natural sort order.

```

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.sortDescending(): Unit {
    if (size > 1) {
        sort()
        reverse()
    }
}

```

* Returns a list of all elements sorted according to their natural sort order.

```

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.sorted():
List<UInt> {
    return copyOf().apply { sort() }.asList()
}

```

* Returns a list of all elements sorted according to their natural sort order.

```

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.sorted(): List<ULong> {
    return copyOf().apply { sort() }.asList()
}

```

* Returns a list of all elements sorted according to their natural sort order.

```

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.sorted(): List<UByte> {
    return copyOf().apply { sort() }.asList()
}

```

* Returns a list of all elements sorted according to their natural sort order.

```

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.sorted():
List<UShort> {
    return copyOf().apply { sort() }.asList()
}

```

* Returns an array with all elements of this array sorted according to their natural sort order.

```

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.sortedArray(): UIntArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sort() }
}

```

* Returns an array with all elements of this array sorted according to their natural sort order.

```

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.sortedArray(): ULongArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sort() }
}

```

* Returns an array with all elements of this array sorted according to their natural sort order.

```

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.sortedArray(): UByteArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sort() }
}

```

* Returns an array with all elements of this array sorted according to their natural sort order.

```

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.sortedArray(): UShortArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sort() }
}

```

* Returns an array with all elements of this array sorted descending according to their natural sort order.

```

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.sortedArrayDescending():
UIntArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sortDescending() }
}

```

* Returns an array with all elements of this array sorted descending according to their natural sort order.

```

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.sortedArrayDescending():
ULongArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sortDescending() }
}

```

* Returns an array with all elements of this array sorted descending according to their natural sort order.

```

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.sortedArrayDescending():
UByteArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sortDescending() }
}

```

* Returns an array with all elements of this array sorted descending according to their natural sort order.

```

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.sortedArrayDescending():
UShortArray {\n  if (isEmpty()) return this\n  return this.copyOf().apply { sortDescending() }\n}\n\n/**\n *
Returns a list of all elements sorted descending according to their natural sort order.\n * \n * The sort is _stable_. It
means that equal elements preserve their order relative to each other after sorting.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.sortedDescending(): List<UInt>
{\n  return copyOf().apply { sort() }.reversed()\n}\n\n/**\n * Returns a list of all elements sorted descending
according to their natural sort order.\n * \n * The sort is _stable_. It means that equal elements preserve their order
relative to each other after sorting.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
ULongArray.sortedDescending(): List<ULong> {\n  return copyOf().apply { sort() }.reversed()\n}\n\n/**\n *
Returns a list of all elements sorted descending according to their natural sort order.\n * \n * The sort is _stable_. It
means that equal elements preserve their order relative to each other after sorting.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
UByteArray.sortedDescending():
List<UByte> {\n  return copyOf().apply { sort() }.reversed()\n}\n\n/**\n * Returns a list of all elements sorted
descending according to their natural sort order.\n * \n * The sort is _stable_. It means that equal elements preserve
their order relative to each other after sorting.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic
fun UShortArray.sortedDescending(): List<UShort> {\n  return copyOf().apply { sort() }.reversed()\n}\n\n/**\n *
Returns an array of type [ByteArray], which is a view of this array where each element is a signed reinterpretation\n
* of the corresponding element of this array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.asByteArray(): ByteArray {\n  return storage\n}\n\n/**\n * Returns an array of type [IntArray], which
is a view of this array where each element is a signed reinterpretation\n * of the corresponding element of this
array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.asIntArray(): IntArray {\n  return storage\n}\n\n/**\n * Returns a [List] that wraps the original array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic expect fun UIntArray.asList():
List<UInt>\n\n/**\n * Returns a [List] that wraps the original array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic expect fun ULongArray.asList():
List<ULong>\n\n/**\n * Returns a [List] that wraps the original array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic expect fun UByteArray.asList():
List<UByte>\n\n/**\n * Returns a [List] that wraps the original array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic expect fun UShortArray.asList():
List<UShort>\n\n/**\n * Returns an array of type [LongArray], which is a view of this array where each element is
a signed reinterpretation\n * of the corresponding element of this array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.asLongArray(): LongArray {\n  return storage\n}\n\n/**\n * Returns an array of type [ShortArray],
which is a view of this array where each element is a signed reinterpretation\n * of the corresponding element of this
array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.asShortArray(): ShortArray {\n  return storage\n}\n\n/**\n * Returns an array of type [UByteArray],
which is a view of this array where each element is an unsigned reinterpretation\n * of the corresponding element of
this array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline
fun ByteArray.asUByteArray(): UByteArray {\n  return UByteArray(this)\n}\n\n/**\n * Returns an array of type
[UIntArray], which is a view of this array where each element is an unsigned reinterpretation\n * of the
corresponding element of this array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
IntArray.asUIntArray(): UIntArray {\n  return UIntArray(this)\n}\n\n/**\n * Returns an array of type
[ULongArray], which is a view of this array where each element is an unsigned reinterpretation\n * of the
corresponding element of this array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
LongArray.asULongArray(): ULongArray {\n  return ULongArray(this)\n}\n\n/**\n * Returns an array of type

```

```

[UShortArray], which is a view of this array where each element is an unsigned reinterpretation of the
corresponding element of this array.

*^@SinceKotlin("1.3")^@ExperimentalUnsignedTypes^@kotlin.internal.InlineOnly^@public inline fun
ShortArray.asUShortArray(): UShortArray {
    return UShortArray(this)
}

/**
 * Returns `true` if the two
specified arrays are *structurally* equal to one another,
 * i.e. contain the same number of the same elements in the
same order.
 *^@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")^@SinceKotlin("1.3")^@DeprecatedSinceKotlin(hiddenSince =
"1.4")^@ExperimentalUnsignedTypes^@public infix fun UIntArray.contentEquals(other: UIntArray): Boolean {
    return this.contentEquals(other)
}

/**
 * Returns `true` if the two specified arrays are *structurally* equal to
one another,
 * i.e. contain the same number of the same elements in the same order.
 *^@Deprecated("Use
Kotlin compiler 1.4 to avoid deprecation
warning.")^@SinceKotlin("1.3")^@DeprecatedSinceKotlin(hiddenSince =
"1.4")^@ExperimentalUnsignedTypes^@public infix fun ULongArray.contentEquals(other: ULongArray):
Boolean {
    return this.contentEquals(other)
}

/**
 * Returns `true` if the two specified arrays are
*structurally* equal to one another,
 * i.e. contain the same number of the same elements in the same order.
 *^@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")^@SinceKotlin("1.3")^@DeprecatedSinceKotlin(hiddenSince =
"1.4")^@ExperimentalUnsignedTypes^@public infix fun UByteArray.contentEquals(other: UByteArray): Boolean
{
    return this.contentEquals(other)
}

/**
 * Returns `true` if the two specified arrays are *structurally* equal
to one another,
 * i.e. contain the same number of the same elements in the same order.
 *^@Deprecated("Use
Kotlin compiler 1.4 to avoid deprecation
warning.")^@SinceKotlin("1.3")^@DeprecatedSinceKotlin(hiddenSince =
"1.4")^@ExperimentalUnsignedTypes^@public infix fun UShortArray.contentEquals(other: UShortArray):
Boolean {
    return this.contentEquals(other)
}

/**
 * Returns `true` if the two specified arrays are
*structurally* equal to one another,
 * i.e. contain the same number of the same elements in the same order.
 *^@SinceKotlin("1.4")^@ExperimentalUnsignedTypes^@public infix fun
UIntArray?.contentEquals(other:
UIntArray?): Boolean {
    return this?.storage.contentEquals(other?.storage)
}

/**
 * Returns `true` if the two
specified arrays are *structurally* equal to one another,
 * i.e. contain the same number of the same elements in the
same order.
 *^@SinceKotlin("1.4")^@ExperimentalUnsignedTypes^@public infix fun
ULongArray?.contentEquals(other: ULongArray?): Boolean {
    return
this?.storage.contentEquals(other?.storage)
}

/**
 * Returns `true` if the two specified arrays are *structurally*
equal to one another,
 * i.e. contain the same number of the same elements in the same order.
 *^@SinceKotlin("1.4")^@ExperimentalUnsignedTypes^@public infix fun
UByteArray?.contentEquals(other:
UByteArray?): Boolean {
    return this?.storage.contentEquals(other?.storage)
}

/**
 * Returns `true` if the
two specified arrays are *structurally* equal to one another,
 * i.e. contain the same number of the same elements
in the same order.
 *^@SinceKotlin("1.4")^@ExperimentalUnsignedTypes^@public infix fun
UShortArray?.contentEquals(other: UShortArray?): Boolean {
    return
this?.storage.contentEquals(other?.storage)
}

/**
 * Returns a hash code based on the contents of this array as
if it is [List].
 *^@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")^@SinceKotlin("1.3")^@DeprecatedSinceKotlin(hiddenSince =
"1.4")^@ExperimentalUnsignedTypes^@public fun UIntArray.contentHashCode(): Int {
    return
this.contentHashCode()
}

/**
 * Returns a hash code based on the contents of this array as if it is [List].
 *^@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")^@SinceKotlin("1.3")^@DeprecatedSinceKotlin(hiddenSince =
"1.4")^@ExperimentalUnsignedTypes^@public fun ULongArray.contentHashCode(): Int {
    return
this.contentHashCode()
}

/**
 * Returns a hash code based on the contents of this array as if it is [List].
 *^@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")^@SinceKotlin("1.3")^@DeprecatedSinceKotlin(hiddenSince =

```

```

\ "1.4")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.contentHashCode(): Int {\n    return
this.contentHashCode()\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n
*\n*\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
warning.")\n@SinceKotlin("1.3")\n@DeprecatedSinceKotlin(hiddenSince =
\ "1.4")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.contentHashCode(): Int {\n    return
this.contentHashCode()\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n
*\n*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UIntArray?.contentHashCode(): Int {\n
return this?.storage.contentHashCode()\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it
is [List].\n *\n*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun
ULongArray?.contentHashCode(): Int {\n    return this?.storage.contentHashCode()\n}\n\n/**\n * Returns a hash
code based on the contents of this array as if it is [List].\n
*\n*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UByteArray?.contentHashCode(): Int {\n
return this?.storage.contentHashCode()\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it
is [List].\n *\n*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun
UShortArray?.contentHashCode(): Int {\n    return this?.storage.contentHashCode()\n}\n\n/**\n * Returns a string
representation of the contents of the specified array as if it is [List].\n *\n * @sample
samples.collections.Arrays.ContentOperations.contentToString\n *\n*\n@Deprecated("Use Kotlin compiler 1.4 to
avoid deprecation warning.")\n@SinceKotlin("1.3")\n@DeprecatedSinceKotlin(hiddenSince =
\ "1.4")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.contentToString(): String {\n    return
this.contentToString()\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is
[List].\n *\n * @sample samples.collections.Arrays.ContentOperations.contentToString\n *\n*\n@Deprecated("Use
Kotlin compiler 1.4 to avoid deprecation
warning.")\n@SinceKotlin("1.3")\n@DeprecatedSinceKotlin(hiddenSince =
\ "1.4")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.contentToString(): String {\n    return
this.contentToString()\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is
[List].\n *\n * @sample samples.collections.Arrays.ContentOperations.contentToString\n *\n*\n@Deprecated("Use
Kotlin compiler 1.4 to avoid deprecation
warning.")\n@SinceKotlin("1.3")\n@DeprecatedSinceKotlin(hiddenSince =
\ "1.4")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.contentToString(): String {\n    return
this.contentToString()\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is
[List].\n *\n * @sample samples.collections.Arrays.ContentOperations.contentToString\n *\n*\n@Deprecated("Use
Kotlin compiler 1.4 to avoid deprecation
warning.")\n@SinceKotlin("1.3")\n@DeprecatedSinceKotlin(hiddenSince =
\ "1.4")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.contentToString(): String {\n    return
this.contentToString()\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is
[List].\n *\n * @sample samples.collections.Arrays.ContentOperations.contentToString\n
*\n*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UIntArray?.contentToString(): String {\n
return this?.joinToString(", ", "[", "]") ?: "null"\n}\n\n/**\n * Returns a string representation of the contents of
the specified array as if it is [List].\n *\n * @sample
samples.collections.Arrays.ContentOperations.contentToString\n
*\n*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun ULongArray?.contentToString(): String {\n
return this?.joinToString(", ", "[", "]") ?: "null"\n}\n\n/**\n * Returns a string representation of the contents of
the specified array as if it is [List].\n *\n * @sample
samples.collections.Arrays.ContentOperations.contentToString\n
*\n*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UByteArray?.contentToString(): String {\n
return this?.joinToString(", ", "[", "]") ?: "null"\n}\n\n/**\n * Returns a string representation of the contents of
the specified array as if it is [List].\n *\n * @sample
samples.collections.Arrays.ContentOperations.contentToString\n

```

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UShortArray?.contentToString(): String {\n    return this?.joinToString(", ", "[", "]") ?: "null"\n}\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * @return the [destination] array.\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nUIntArray.copyInto(destination: UIntArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size):\nUIntArray {\n    storage.copyInto(destination.storage, destinationOffset, startIndex, endIndex)\n    return\n    destination\n}\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * @return the [destination] array.\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nULongArray.copyInto(destination: ULongArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size):\nULongArray {\n    storage.copyInto(destination.storage, destinationOffset, startIndex, endIndex)\n    return\n    destination\n}\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * @return the [destination] array.\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nUByteArray.copyInto(destination: UByteArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size):\nUByteArray {\n    storage.copyInto(destination.storage, destinationOffset, startIndex, endIndex)\n    return\n    destination\n}\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * @return the [destination] array.\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
```

```

UShortArray.copyInto(destination: UShortArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int =
size): UShortArray {\n    storage.copyInto(destination.storage, destinationOffset, startIndex, endIndex)\n    return
destination\n}\n\n/**\n * Returns new array which is a copy of the original array.\n * \n * @sample
samples.collections.Arrays.CopyOfOperations.copyOfOf\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.copyOfOf(): UIntArray {\n    return UIntArray(storage.copyOfOf())\n}\n\n/**\n * Returns new array which is
a copy of the original array.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.copyOfOf\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.copyOfOf(): ULongArray {\n    return ULongArray(storage.copyOfOf())\n}\n\n/**\n * Returns new array
which is a copy of the original array.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.copyOfOf\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.copyOfOf(): UByteArray {\n    return UByteArray(storage.copyOfOf())\n}\n\n/**\n * Returns new array
which is a copy of the original array.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.copyOfOf\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.copyOfOf(): UShortArray {\n    return UShortArray(storage.copyOfOf())\n}\n\n/**\n * Returns new array
which is a copy of the original array, resized to the given [newSize].\n * The copy is either truncated or padded at
the end with zero values if necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array
is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the
copy array are filled with zero values.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.copyOfOf(newSize: Int): UIntArray {\n    return UIntArray(storage.copyOfOf(newSize))\n}\n\n/**\n *
Returns new array which is a copy of the original array, resized to the given [newSize].\n * The copy is either
truncated or padded at the end with zero values if necessary.\n * \n * - If [newSize] is less than the size of the
original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original
array, the extra elements in the copy array are filled with zero values.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.copyOfOf(newSize: Int): ULongArray {\n    return ULongArray(storage.copyOfOf(newSize))\n}\n\n/**\n *
Returns new array which is a copy of the original array, resized to the given [newSize].\n * The copy is either
truncated or padded at the end with zero values if necessary.\n * \n * - If [newSize] is less than the size of the
original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original
array, the extra elements in the copy array are filled with zero values.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.copyOfOf(newSize: Int): UByteArray {\n    return UByteArray(storage.copyOfOf(newSize))\n}\n\n/**\n *
Returns new array which is a copy of the original array, resized to the given [newSize].\n * The copy is either
truncated or padded at the end with zero values if necessary.\n * \n * - If [newSize] is less than the size of the
original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original
array, the extra elements in the copy array are filled with zero values.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.copyOfOf(newSize: Int): UShortArray {\n    return UShortArray(storage.copyOfOf(newSize))\n}\n\n/**\n *
Returns a new array which is a copy of the specified range of the original array.\n * \n * @param fromIndex the start
of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to copy.\n * \n * @throws
IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n *
*\n@throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.copyOfOfRange(fromIndex: Int, toIndex: Int): UIntArray {\n    return
UIntArray(storage.copyOfOfRange(fromIndex, toIndex))\n}\n\n/**\n * Returns a new array which is a copy of the
specified range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n *
*\n@param toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException if

```

[fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws
 IllegalArgumentException if [fromIndex] is greater than [toIndex].\n

```

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
  ULongArray.copyOfRange(fromIndex: Int, toIndex: Int): ULongArray {\n  return
  ULongArray(storage.copyOfRange(fromIndex, toIndex))\n}\n\n/**\n * Returns a new array which is a copy of the
  specified range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n *
  @param toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException if
  [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws
  IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
  *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
  UByteArray.copyOfRange(fromIndex: Int, toIndex: Int): UByteArray {\n  return
  UByteArray(storage.copyOfRange(fromIndex, toIndex))\n}\n\n/**\n * Returns a new array which is a copy of the
  specified range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n *
  @param toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException if
  [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws
  IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
  *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
  UShortArray.copyOfRange(fromIndex: Int, toIndex: Int): UShortArray {\n  return
  UShortArray(storage.copyOfRange(fromIndex, toIndex))\n}\n\n/**\n * Fills this array or its subrange with the
  specified [element] value.\n * \n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n *
  @param toIndex the end of the range (exclusive) to fill, size of this array by default.\n * \n * @throws
  IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n *
  @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
  *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.fill(element: UInt, fromIndex:
  Int = 0, toIndex: Int = size): Unit {\n  storage.fill(element.toInt(), fromIndex, toIndex)\n}\n\n/**\n * Fills this array
  or its subrange with the specified [element] value.\n * \n * @param fromIndex the start of the range (inclusive) to
  fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by default.\n *
  @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this
  array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
  *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.fill(element: ULong,
  fromIndex: Int = 0, toIndex: Int = size): Unit {\n  storage.fill(element.toLong(), fromIndex, toIndex)\n}\n\n/**\n *
  Fills this array or its subrange with the specified [element] value.\n * \n * @param fromIndex the start of the range
  (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by
  default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than
  the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
  *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.fill(element: UByte,
  fromIndex: Int = 0, toIndex: Int = size): Unit {\n  storage.fill(element.toByte(), fromIndex, toIndex)\n}\n\n/**\n *
  Fills this array or its subrange with the specified [element] value.\n * \n * @param fromIndex the start of the range
  (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by
  default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than
  the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
  *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.fill(element: UShort,
  fromIndex: Int = 0, toIndex: Int = size): Unit {\n  storage.fill(element.toShort(), fromIndex, toIndex)\n}\n\n/**\n *
  Returns the range of valid indices for the array.\n
  *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic inline val UIntArray.indices: IntRange\n  get()
  = storage.indices\n\n/**\n * Returns the range of valid indices for the array.\n
  *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic inline val ULongArray.indices: IntRange\n  get()
  = storage.indices\n\n/**\n * Returns the range of valid indices for the array.\n
  
```

```

*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes\npublic inline val UByteArray.indices: IntRange\n
get() = storage.indices\n\n/**\n * Returns the range of valid indices for the array.\n
*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes\npublic inline val UShortArray.indices: IntRange\n
get() = storage.indices\n\n/**\n * Returns the last valid index for the array.\n
*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes\npublic inline val UIntArray.lastIndex: Int\n
get() = storage.lastIndex\n\n/**\n * Returns the last valid index for the array.\n
*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes\npublic inline val ULongArray.lastIndex: Int\n
get() = storage.lastIndex\n\n/**\n * Returns the last valid index for the array.\n
*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes\npublic inline val UByteArray.lastIndex: Int\n
get() = storage.lastIndex\n\n/**\n * Returns the last valid index for the array.\n
*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes\npublic inline val UShortArray.lastIndex: Int\n
get() = storage.lastIndex\n\n/**\n * Returns an array containing all elements of the original array and then the given
[element].\n
*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline
operator fun UIntArray.plus(element: UInt): UIntArray {\n
return UIntArray(storage +
element.toInt())\n}\n\n/**\n * Returns an array containing all elements of the original array and then the given
[element].\n
*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline
operator fun ULongArray.plus(element: ULong): ULongArray {\n
return ULongArray(storage +
element.toLong())\n}\n\n/**\n * Returns an array containing all elements of the original array and then the given
[element].\n
*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline
operator fun UByteArray.plus(element: UByte): UByteArray {\n
return UByteArray(storage +
element.toByte())\n}\n\n/**\n * Returns an array containing all elements of the original array and then the given
[element].\n
*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline
operator fun UShortArray.plus(element: UShort): UShortArray {\n
return UShortArray(storage +
element.toShort())\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements
of the given [elements] collection.\n
*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes\npublic operator
fun UIntArray.plus(elements: Collection<UInt>): UIntArray {\n
var index = size\n
val result =
storage.copyOf(size + elements.size)\n
for (element in elements) result[index++] = element.toInt()\n
return
UIntArray(result)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements
of the given [elements] collection.\n
*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes\npublic operator
fun ULongArray.plus(elements: Collection<ULong>): ULongArray {\n
var index = size\n
val result =
storage.copyOf(size + elements.size)\n
for (element in elements) result[index++] = element.toLong()\n
return
ULongArray(result)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all
elements of the given [elements] collection.\n
*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes\npublic
operator fun UByteArray.plus(elements: Collection<UByte>): UByteArray {\n
var index = size\n
val result =
storage.copyOf(size + elements.size)\n
for (element in elements) result[index++] = element.toByte()\n
return
UByteArray(result)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements
of the given [elements] collection.\n
*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes\npublic operator
fun UShortArray.plus(elements: Collection<UShort>): UShortArray {\n
var index = size\n
val result =
storage.copyOf(size + elements.size)\n
for (element in elements) result[index++] = element.toShort()\n
return
UShortArray(result)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all
elements of the given [elements] array.\n
*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun
UIntArray.plus(elements: UIntArray): UIntArray {\n
return UIntArray(storage + elements.storage)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n
*^@SinceKotlin("1.3")@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun
ULongArray.plus(elements: ULongArray): ULongArray {\n
return ULongArray(storage +
elements.storage)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements
of the given [elements] array.\n

```

```

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun
UByteArray.plus(elements: UByteArray): UByteArray {\n    return UByteArray(storage +
elements.storage)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements
of the given [elements] array.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun
UShortArray.plus(elements: UShortArray): UShortArray {\n    return UShortArray(storage +
elements.storage)\n}\n\n/**\n * Sorts the array in-place.\n * \n * @sample
samples.collections.Arrays.Sorting.sortArray\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic
fun UIntArray.sort(): Unit {\n    if (size > 1) sortArray(this, 0, size)\n}\n\n/**\n * Sorts the array in-place.\n * \n *
@sample samples.collections.Arrays.Sorting.sortArray\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.sort(): Unit {\n    if (size > 1)
sortArray(this, 0, size)\n}\n\n/**\n * Sorts the array in-place.\n * \n * @sample
samples.collections.Arrays.Sorting.sortArray\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic
fun UByteArray.sort(): Unit {\n    if (size > 1) sortArray(this, 0, size)\n}\n\n/**\n * Sorts the array in-place.\n * \n *
@sample samples.collections.Arrays.Sorting.sortArray\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.sort(): Unit {\n    if (size > 1)
sortArray(this, 0, size)\n}\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex the start of the
range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array
by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater
than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n *
@sample samples.collections.Arrays.Sorting.sortRangeOfArray\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.sort(fromIndex: Int = 0, toIndex:
Int = size): Unit {\n    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    sortArray(this, fromIndex,
toIndex)\n}\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex the start of the range
(inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by
default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than
the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n *
@sample samples.collections.Arrays.Sorting.sortRangeOfArray\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.sort(fromIndex: Int = 0,
toIndex: Int = size): Unit {\n    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    sortArray(this,
fromIndex, toIndex)\n}\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex the start of the
range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array
by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater
than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n *
@sample samples.collections.Arrays.Sorting.sortRangeOfArray\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.sort(fromIndex: Int = 0,
toIndex: Int = size): Unit {\n    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    sortArray(this,
fromIndex, toIndex)\n}\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex the start of the
range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array
by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater
than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n *
@sample samples.collections.Arrays.Sorting.sortRangeOfArray\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.sort(fromIndex: Int = 0,
toIndex: Int = size): Unit {\n    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    sortArray(this,
fromIndex, toIndex)\n}\n\n/**\n * Sorts elements of the array in the specified range in-place.\n * The elements are
sorted descending according to their natural sort order.\n * \n * @param fromIndex the start of the range (inclusive)
to sort.\n * @param toIndex the end of the range (exclusive) to sort.\n * \n * @throws IndexOutOfBoundsException
if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws

```

IllegalArgumentException if [fromIndex] is greater than [toIndex].\n

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.sortDescending(fromIndex: Int, toIndex: Int): Unit {\n    sort(fromIndex, toIndex)\n    reverse(fromIndex, toIndex)\n}\n\n/**\n * Sorts elements of the array in the specified range in-place.\n * The elements are sorted descending according to their natural sort order.\n * \n * @param fromIndex the start of the range (inclusive) to sort.\n * @param toIndex the end of the range (exclusive) to sort.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.sortDescending(fromIndex: Int, toIndex: Int): Unit {\n    sort(fromIndex, toIndex)\n    reverse(fromIndex, toIndex)\n}\n\n/**\n * Sorts elements of the array in the specified range in-place.\n * The elements are sorted descending according to their natural sort order.\n * \n * @param fromIndex the start of the range (inclusive) to sort.\n * @param toIndex the end of the range (exclusive) to sort.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.sortDescending(fromIndex: Int, toIndex: Int): Unit {\n    sort(fromIndex, toIndex)\n    reverse(fromIndex, toIndex)\n}\n\n/**\n * Sorts elements of the array in the specified range in-place.\n * The elements are sorted descending according to their natural sort order.\n * \n * @param fromIndex the start of the range (inclusive) to sort.\n * @param toIndex the end of the range (exclusive) to sort.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.sortDescending(fromIndex: Int, toIndex: Int): Unit {\n    sort(fromIndex, toIndex)\n    reverse(fromIndex, toIndex)\n}\n\n/**\n * Returns an array of type [ByteArray], which is a copy of this array where each element is a signed reinterpretation\n * of the corresponding element of this array.\n */\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.toByteArray(): ByteArray {\n    return storage.copyOf()\n}\n\n/**\n * Returns an array of type [IntArray], which is a copy of this array where each element is a signed reinterpretation\n * of the corresponding element of this array.\n */\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.toIntArray(): IntArray {\n    return storage.copyOf()\n}\n\n/**\n * Returns an array of type [LongArray], which is a copy of this array where each element is a signed reinterpretation\n * of the corresponding element of this array.\n */\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.toLongArray(): LongArray {\n    return storage.copyOf()\n}\n\n/**\n * Returns an array of type [ShortArray], which is a copy of this array where each element is a signed reinterpretation\n * of the corresponding element of this array.\n */\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.toShortArray(): ShortArray {\n    return storage.copyOf()\n}\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive array.\n */\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.toTypedArray(): Array<UInt> {\n    return Array(size) { index -> this[index] }\n}\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive array.\n */\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.toTypedArray(): Array<ULong> {\n    return Array(size) { index -> this[index] }\n}\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive array.\n */\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.toTypedArray(): Array<UByte> {\n    return Array(size) { index -> this[index] }\n}\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive array.\n */\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.toTypedArray(): Array<UShort> {\n    return Array(size) { index -> this[index] }\n}\n\n/**\n * Returns an array of UByte containing all of the elements of this generic array.\n */\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun
```

```

Array<out UByte>.toUByteArray(): UByteArray {
    return UByteArray(size) { index -> this[index] }
}

```

* Returns an array of type [UByteArray], which is a copy of this array where each element is an unsigned reinterpretation of the corresponding element of this array.

```

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun
ByteArray.toUByteArray(): UByteArray {
    return UByteArray(this.copyOf())
}

```

* Returns an array of UInt containing all of the elements of this generic array.

```

@SinceKotlin("1.3")@ExperimentalUnsignedTypes
public fun Array<out UInt>.toUIntArray(): UIntArray
{
    return UIntArray(size) { index -> this[index] }
}

```

* Returns an array of type [UIntArray], which is a copy of this array where each element is an unsigned reinterpretation of the corresponding element of this array.

```

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun
IntArray.toUIntArray(): UIntArray {
    return UIntArray(this.copyOf())
}

```

* Returns an array of ULong containing all of the elements of this generic array.

```

@SinceKotlin("1.3")@ExperimentalUnsignedTypes
public fun Array<out ULong>.toULongArray():
ULongArray {
    return ULongArray(size) { index -> this[index] }
}

```

* Returns an array of type [ULongArray], which is a copy of this array where each element is an unsigned reinterpretation of the corresponding element of this array.

```

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun
LongArray.toULongArray(): ULongArray {
    return ULongArray(this.copyOf())
}

```

* Returns an array of UShort containing all of the elements of this generic array.

```

@SinceKotlin("1.3")@ExperimentalUnsignedTypes
public fun Array<out UShort>.toUShortArray():
UShortArray {
    return UShortArray(size) { index -> this[index] }
}

```

* Returns an array of type [UShortArray], which is a copy of this array where each element is an unsigned reinterpretation of the corresponding element of this array.

```

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun
ShortArray.toUShortArray(): UShortArray {
    return UShortArray(this.copyOf())
}

```

* Returns a [Map] where keys are elements from the given array and values are produced by the [valueSelector] function applied to each element. If any two elements are equal, the last one gets added to the map. The returned map preserves the entry iteration order of the original array.

```

@sample
samples.collections.Collections.Transformations.associateWith

```

```

@SinceKotlin("1.4")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <V>
UIntArray.associateWith(valueSelector: (UInt) -> V): Map<UInt, V> {
    val result = LinkedHashMap<UInt, V>(mapCapacity(size).coerceAtLeast(16))
    return associateWithTo(result, valueSelector)
}

```

* Returns a [Map] where keys are elements from the given array and values are produced by the [valueSelector] function applied to each element. If any two elements are equal, the last one gets added to the map. The returned map preserves the entry iteration order of the original array.

```

@sample
samples.collections.Collections.Transformations.associateWith

```

```

@SinceKotlin("1.4")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <V>
ULongArray.associateWith(valueSelector: (ULong) -> V): Map<ULong, V> {
    val result =
    LinkedHashMap<ULong, V>(mapCapacity(size).coerceAtLeast(16))
    return associateWithTo(result,
    valueSelector)
}

```

* Returns a [Map] where keys are elements from the given array and values are produced by the [valueSelector] function applied to each element. If any two elements are equal, the last one gets added to the map. The returned map preserves the entry iteration order of the original array.

```

@sample
samples.collections.Collections.Transformations.associateWith

```

```

@SinceKotlin("1.4")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <V>
UByteArray.associateWith(valueSelector: (UByte) -> V): Map<UByte, V> {
    val result =
    LinkedHashMap<UByte, V>(mapCapacity(size).coerceAtLeast(16))
    return associateWithTo(result,
    valueSelector)
}

```

* Returns a [Map] where keys are elements from the given array and values are produced by the [valueSelector] function applied to each element. If any two elements are equal, the last one

```

gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n *
@sample samples.collections.Collections.Transformations.associateWith\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <V>
UShortArray.associateWith(valueSelector: (UShort) -> V): Map<UShort, V> {\n    val result =
LinkedHashMap<UShort, V>(mapCapacity(size).coerceAtLeast(16))\n    return associateWithTo(result,
valueSelector)\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs for each
element of the given array,\n * where key is the element itself and value is provided by the [valueSelector] function
applied to that key.\n * \n * If any two elements are equal, the last one overwrites the former value in the map.\n * \n *
@sample samples.collections.Collections.Transformations.associateWithTo\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <V, M :
MutableMap<in UInt, in V>> UIntArray.associateWithTo(destination: M, valueSelector: (UInt) -> V): M {\n    for
(element in this) {\n        destination.put(element, valueSelector(element))\n    }\n    return destination\n}\n\n/**\n *
Populates and returns the [destination] mutable map with key-value pairs for each element of the given array,\n *
where key is the element itself and value is provided by the [valueSelector] function applied to that key.\n * \n * If
any two elements are equal, the last one overwrites the former value in the map.\n * \n * @sample
samples.collections.Collections.Transformations.associateWithTo\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <V, M :
MutableMap<in ULong, in V>> ULongArray.associateWithTo(destination: M, valueSelector: (ULong) -> V): M
{\n    for (element in this) {\n        destination.put(element, valueSelector(element))\n    }\n    return
destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs for each element
of the given array,\n * where key is the element itself and value is provided by the [valueSelector] function applied
to that key.\n * \n * If any two elements are equal, the last one overwrites the former value in the map.\n * \n *
@sample samples.collections.Collections.Transformations.associateWithTo\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <V, M :
MutableMap<in UByte, in V>> UByteArray.associateWithTo(destination: M, valueSelector: (UByte) -> V): M {\n
for (element in this) {\n        destination.put(element, valueSelector(element))\n    }\n    return
destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs for each element
of the given array,\n * where key is the element itself and value is provided by the [valueSelector] function applied
to that key.\n * \n * If any two elements are equal, the last one overwrites the former value in the map.\n * \n *
@sample samples.collections.Collections.Transformations.associateWithTo\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <V, M :
MutableMap<in UShort, in V>> UShortArray.associateWithTo(destination: M, valueSelector: (UShort) -> V): M
{\n    for (element in this) {\n        destination.put(element, valueSelector(element))\n    }\n    return
destination\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being
invoked on each element of original array.\n * \n * @sample
samples.collections.Collections.Transformations.flatMap\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UIntArray.flatMap(transform: (UInt) -> Iterable<R>): List<R> {\n    return flatMapTo(ArrayList<R>(),
transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being
invoked on each element of original array.\n * \n * @sample
samples.collections.Collections.Transformations.flatMap\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
ULongArray.flatMap(transform: (ULong) -> Iterable<R>): List<R> {\n    return flatMapTo(ArrayList<R>(),
transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being
invoked on each element of original array.\n * \n * @sample
samples.collections.Collections.Transformations.flatMap\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UByteArray.flatMap(transform: (UByte) -> Iterable<R>): List<R> {\n    return flatMapTo(ArrayList<R>(),

```

```

transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being
invoked on each element of original array.\n * \n * @sample
samples.collections.Collections.Transformations.flatMap\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UShortArray.flatMap(transform: (UShort) -> Iterable<R>): List<R> {\n    return flatMapTo(ArrayList<R>(),
transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being
invoked on each element\n * and its index in the original array.\n * \n * @sample
samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UIntArray.flatMapIndexed(transform: (index: Int, UInt) -> Iterable<R>): List<R> {\n    return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n *
@sample samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
ULongArray.flatMapIndexed(transform: (index: Int, ULong) -> Iterable<R>): List<R> {\n    return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n *
@sample samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UByteArray.flatMapIndexed(transform: (index: Int, UByte) -> Iterable<R>): List<R> {\n    return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n *
@sample samples.collections.Collections.Transformations.flatMapIndexed\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UShortArray.flatMapIndexed(transform: (index: Int, UShort) -> Iterable<R>): List<R> {\n    return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Appends all elements yielded from results of
[transform] function being invoked on each element\n * and its index in the original array, to the given
[destination].\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :
MutableCollection<in R>> UIntArray.flatMapIndexedTo(destination: C, transform: (index: Int, UInt) ->
Iterable<R>): C {\n    var index = 0\n    for (element in this) {\n        val list = transform(index++, element)\n
destination.addAll(list)\n    }\n    return destination\n}\n\n/**\n * Appends all elements yielded from results of
[transform] function being invoked on each element\n * and its index in the original array, to the given
[destination].\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :
MutableCollection<in R>> ULongArray.flatMapIndexedTo(destination: C, transform: (index: Int, ULong) ->
Iterable<R>): C {\n    var index = 0\n    for (element in this) {\n        val list = transform(index++, element)\n
destination.addAll(list)\n    }\n    return destination\n}\n\n/**\n * Appends all elements yielded from results of
[transform] function being invoked on each element\n * and its index in the original array, to the given
[destination].\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :

```

```

MutableCollection<in R>> UByteArray.flatMapIndexedTo(destination: C, transform: (index: Int, UByte) ->
Iterable<R>): C {\n  var index = 0\n  for (element in this) {\n    val list = transform(index++, element)\n  }\n  destination.addAll(list)\n }\n return destination\n}\n\n/**\n * Appends all elements yielded from results of
[transform] function being invoked on each element\n * and its index in the original array, to the given
[destination].\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :
MutableCollection<in R>> UShortArray.flatMapIndexedTo(destination: C, transform: (index: Int, UShort) ->
Iterable<R>): C {\n  var index = 0\n  for (element in this) {\n    val list = transform(index++, element)\n  }\n  destination.addAll(list)\n }\n return destination\n}\n\n/**\n * Appends all elements yielded from results of
[transform] function being invoked on each element of original array, to the given [destination].\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :
MutableCollection<in R>> UIntArray.flatMapTo(destination: C, transform: (UInt) -> Iterable<R>): C {\n  for
(element in this) {\n    val list = transform(element)\n  }\n  destination.addAll(list)\n }\n return
destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each
element of original array, to the given [destination].\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :
MutableCollection<in R>> ULongArray.flatMapTo(destination: C, transform: (ULong) -> Iterable<R>): C {\n  for
(element in this) {\n    val list = transform(element)\n  }\n  destination.addAll(list)\n }\n return
destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each
element of original array, to the given [destination].\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :
MutableCollection<in R>> UByteArray.flatMapTo(destination: C, transform: (UByte) -> Iterable<R>): C {\n  for
(element in this) {\n    val list = transform(element)\n  }\n  destination.addAll(list)\n }\n return
destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each
element of original array, to the given [destination].\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :
MutableCollection<in R>> UShortArray.flatMapTo(destination: C, transform: (UShort) -> Iterable<R>): C {\n  for
(element in this) {\n    val list = transform(element)\n  }\n  destination.addAll(list)\n }\n return
destination\n}\n\n/**\n * Groups elements of the original array by the key returned by the given [keySelector]
function\n * applied to each element and returns a map where each group key is associated with a list of
corresponding elements.\n * \n * The returned map preserves the entry iteration order of the keys produced from the
original array.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K>
UIntArray.groupBy(keySelector: (UInt) -> K): Map<K, List<UInt>> {\n  return groupByTo(LinkedHashMap<K,
MutableList<UInt>>(), keySelector)\n }\n\n/**\n * Groups elements of the original array by the key returned by the
given [keySelector] function\n * applied to each element and returns a map where each group key is associated with
a list of corresponding elements.\n * \n * The returned map preserves the entry iteration order of the keys produced
from the original array.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K>
ULongArray.groupBy(keySelector: (ULong) -> K): Map<K, List<ULong>> {\n  return
groupByTo(LinkedHashMap<K, MutableList<ULong>>(), keySelector)\n }\n\n/**\n * Groups elements of the
original array by the key returned by the given [keySelector] function\n * applied to each element and returns a map
where each group key is associated with a list of corresponding elements.\n * \n * The returned map preserves the
entry iteration order of the keys produced from the original array.\n * \n * @sample
samples.collections.Collections.Transformations.groupBy\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K>
UByteArray.groupBy(keySelector: (UByte) -> K): Map<K, List<UByte>> {\n  return

```

```

groupByTo(LinkedHashMap<K, MutableList<UByte>>(), keySelector)\n\n/**\n * Groups elements of the
original array by the key returned by the given [keySelector] function\n * applied to each element and returns a map
where each group key is associated with a list of corresponding elements.\n * \n * The returned map preserves the
entry iteration order of the keys produced from the original array.\n * \n * @sample
samples.collections.Collections.Transformations.groupBy\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K>
UShortArray.groupBy(keySelector: (UShort) -> K): Map<K, List<UShort>> {\n    return
groupByTo(LinkedHashMap<K, MutableList<UShort>>(), keySelector)\n\n}\n\n/**\n * Groups values returned by
the [valueTransform] function applied to each element of the original array\n * by the key returned by the given
[keySelector] function applied to the element\n * and returns a map where each group key is associated with a list of
corresponding values.\n * \n * The returned map preserves the entry iteration order of the keys produced from the
original array.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeysAndValues\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, V>
UIntArray.groupBy(keySelector: (UInt) -> K, valueTransform: (UInt) -> V): Map<K, List<V>> {\n    return
groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n\n}\n\n/**\n * Groups values
returned by the [valueTransform] function applied to each element of the original array\n * by the key returned by
the given [keySelector] function applied to the element\n * and returns a map where each group key is associated
with a list of corresponding values.\n * \n * The returned map preserves the entry iteration order of the keys
produced from the original array.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeysAndValues\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, V>
ULongArray.groupBy(keySelector: (ULong) -> K, valueTransform: (ULong) -> V): Map<K, List<V>> {\n    return
groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n\n}\n\n/**\n * Groups values
returned by the [valueTransform] function applied to each element of the original array\n * by the key returned by
the given [keySelector] function applied to the element\n * and returns a map where each group key is associated
with a list of corresponding values.\n * \n * The returned map preserves the entry iteration order of the keys
produced from the original array.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeysAndValues\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, V>
UByteArray.groupBy(keySelector: (UByte) -> K, valueTransform: (UByte) -> V): Map<K, List<V>> {\n    return
groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n\n}\n\n/**\n * Groups values
returned by the [valueTransform] function applied to each element of the original array\n * by the key returned by
the given [keySelector] function applied to the element\n * and returns a map where each group key is associated
with a list of corresponding values.\n * \n * The returned map preserves the entry iteration order of the keys
produced from the original array.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeysAndValues\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, V>
UShortArray.groupBy(keySelector: (UShort) -> K, valueTransform: (UShort) -> V): Map<K, List<V>> {\n    return
groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n\n}\n\n/**\n * Groups elements
of the original array by the key returned by the given [keySelector] function\n * applied to each element and puts to
the [destination] map each group key associated with a list of corresponding elements.\n * \n * @return The
[destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, M :
MutableMap<in K, MutableList<UInt>>> UIntArray.groupByTo(destination: M, keySelector: (UInt) -> K): M {\n
for (element in this) {\n    val key = keySelector(element)\n    val list = destination.getOrPut(key) {
ArrayList<UInt>() }\n    list.add(element)\n } return destination\n}\n\n/**\n * Groups elements of the
original array by the key returned by the given [keySelector] function\n * applied to each element and puts to the
[destination] map each group key associated with a list of corresponding elements.\n * \n * @return The

```

```

[destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, M :
MutableMap<in K, MutableList<ULong>>> ULongArray.groupByTo(destination: M, keySelector: (ULong) -> K):
M {\n for (element in this) {\n val key = keySelector(element)\n val list = destination.getOrPut(key) {
ArrayList<ULong>() }\n list.add(element)\n }\n return destination\n}\n\n/**\n * Groups elements of the
original array by the key returned by the given [keySelector] function\n * applied to each element and puts to the
[destination] map each group key associated with a list of corresponding elements.\n * \n * @return The
[destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, M :
MutableMap<in K, MutableList<UByte>>> UByteArray.groupByTo(destination: M, keySelector: (UByte) -> K):
M {\n for (element in this) {\n val key = keySelector(element)\n val list = destination.getOrPut(key) {
ArrayList<UByte>() }\n list.add(element)\n }\n return destination\n}\n\n/**\n * Groups elements of the
original array by the key returned by the given [keySelector] function\n * applied to each element and puts to the
[destination] map each group key associated with a list of corresponding elements.\n * \n * @return The
[destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, M :
MutableMap<in K, MutableList<UShort>>> UShortArray.groupByTo(destination: M, keySelector: (UShort) -> K):
M {\n for (element in this) {\n val key = keySelector(element)\n val list = destination.getOrPut(key) {
ArrayList<UShort>() }\n list.add(element)\n }\n return destination\n}\n\n/**\n * Groups values returned by
the [valueTransform] function applied to each element of the original array\n * by the key returned by the given
[keySelector] function applied to the element\n * and puts to the [destination] map each group key associated with a
list of corresponding values.\n * \n * @return The [destination] map.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeysAndValues\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, V,
M : MutableMap<in K, MutableList<V>>> UIntArray.groupByTo(destination: M, keySelector: (UInt) -> K,
valueTransform: (UInt) -> V): M {\n for (element in this) {\n val key = keySelector(element)\n val list =
destination.getOrPut(key) { ArrayList<V>() }\n list.add(valueTransform(element))\n }\n return
destination\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the
original array\n * by the key returned by the given [keySelector] function applied to the element\n * and puts to the
[destination] map each group key associated with a list of corresponding values.\n * \n * @return The [destination]
map.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeysAndValues\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, V,
M : MutableMap<in K, MutableList<V>>> ULongArray.groupByTo(destination: M, keySelector: (ULong) -> K,
valueTransform: (ULong) -> V): M {\n for (element in this) {\n val key = keySelector(element)\n val list =
destination.getOrPut(key) { ArrayList<V>() }\n list.add(valueTransform(element))\n }\n return
destination\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the
original array\n * by the key returned by the given [keySelector] function applied to the element\n * and puts to the
[destination] map each group key associated with a list of corresponding values.\n * \n * @return The [destination]
map.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeysAndValues\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, V,
M : MutableMap<in K, MutableList<V>>> UByteArray.groupByTo(destination: M, keySelector: (UByte) -> K,
valueTransform: (UByte) -> V): M {\n for (element in this) {\n val key = keySelector(element)\n val list =
destination.getOrPut(key) { ArrayList<V>() }\n list.add(valueTransform(element))\n }\n return
destination\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the
original array\n * by the key returned by the given [keySelector] function applied to the element\n * and puts to the
[destination] map each group key associated with a list of corresponding values.\n * \n * @return The [destination]
map.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeysAndValues\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, V,

```

```

M : MutableMap<in K, MutableList<V>>> UShortArray.groupByTo(destination: M, keySelector: (UShort) -> K,
valueTransform: (UShort) -> V): M {\n  for (element in this) {\n    val key = keySelector(element)\n    val list
= destination.getOrPut(key) { ArrayList<V>() }\n    list.add(valueTransform(element))\n  }\n  return
destination}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each
element in the original array.\n * \n * @sample samples.collections.Collections.Transformations.map\n
*/\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UIntArray.map(transform: (UInt) -> R): List<R> {\n  return mapTo(ArrayList<R>(size), transform)\n}\n\n/**\n *
Returns a list containing the results of applying the given [transform] function\n * to each element in the original
array.\n * \n * @sample samples.collections.Collections.Transformations.map\n
*/\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UIntArray.map(transform: (UInt) -> R): List<R> {\n  return mapTo(ArrayList<R>(size),
transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each
element in the original array.\n * \n * @sample samples.collections.Collections.Transformations.map\n
*/\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UByteArray.map(transform: (UByte) -> R): List<R> {\n  return mapTo(ArrayList<R>(size),
transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each
element in the original array.\n * \n * @sample samples.collections.Collections.Transformations.map\n
*/\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UShortArray.map(transform: (UShort) -> R): List<R> {\n  return mapTo(ArrayList<R>(size),
transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each
element and its index in the original array.\n * @param [transform] function that takes the index of an element and
the element itself\n * and returns the result of the transform applied to the element.\n
*/\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UIntArray.mapIndexed(transform: (index: Int, UInt) -> R): List<R> {\n  return
mapIndexedTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the
given [transform] function\n * to each element and its index in the original array.\n * @param [transform] function
that takes the index of an element and the element itself\n * and returns the result of the transform applied to the
element.\n */\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline
fun <R> ULongArray.mapIndexed(transform: (index: Int, ULong) -> R): List<R> {\n  return
mapIndexedTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the
given [transform] function\n * to each element and its index in the original array.\n * @param [transform] function
that takes the index of an element and the element itself\n * and returns the result of the transform applied to the
element.\n */\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline
fun <R> UByteArray.mapIndexed(transform: (index: Int, UByte) -> R): List<R> {\n  return
mapIndexedTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the
given [transform] function\n * to each element and its index in the original array.\n * @param [transform] function
that takes the index of an element and the element itself\n * and returns the result of the transform applied to the
element.\n */\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline
fun <R> UShortArray.mapIndexed(transform: (index: Int, UShort) -> R): List<R> {\n  return
mapIndexedTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Applies the given [transform] function to each
element and its index in the original array\n * and appends the results to the given [destination].\n * @param
[transform] function that takes the index of an element and the element itself\n * and returns the result of the
transform applied to the element.\n
*/\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :
MutableCollection<in R>> UIntArray.mapIndexedTo(destination: C, transform: (index: Int, UInt) -> R): C {\n  var
index = 0\n  for (item in this)\n    destination.add(transform(index++, item))\n  return destination}\n\n/**\n *
Applies the given [transform] function to each element and its index in the original array\n * and appends the results
to the given [destination].\n * @param [transform] function that takes the index of an element and the element

```

```

itself\n * and returns the result of the transform applied to the element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :
MutableCollection<in R>> ULongArray.mapIndexedTo(destination: C, transform: (index: Int, ULong) -> R): C {\n
var index = 0\n for (item in this)\n destination.add(transform(index++, item))\n return
destination\n}\n\n/**\n * Applies the given [transform] function to each element and its index in the original array\n
* and appends the results to the given [destination].\n * @param [transform] function that takes the index of an
element and the element itself\n * and returns the result of the transform applied to the element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :
MutableCollection<in R>> UByteArray.mapIndexedTo(destination: C, transform: (index: Int, UByte) -> R): C {\n
var index = 0\n for (item in this)\n destination.add(transform(index++, item))\n return
destination\n}\n\n/**\n * Applies the given [transform] function to each element and its index in the original array\n
* and appends the results to the given [destination].\n * @param [transform] function that takes the index of an
element and the element itself\n * and returns the result of the transform applied to the element.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :
MutableCollection<in R>> UShortArray.mapIndexedTo(destination: C, transform: (index: Int, UShort) -> R): C {\n
var index = 0\n for (item in this)\n destination.add(transform(index++, item))\n return
destination\n}\n\n/**\n * Applies the given [transform] function to each element of the original array\n * and
appends the results to the given [destination].\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :
MutableCollection<in R>> UIntArray.mapTo(destination: C, transform: (UInt) -> R): C {\n for (item in this)\n
destination.add(transform(item))\n return destination\n}\n\n/**\n * Applies the given [transform] function to each
element of the original array\n * and appends the results to the given [destination].\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :
MutableCollection<in R>> ULongArray.mapTo(destination: C, transform: (ULong) -> R): C {\n for (item in
this)\n destination.add(transform(item))\n return destination\n}\n\n/**\n * Applies the given [transform]
function to each element of the original array\n * and appends the results to the given [destination].\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :
MutableCollection<in R>> UByteArray.mapTo(destination: C, transform: (UByte) -> R): C {\n for (item in this)\n
destination.add(transform(item))\n return destination\n}\n\n/**\n * Applies the given [transform] function to
each element of the original array\n * and appends the results to the given [destination].\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :
MutableCollection<in R>> UShortArray.mapTo(destination: C, transform: (UShort) -> R): C {\n for (item in
this)\n destination.add(transform(item))\n return destination\n}\n\n/**\n * Returns a lazy [Iterable] that wraps
each element of the original array\n * into an [IndexedValue] containing the index of that element and the element
itself.\n * @SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.withIndex():
Iterable<IndexedValue<UInt>> {\n return IndexingIterable { iterator() }\n}\n\n/**\n * Returns a lazy [Iterable]
that wraps each element of the original array\n * into an [IndexedValue] containing the index of that element and the
element itself.\n * @SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.withIndex():
Iterable<IndexedValue<ULong>> {\n return IndexingIterable { iterator() }\n}\n\n/**\n * Returns a lazy [Iterable]
that wraps each element of the original array\n * into an [IndexedValue] containing the index of that element and the
element itself.\n * @SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.withIndex():
Iterable<IndexedValue<UByte>> {\n return IndexingIterable { iterator() }\n}\n\n/**\n * Returns a lazy [Iterable]
that wraps each element of the original array\n * into an [IndexedValue] containing the index of that element and the
element itself.\n * @SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.withIndex():
Iterable<IndexedValue<UShort>> {\n return IndexingIterable { iterator() }\n}\n\n/**\n * Returns `true` if all
elements match the given [predicate].\n * @sample samples.collections.Collections.Aggregates.all\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.all(predicate: (UInt) -> Boolean): Boolean {\n for (element in this) if (!predicate(element)) return

```

```

false\n    return true\n}\n\n/**\n * Returns `true` if all elements match the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.all\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.all(predicate: (ULong) -> Boolean): Boolean {\n    for (element in this) if (!predicate(element)) return
false\n    return true\n}\n\n/**\n * Returns `true` if all elements match the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.all\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.all(predicate: (UByte) -> Boolean): Boolean {\n    for (element in this) if (!predicate(element)) return
false\n    return true\n}\n\n/**\n * Returns `true` if all elements match the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.all\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.all(predicate: (UShort) -> Boolean): Boolean {\n    for (element in this) if (!predicate(element)) return
false\n    return true\n}\n\n/**\n * Returns `true` if array has at least one element.\n * \n * @sample
samples.collections.Collections.Aggregates.any\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.any(): Boolean {\n    return storage.any()\n}\n\n/**\n * Returns `true` if array has at least one element.\n
 * \n * @sample samples.collections.Collections.Aggregates.any\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.any(): Boolean {\n    return storage.any()\n}\n\n/**\n * Returns `true` if array has at least one
element.\n * \n * @sample samples.collections.Collections.Aggregates.any\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.any(): Boolean {\n    return storage.any()\n}\n\n/**\n * Returns `true` if array has at least one
element.\n * \n * @sample samples.collections.Collections.Aggregates.any\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.any(): Boolean {\n    return storage.any()\n}\n\n/**\n * Returns `true` if at least one element matches
the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.anyWithPredicate\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.any(predicate: (UInt) -> Boolean): Boolean {\n    for (element in this) if (predicate(element)) return true\n
    return false\n}\n\n/**\n * Returns `true` if at least one element matches the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.anyWithPredicate\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.any(predicate: (ULong) -> Boolean): Boolean {\n    for (element in this) if (predicate(element)) return
true\n    return false\n}\n\n/**\n * Returns `true` if at least one element matches the given [predicate].\n * \n *
@sample samples.collections.Collections.Aggregates.anyWithPredicate\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.any(predicate: (UByte) -> Boolean): Boolean {\n    for (element in this) if (predicate(element)) return
true\n    return false\n}\n\n/**\n * Returns `true` if at least one element matches the given [predicate].\n * \n *
@sample samples.collections.Collections.Aggregates.anyWithPredicate\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.any(predicate: (UShort) -> Boolean): Boolean {\n    for (element in this) if (predicate(element)) return
true\n    return false\n}\n\n/**\n * Returns the number of elements matching the given [predicate].\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.count(predicate: (UInt) -> Boolean): Int {\n    var count = 0\n    for (element in this) if
(predicate(element)) ++count\n    return count\n}\n\n/**\n * Returns the number of elements matching the given
[predicate].\n * \n *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun ULongArray.count(predicate: (ULong) -> Boolean): Int {\n    var count = 0\n    for (element in this) if
(predicate(element)) ++count\n    return count\n}\n\n/**\n * Returns the number of elements matching the given
[predicate].\n * \n *\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic

```

```

inline fun UByteArray.count(predicate: (UByte) -> Boolean): Int {
    var count = 0
    for (element in this) if (predicate(element)) ++count
    return count
}

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly@public
inline fun UShortArray.count(predicate: (UShort) -> Boolean): Int {
    var count = 0
    for (element in this) if (predicate(element)) ++count
    return count
}

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly@public
inline fun <R> UIntArray.fold(initial: R, operation: (acc: R, UInt) -> R): R {
    var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator
}

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly@public
inline fun <R> ULongArray.fold(initial: R, operation: (acc: R, ULong) -> R): R {
    var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator
}

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly@public
inline fun <R> UByteArray.fold(initial: R, operation: (acc: R, UByte) -> R): R {
    var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator
}

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly@public
inline fun <R> UShortArray.fold(initial: R, operation: (acc: R, UShort) -> R): R {
    var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator
}

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly@public
inline fun <R> UIntArray.foldIndexed(initial: R, operation: (index: Int, acc: R, UInt) -> R): R {
    var index = 0
    var accumulator = initial
    for (element in this) accumulator = operation(index++, accumulator, element)
    return accumulator
}

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly@public
inline fun <R> ULongArray.foldIndexed(initial: R, operation: (index: Int, acc: R, ULong) -> R): R {
    var index = 0
    var accumulator = initial
    for (element in this) accumulator = operation(index++, accumulator, element)
    return accumulator
}

```

```

ByteArray.foldIndexed(initial: R, operation: (index: Int, acc: R, UByte) -> R): R {
    var index = 0
    var accumulator = initial
    for (element in this) accumulator = operation(index++, accumulator, element)
    return accumulator
}

/** Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element with its index in the original array. Returns the specified [initial] value if the array is empty.
 * @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.
 */
@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <R>
UShortArray.foldIndexed(initial: R, operation: (index: Int, acc: R, UShort) -> R): R {
    var index = 0
    var accumulator = initial
    for (element in this) accumulator = operation(index++, accumulator, element)
    return accumulator
}

/** Accumulates value starting with [initial] value and applying [operation] from right to left to each element and current accumulator value. Returns the specified [initial] value if the array is empty.
 * @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value.
 */
@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <R>
UIntArray.foldRight(initial: R, operation: (UInt, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}

/** Accumulates value starting with [initial] value and applying [operation] from right to left to each element and current accumulator value. Returns the specified [initial] value if the array is empty.
 * @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value.
 */
@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <R>
ULongArray.foldRight(initial: R, operation: (ULong, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}

/** Accumulates value starting with [initial] value and applying [operation] from right to left to each element and current accumulator value. Returns the specified [initial] value if the array is empty.
 * @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value.
 */
@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <R>
UByteArray.foldRight(initial: R, operation: (UByte, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}

/** Accumulates value starting with [initial] value and applying [operation] from right to left to each element with its index in the original array and current accumulator value. Returns the specified [initial] value if the array is empty.
 * @param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.
 */
@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <R>
UShortArray.foldRightIndexed(initial: R, operation: (index: Int, UShort, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

/** Accumulates value starting with [initial] value and applying [operation] from right to left to each element with its index in the original array and current accumulator value. Returns the specified [initial] value if the array is empty.
 * @param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator
 */

```

value.\n *\n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline fun
<R> ULongArray.foldRightIndexed(initial: R, operation: (index: Int, ULong, acc: R) -> R): R {\n var index =
lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator = operation(index, get(index),
accumulator)\n --index\n }\n return accumulator\n }\n\n/**\n * Accumulates value starting with [initial]
value and applying [operation] from right to left\n * to each element with its index in the original array and current
accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation]
function that takes the index of an element, the element itself\n * and current accumulator value, and calculates the
next accumulator value.\n

*\n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline fun <R>
UByteArray.foldRightIndexed(initial: R, operation: (index: Int, UByte, acc: R) -> R): R {\n var index =
lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator = operation(index, get(index),
accumulator)\n --index\n }\n return accumulator\n }\n\n/**\n * Accumulates value starting with [initial]
value and applying [operation] from right to left\n * to each element with its index in the original array and current
accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation]
function that takes the index of an element, the element itself\n * and current accumulator value, and calculates the
next accumulator value.\n

*\n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline fun <R>
UShortArray.foldRightIndexed(initial: R, operation: (index: Int, UShort, acc: R) -> R): R {\n var index =
lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator = operation(index, get(index),
accumulator)\n --index\n }\n return accumulator\n }\n\n/**\n * Performs the given [action] on each
element.\n *\n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline
fun UIntArray.forEach(action: (UInt) -> Unit): Unit {\n for (element in this) action(element)\n }\n\n/**\n * Performs the given [action] on each element.\n

*\n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline fun
ULongArray.forEach(action: (ULong) -> Unit): Unit {\n for (element in this) action(element)\n }\n\n/**\n * Performs the given [action] on each element.\n

*\n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline fun
UByteArray.forEach(action: (UByte) -> Unit): Unit {\n for (element in this) action(element)\n }\n\n/**\n * Performs the given [action] on each element.\n

*\n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline fun
UShortArray.forEach(action: (UShort) -> Unit): Unit {\n for (element in this) action(element)\n }\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element.\n * \n * @param [action]
function that takes the index of an element and the element itself\n * and performs the action on the element.\n

*\n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline fun
UIntArray.forEachIndexed(action: (index: Int, UInt) -> Unit): Unit {\n var index = 0\n for (item in this)
action(index++, item)\n }\n\n/**\n * Performs the given [action] on each element, providing sequential index with
the element.\n * \n * @param [action] function that takes the index of an element and the element itself\n * and performs
the action on the element.\n

*\n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline fun
ULongArray.forEachIndexed(action: (index: Int, ULong) -> Unit): Unit {\n var index = 0\n for (item in this)
action(index++, item)\n }\n\n/**\n * Performs the given [action] on each element, providing sequential index with
the element.\n * \n * @param [action] function that takes the index of an element and the element itself\n * and performs
the action on the element.\n

*\n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline fun
UByteArray.forEachIndexed(action: (index: Int, UByte) -> Unit): Unit {\n var index = 0\n for (item in this)
action(index++, item)\n }\n\n/**\n * Performs the given [action] on each element, providing sequential index with
the element.\n * \n * @param [action] function that takes the index of an element and the element itself\n * and performs
the action on the element.\n

```

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.forEachIndexed(action: (index: Int, UShort) -> Unit): Unit {\n    var index = 0\n    for (item in this)
action(index++, item)\n}\n\n@Deprecated("Use maxOrNull instead.\",
ReplaceWith("this.maxOrNull()")\n)\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5",
hiddenSince = "1.6")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.max():
UInt? {\n    return maxOrNull()\n}\n\n@Deprecated("Use maxOrNull instead.\",
ReplaceWith("this.maxOrNull()")\n)\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5",
hiddenSince = "1.6")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.max():
ULong? {\n    return maxOrNull()\n}\n\n@Deprecated("Use maxOrNull instead.\",
ReplaceWith("this.maxOrNull()")\n)\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5",
hiddenSince = "1.6")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.max():
UByte? {\n    return maxOrNull()\n}\n\n@Deprecated("Use maxOrNull instead.\",
ReplaceWith("this.maxOrNull()")\n)\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5",
hiddenSince = "1.6")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.max():
UShort? {\n    return maxOrNull()\n}\n\n@Deprecated("Use maxByOrNull instead.\",
ReplaceWith("this.maxByOrNull(selector)")\n)\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince =
"1.5", hiddenSince =
"1.6")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
<R : Comparable<R>> UIntArray.maxBy(selector: (UInt) -> R): UInt? {\n    return
maxByOrNull(selector)\n}\n\n@Deprecated("Use maxByOrNull instead.\",
ReplaceWith("this.maxByOrNull(selector)")\n)\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince =
"1.5", hiddenSince =
"1.6")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
<R : Comparable<R>> ULongArray.maxBy(selector: (ULong) -> R): ULong? {\n    return
maxByOrNull(selector)\n}\n\n@Deprecated("Use maxByOrNull instead.\",
ReplaceWith("this.maxByOrNull(selector)")\n)\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince =
"1.5", hiddenSince =
"1.6")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
<R : Comparable<R>> UByteArray.maxBy(selector: (UByte) -> R): UByte? {\n    return
maxByOrNull(selector)\n}\n\n@Deprecated("Use maxByOrNull instead.\",
ReplaceWith("this.maxByOrNull(selector)")\n)\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince =
"1.5", hiddenSince =
"1.6")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
<R : Comparable<R>> UShortArray.maxBy(selector: (UShort) -> R): UShort? {\n    return
maxByOrNull(selector)\n}\n\n/**\n * Returns the first element yielding the largest value of the given function or
`null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.maxByOrNull\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :
Comparable<R>> UIntArray.maxByOrNull(selector: (UInt) -> R): UInt? {\n    if (isEmpty()) return null\n    var
maxElem = this[0]\n    val lastIndex = this.lastIndex\n    if (lastIndex == 0) return maxElem\n    var maxValue =
selector(maxElem)\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        val v = selector(e)\n        if (maxValue < v)
{\n            maxElem = e\n            maxValue = v\n        }\n    }\n    return maxElem\n}\n\n/**\n * Returns the first
element yielding the largest value of the given function or `null` if there are no elements.\n * \n * @sample
samples.collections.Collections.Aggregates.maxByOrNull\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :
Comparable<R>> ULongArray.maxByOrNull(selector: (ULong) -> R): ULong? {\n    if (isEmpty()) return null\n    var
maxElem = this[0]\n    val lastIndex = this.lastIndex\n    if (lastIndex == 0) return maxElem\n    var maxValue =
selector(maxElem)\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        val v = selector(e)\n        if (maxValue < v)
{\n            maxElem = e\n            maxValue = v\n        }\n    }\n    return maxElem\n}\n\n/**\n * Returns the first

```

```

element yielding the largest value of the given function or `null` if there are no elements.
samples.collections.Collections.Aggregates.maxByOrNull

*\/n@SinceKotlin("1.4")n@ExperimentalUnsignedTypesn@kotlin.internal.InlineOnlynpublic inline fun <R :
Comparable<R>> UByteArray.maxByOrNull(selector: (UByte) -> R): UByte? {n if (isEmpty()) return nulln
var maxElem = this[0]n val lastIndex = this.lastIndexn if (lastIndex == 0) return maxElemn var maxValue =
selector(maxElem)n for (i in 1..lastIndex) {n val e = this[i]n val v = selector(e)n if (maxValue < v)
{n maxElem = e\n maxValue = v\n }n }n return maxElemn}n/n/**n * Returns the first
element yielding the largest value of the given function or `null` if there are no elements.
samples.collections.Collections.Aggregates.maxByOrNull

*\/n@SinceKotlin("1.4")n@ExperimentalUnsignedTypesn@kotlin.internal.InlineOnlynpublic inline fun <R :
Comparable<R>> UShortArray.maxByOrNull(selector: (UShort) -> R): UShort? {n if (isEmpty()) return nulln
var maxElem = this[0]n val lastIndex = this.lastIndexn if (lastIndex == 0) return maxElemn var maxValue =
selector(maxElem)n for (i in 1..lastIndex) {n val e = this[i]n val v = selector(e)n if (maxValue < v)
{n maxElem = e\n maxValue = v\n }n }n return maxElemn}n/n/**n * Returns the largest
value among all values produced by [selector] functionn * applied to each element in the array.n * If any of
values produced by [selector] function is `NaN`, the returned result is `NaN`.n * @throws
NoSuchElementException if the array is empty.

*\/n@SinceKotlin("1.4")n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)n@OverloadResolution
ByLambdaReturnTypen@ExperimentalUnsignedTypesn@kotlin.internal.InlineOnlynpublic inline fun
UIntArray.maxOf(selector: (UInt) -> Double): Double {n if (isEmpty()) throw NoSuchElementException()n
var maxValue = selector(this[0])n for (i in 1..lastIndex) {n val v = selector(this[i])n maxValue =
maxOf(maxValue, v)n }n return maxValue}n/n/**n * Returns the largest value among all values produced
by [selector] functionn * applied to each element in the array.n * If any of values produced by [selector]
function is `NaN`, the returned result is `NaN`.n * @throws NoSuchElementException if the array is empty.

*\/n@SinceKotlin("1.4")n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)n@OverloadResolution
ByLambdaReturnTypen@ExperimentalUnsignedTypesn@kotlin.internal.InlineOnlynpublic inline fun
ULongArray.maxOf(selector: (ULong) -> Double): Double {n if (isEmpty()) throw NoSuchElementException()n
var maxValue = selector(this[0])n for (i in 1..lastIndex) {n val v = selector(this[i])n maxValue =
maxOf(maxValue, v)n }n return maxValue}n/n/**n * Returns the largest value among all values produced
by [selector] functionn * applied to each element in the array.n * If any of values produced by [selector]
function is `NaN`, the returned result is `NaN`.n * @throws NoSuchElementException if the array is empty.

*\/n@SinceKotlin("1.4")n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)n@OverloadResolution
ByLambdaReturnTypen@ExperimentalUnsignedTypesn@kotlin.internal.InlineOnlynpublic inline fun
UByteArray.maxOf(selector: (UByte) -> Double): Double {n if (isEmpty()) throw NoSuchElementException()n
var maxValue = selector(this[0])n for (i in 1..lastIndex) {n val v = selector(this[i])n maxValue =
maxOf(maxValue, v)n }n return maxValue}n/n/**n * Returns the largest value among all values produced
by [selector] functionn * applied to each element in the array.n * If any of values produced by [selector]
function is `NaN`, the returned result is `NaN`.n * @throws NoSuchElementException if the array is empty.

*\/n@SinceKotlin("1.4")n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)n@OverloadResolution
ByLambdaReturnTypen@ExperimentalUnsignedTypesn@kotlin.internal.InlineOnlynpublic inline fun
UShortArray.maxOf(selector: (UShort) -> Double): Double {n if (isEmpty()) throw
NoSuchElementException()n var maxValue = selector(this[0])n for (i in 1..lastIndex) {n val v =
selector(this[i])n maxValue = maxOf(maxValue, v)n }n return maxValue}n/n/**n * Returns the
largest value among all values produced by [selector] functionn * applied to each element in the array.n * If
any of values produced by [selector] function is `NaN`, the returned result is `NaN`.n * @throws
NoSuchElementException if the array is empty.

*\/n@SinceKotlin("1.4")n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)n@OverloadResolution
ByLambdaReturnTypen@ExperimentalUnsignedTypesn@kotlin.internal.InlineOnlynpublic inline fun

```

```

UIntArray.maxOf(selector: (UInt) -> Float): Float {
    if (isEmpty()) throw NoSuchElementException()
    var maxV = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxV = maxOf(maxV, v)
    }
    return maxV
}
Returns the largest value among all values produced
by [selector] function
applied to each element in the array.
If any of values produced by [selector]
function is NaN, the returned result is NaN.
@throws NoSuchElementException if the array is empty.
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
ULongArray.maxOf(selector: (ULong) -> Float): Float {
    if (isEmpty()) throw NoSuchElementException()
    var maxV = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxV = maxOf(maxV, v)
    }
    return maxV
}
Returns the largest value among all values produced
by [selector] function
applied to each element in the array.
If any of values produced by [selector]
function is NaN, the returned result is NaN.
@throws NoSuchElementException if the array is empty.
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UByteArray.maxOf(selector: (UByte) -> Float): Float {
    if (isEmpty()) throw NoSuchElementException()
    var maxV = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxV = maxOf(maxV, v)
    }
    return maxV
}
Returns the largest value among all values produced
by [selector] function
applied to each element in the array.
If any of values produced by [selector]
function is NaN, the returned result is NaN.
@throws NoSuchElementException if the array is empty.
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UShortArray.maxOf(selector: (UShort) -> Float): Float {
    if (isEmpty()) throw NoSuchElementException()
    var maxV = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxV = maxOf(maxV, v)
    }
    return maxV
}
Returns the largest value among all values produced
by [selector] function
applied to each element in the array.
@throws NoSuchElementException if the
array is empty.
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R :
Comparable<R>> UIntArray.maxOf(selector: (UInt) -> R): R {
    if (isEmpty()) throw
NoSuchElementException()
    var maxV = selector(this[0])
    for (i in 1..lastIndex) {
        val v =
selector(this[i])
        if (maxV < v) {
            maxV = v
        }
    }
    return maxV
}
Returns the largest value among all values produced by [selector] function
applied to each element in the
array.
@throws NoSuchElementException if the array is empty.
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R :
Comparable<R>> ULongArray.maxOf(selector: (ULong) -> R): R {
    if (isEmpty()) throw
NoSuchElementException()
    var maxV = selector(this[0])
    for (i in 1..lastIndex) {
        val v =
selector(this[i])
        if (maxV < v) {
            maxV = v
        }
    }
    return maxV
}
Returns the largest value among all values produced by [selector] function
applied to each element in the
array.
@throws NoSuchElementException if the array is empty.
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R :
Comparable<R>> UByteArray.maxOf(selector: (UByte) -> R): R {
    if (isEmpty()) throw
NoSuchElementException()
    var maxV = selector(this[0])
    for (i in 1..lastIndex) {
        val v =
selector(this[i])
        if (maxV < v) {
            maxV = v
        }
    }
    return maxV
}
Returns the largest value among all values produced by [selector] function
applied to each element in the
array.
@throws NoSuchElementException if the array is empty.
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution

```

```

ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :
Comparable<R>> UShortArray.maxOf(selector: (UShort) -> R): R {\n  if (isEmpty()) throw
NoSuchElementException()\n  var maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v =
selector(this[i])\n    if (maxValue < v) {\n      maxValue = v\n    }\n  }\n  return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned
result is `NaN`.\n
*/\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.maxOfOrNull(selector: (UInt) -> Double): Double? {\n  if (isEmpty()) return null\n  var maxValue =
selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    maxValue = maxOf(maxValue, v)\n  }\n  return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector]
function is `NaN`, the returned result is `NaN`.\n
*/\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.maxOfOrNull(selector: (ULong) -> Double): Double? {\n  if (isEmpty()) return null\n  var
maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    maxValue =
maxOf(maxValue, v)\n  }\n  return maxValue\n}\n\n/**\n * Returns the largest value among all values produced
by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of
values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*/\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.maxOfOrNull(selector: (UByte) -> Double): Double? {\n  if (isEmpty()) return null\n  var maxValue
= selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    maxValue = maxOf(maxValue,
v)\n  }\n  return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector]
function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced
by [selector] function is `NaN`, the returned result is `NaN`.\n
*/\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.maxOfOrNull(selector: (UShort) -> Double): Double? {\n  if (isEmpty()) return null\n  var
maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    maxValue =
maxOf(maxValue, v)\n  }\n  return maxValue\n}\n\n/**\n * Returns the largest value among all values produced
by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of
values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*/\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.maxOfOrNull(selector: (UInt) -> Float): Float? {\n  if (isEmpty()) return null\n  var maxValue =
selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    maxValue = maxOf(maxValue, v)\n  }\n  return
maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector]
function is `NaN`, the returned result is `NaN`.\n
*/\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.maxOfOrNull(selector: (ULong) -> Float): Float? {\n  if (isEmpty()) return null\n  var maxValue =
selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    maxValue = maxOf(maxValue, v)\n  }\n  return
maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector]
function is `NaN`, the returned result is `NaN`.\n
*/

```

function is `NaN`, the returned result is `NaN`.\n

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nUByteArray.maxOfOrNull(selector: (UByte) -> Float): Float? {\n    if (isEmpty()) return null\n    var maxValue =\n    selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector]\n function is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nUShortArray.maxOfOrNull(selector: (UShort) -> Float): Float? {\n    if (isEmpty()) return null\n    var maxValue =\n    selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>\nUIntArray.maxOfOrNull(selector: (UInt) -> R): R? {\n    if (isEmpty()) return null\n    var\n    maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (maxValue < v) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values\n produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>\nULongArray.maxOfOrNull(selector: (ULong) -> R): R? {\n    if (isEmpty()) return null\n    var\n    maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (maxValue < v) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values\n produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>\nUByteArray.maxOfOrNull(selector: (UByte) -> R): R? {\n    if (isEmpty()) return null\n    var\n    maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (maxValue < v) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values\n produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>\nUShortArray.maxOfOrNull(selector: (UShort) -> R): R? {\n    if (isEmpty()) return null\n    var\n    maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (maxValue < v) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value according to the\n provided [comparator]\n * among all values produced by [selector] function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>\nUIntArray.maxOfWith(comparator: Comparator<in R>, selector: (UInt) -> R): R {\n    if (isEmpty()) throw\n    NoSuchElementException()\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v =\n        selector(this[i])\n        if (comparator.compare(maxValue, v) < 0) {\n            maxValue = v\n        }\n    }\n    return\n    maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values\n produced by [selector] function applied to each element in the array.\n * \n * @throws NoSuchElementException if\n the array is empty.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
```

```

ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
ULongArray.maxOfWith(comparator: Comparator<in R>, selector: (ULong) -> R): R {\n  if (isEmpty()) throw
NoSuchElementException()\n  var maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v =
selector(this[i])\n    if (comparator.compare(maxValue, v) < 0) {\n      maxValue = v\n    }\n  }\n  return
maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values
produced by [selector] function applied to each element in the array.\n * \n * @throws NoSuchElementException if
the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UByteArray.maxOfWith(comparator: Comparator<in R>, selector: (UByte) -> R): R {\n  if (isEmpty()) throw
NoSuchElementException()\n  var maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v =
selector(this[i])\n    if (comparator.compare(maxValue, v) < 0) {\n      maxValue = v\n    }\n  }\n  return
maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values
produced by [selector] function applied to each element in the array.\n * \n * @throws NoSuchElementException if
the array is empty.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UShortArray.maxOfWith(comparator: Comparator<in R>, selector: (UShort) -> R): R {\n  if (isEmpty()) throw
NoSuchElementException()\n  var maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v =
selector(this[i])\n    if (comparator.compare(maxValue, v) < 0) {\n      maxValue = v\n    }\n  }\n  return
maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values
produced by [selector] function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UIntArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (UInt) -> R): R? {\n  if (isEmpty()) return
null\n  var maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if
(comparator.compare(maxValue, v) < 0) {\n      maxValue = v\n    }\n  }\n  return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
ULongArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (ULong) -> R): R? {\n  if (isEmpty())
return null\n  var maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if
(comparator.compare(maxValue, v) < 0) {\n      maxValue = v\n    }\n  }\n  return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UByteArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (UByte) -> R): R? {\n  if (isEmpty())
return null\n  var maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if
(comparator.compare(maxValue, v) < 0) {\n      maxValue = v\n    }\n  }\n  return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array or `null` if there are no elements.\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UShortArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (UShort) -> R): R? {\n  if (isEmpty())
return null\n  var maxValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    if
(comparator.compare(maxValue, v) < 0) {\n      maxValue = v\n    }\n  }\n  return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array or `null` if there are no elements.\n

```

Returns the largest element or `null` if there are no elements.

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.maxOrNull(): UInt? {\n    if\n    (isEmpty()) return null\n    var max = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (max < e) max\n        = e\n    }\n    return max\n}\n\n/**\n * Returns the largest element or `null` if there are no elements.\n */\n*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.maxOrNull(): ULong? {\n    if\n    (isEmpty()) return null\n    var max = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (max < e) max\n        = e\n    }\n    return max\n}\n\n/**\n * Returns the largest element or `null` if there are no elements.\n */\n*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.maxOrNull(): UByte? {\n    if\n    (isEmpty()) return null\n    var max = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (max < e) max\n        = e\n    }\n    return max\n}\n\n/**\n * Returns the largest element or `null` if there are no elements.\n */\n*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.maxOrNull(): UShort? {\n    if\n    (isEmpty()) return null\n    var max = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (max < e) max\n        = e\n    }\n    return max\n}\n\n@Deprecated("Use maxWithOrNull instead.")\nReplaceWith("this.maxWithOrNull(comparator)")\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince\n= "1.5", hiddenSince = "1.6")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun\nUIntArray.maxWith(comparator: Comparator<in UInt>): UInt? {\n    return\n    maxWithOrNull(comparator)\n}\n\n@Deprecated("Use maxWithOrNull instead.")\nReplaceWith("this.maxWithOrNull(comparator)")\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince\n= "1.5", hiddenSince = "1.6")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun\nULongArray.maxWith(comparator: Comparator<in ULong>): ULong? {\n    return\n    maxWithOrNull(comparator)\n}\n\n@Deprecated("Use maxWithOrNull instead.")\nReplaceWith("this.maxWithOrNull(comparator)")\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince\n= "1.5", hiddenSince = "1.6")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun\nUByteArray.maxWith(comparator: Comparator<in UByte>): UByte? {\n    return\n    maxWithOrNull(comparator)\n}\n\n@Deprecated("Use maxWithOrNull instead.")\nReplaceWith("this.maxWithOrNull(comparator)")\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince\n= "1.5", hiddenSince = "1.6")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun\nUShortArray.maxWith(comparator: Comparator<in UShort>): UShort? {\n    return\n    maxWithOrNull(comparator)\n}\n\n/**\n * Returns the first element having the largest value according to the\n * provided [comparator] or `null` if there are no elements.\n */\n*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.maxWithOrNull(comparator:\nComparator<in UInt>): UInt? {\n    if (isEmpty()) return null\n    var max = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (comparator.compare(max, e) < 0) max = e\n    }\n    return max\n}\n\n/**\n * Returns the\n * first element having the largest value according to the provided [comparator] or `null` if there are no elements.\n */\n*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.maxWithOrNull(comparator:\nComparator<in ULong>): ULong? {\n    if (isEmpty()) return null\n    var max = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (comparator.compare(max, e) < 0) max = e\n    }\n    return max\n}\n\n/**\n * Returns the\n * first element having the largest value according to the provided [comparator] or `null` if there are no elements.\n */\n*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.maxWithOrNull(comparator:\nComparator<in UByte>): UByte? {\n    if (isEmpty()) return null\n    var max = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (comparator.compare(max, e) < 0) max = e\n    }\n    return max\n}\n\n/**\n * Returns the\n * first element having the largest value according to the provided [comparator] or `null` if there are no elements.\n */\n*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.maxWithOrNull(comparator:\nComparator<in UShort>): UShort? {\n    if (isEmpty()) return null\n    var max = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (comparator.compare(max, e) < 0) max = e\n    }\n    return\n    max\n}\n\n@Deprecated("Use minOrNull instead.")\nReplaceWith("this.minOrNull()")\n@DeprecatedSinceKotlin(warningSince = "1.4", errorSince = "1.5",\nhiddenSince = "1.6")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.min(): UInt?
```

```

{\n  return minOrNull()\n}\n\n@Deprecated("Use minOrNull instead.\",
ReplaceWith("this.minOrNull()"))\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince = \"1.5\",
hiddenSince = \"1.6\")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.min():
ULong? {\n  return minOrNull()\n}\n\n@Deprecated("Use minOrNull instead.\",
ReplaceWith("this.minOrNull()"))\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince = \"1.5\",
hiddenSince = \"1.6\")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.min():
UByte? {\n  return minOrNull()\n}\n\n@Deprecated("Use minOrNull instead.\",
ReplaceWith("this.minOrNull()"))\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince = \"1.5\",
hiddenSince = \"1.6\")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.min():
UShort? {\n  return minOrNull()\n}\n\n@Deprecated("Use minByOrNull instead.\",
ReplaceWith("this.minByOrNull(selector)"))\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince =
\"1.5\", hiddenSince =
\"1.6\")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
<R : Comparable<R>> UIntArray.minBy(selector: (UInt) -> R): UInt? {\n  return
minByOrNull(selector)\n}\n\n@Deprecated("Use minByOrNull instead.\",
ReplaceWith("this.minByOrNull(selector)"))\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince =
\"1.5\", hiddenSince =
\"1.6\")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
<R : Comparable<R>> ULongArray.minBy(selector: (ULong) -> R): ULong? {\n  return
minByOrNull(selector)\n}\n\n@Deprecated("Use minByOrNull instead.\",
ReplaceWith("this.minByOrNull(selector)"))\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince =
\"1.5\", hiddenSince =
\"1.6\")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
<R : Comparable<R>> UByteArray.minBy(selector: (UByte) -> R): UByte? {\n  return
minByOrNull(selector)\n}\n\n@Deprecated("Use minByOrNull instead.\",
ReplaceWith("this.minByOrNull(selector)"))\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince =
\"1.5\", hiddenSince =
\"1.6\")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
<R : Comparable<R>> UShortArray.minBy(selector: (UShort) -> R): UShort? {\n  return
minByOrNull(selector)\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function or
`null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.minByOrNull\n
*\n*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :
Comparable<R>> UIntArray.minByOrNull(selector: (UInt) -> R): UInt? {\n  if (isEmpty()) return null\n  var
minElem = this[0]\n  val lastIndex = this.lastIndex\n  if (lastIndex == 0) return minElem\n  var minValue =
selector(minElem)\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    val v = selector(e)\n    if (minValue > v)
{\n      minElem = e\n      minValue = v\n    }\n  }\n  return minElem\n}\n\n/**\n * Returns the first
element yielding the smallest value of the given function or `null` if there are no elements.\n * \n * @sample
samples.collections.Collections.Aggregates.minByOrNull\n
*\n*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :
Comparable<R>> ULongArray.minByOrNull(selector: (ULong) -> R): ULong? {\n  if (isEmpty()) return null\n
var minElem = this[0]\n  val lastIndex = this.lastIndex\n  if (lastIndex == 0) return minElem\n  var minValue =
selector(minElem)\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    val v = selector(e)\n    if (minValue > v)
{\n      minElem = e\n      minValue = v\n    }\n  }\n  return minElem\n}\n\n/**\n * Returns the first
element yielding the smallest value of the given function or `null` if there are no elements.\n * \n * @sample
samples.collections.Collections.Aggregates.minByOrNull\n
*\n*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :
Comparable<R>> UByteArray.minByOrNull(selector: (UByte) -> R): UByte? {\n  if (isEmpty()) return null\n
var minElem = this[0]\n  val lastIndex = this.lastIndex\n  if (lastIndex == 0) return minElem\n  var minValue =

```

```

selector(minElem)\n for (i in 1..lastIndex) {\n     val e = this[i]\n     val v = selector(e)\n     if (minValue > v)\n {\n     minElem = e\n     minValue = v\n     }\n }\n return minElem\n}\n\n/**\n * Returns the first\n element yielding the smallest value of the given function or `null` if there are no elements.\n * \n * @sample\n samples.collections.Collections.Aggregates.minByOrNull\n\n *\n @SinceKotlin("1.4")\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline fun <R : Comparable<R>> UShortArray.minByOrNull(selector: (UShort) -> R): UShort? {\n     if (isEmpty()) return null\n     var minElem = this[0]\n     val lastIndex = this.lastIndex\n     if (lastIndex == 0) return minElem\n     var minValue = selector(minElem)\n     for (i in 1..lastIndex) {\n         val e = this[i]\n         val v = selector(e)\n         if (minValue > v)\n {\n         minElem = e\n         minValue = v\n         }\n     }\n     return minElem\n}\n\n/**\n * Returns the smallest\n value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of\n values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws\n NoSuchElementException if the array is empty.\n\n *\n @SinceKotlin("1.4")\n @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n @OverloadResolution\n ByLambdaReturnType\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline fun\n UIntArray.minOf(selector: (UInt) -> Double): Double {\n     if (isEmpty()) throw NoSuchElementException()\n     var minValue = selector(this[0])\n     for (i in 1..lastIndex) {\n         val v = selector(this[i])\n         minValue =\n minOf(minValue, v)\n     }\n     return minValue\n}\n\n/**\n * Returns the smallest value among all values produced\n by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector]\n function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n\n *\n @SinceKotlin("1.4")\n @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n @OverloadResolution\n ByLambdaReturnType\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline fun\n ULongArray.minOf(selector: (ULong) -> Double): Double {\n     if (isEmpty()) throw NoSuchElementException()\n     var minValue = selector(this[0])\n     for (i in 1..lastIndex) {\n         val v = selector(this[i])\n         minValue =\n minOf(minValue, v)\n     }\n     return minValue\n}\n\n/**\n * Returns the smallest value among all values produced\n by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector]\n function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n\n *\n @SinceKotlin("1.4")\n @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n @OverloadResolution\n ByLambdaReturnType\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline fun\n UByteArray.minOf(selector: (UByte) -> Double): Double {\n     if (isEmpty()) throw NoSuchElementException()\n     var minValue = selector(this[0])\n     for (i in 1..lastIndex) {\n         val v = selector(this[i])\n         minValue =\n minOf(minValue, v)\n     }\n     return minValue\n}\n\n/**\n * Returns the smallest value among all values produced\n by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector]\n function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n\n *\n @SinceKotlin("1.4")\n @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n @OverloadResolution\n ByLambdaReturnType\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline fun\n UShortArray.minOf(selector: (UShort) -> Double): Double {\n     if (isEmpty()) throw NoSuchElementException()\n     var minValue = selector(this[0])\n     for (i in 1..lastIndex) {\n         val v = selector(this[i])\n         minValue =\n minOf(minValue, v)\n     }\n     return minValue\n}\n\n/**\n * Returns the smallest value among all values produced\n by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector]\n function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n\n *\n @SinceKotlin("1.4")\n @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n @OverloadResolution\n ByLambdaReturnType\n @ExperimentalUnsignedTypes\n @kotlin.internal.InlineOnly\n public inline fun\n UIntArray.minOf(selector: (UInt) -> Float): Float {\n     if (isEmpty()) throw NoSuchElementException()\n     var\n minValue = selector(this[0])\n     for (i in 1..lastIndex) {\n         val v = selector(this[i])\n         minValue =\n minOf(minValue, v)\n     }\n     return minValue\n}\n\n/**\n * Returns the smallest value among all values produced\n by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector]\n function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n\n *\n @SinceKotlin("1.4")\n @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n @OverloadResolution

```

```

ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.minOf(selector: (ULong) -> Float): Float {\n  if (isEmpty()) throw NoSuchElementException()\n
var minValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    minValue =
minOf(minValue, v)\n  }\n  return minValue\n}\n\n/**\n * Returns the smallest value among all values produced
by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector]
function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.minOf(selector: (UByte) -> Float): Float {\n  if (isEmpty()) throw NoSuchElementException()\n  var
minValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    minValue =
minOf(minValue, v)\n  }\n  return minValue\n}\n\n/**\n * Returns the smallest value among all values produced
by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector]
function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.minOf(selector: (UShort) -> Float): Float {\n  if (isEmpty()) throw NoSuchElementException()\n
var minValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v = selector(this[i])\n    minValue =
minOf(minValue, v)\n  }\n  return minValue\n}\n\n/**\n * Returns the smallest value among all values produced
by [selector] function\n * applied to each element in the array.\n * \n * @throws NoSuchElementException if the
array is empty.\n
*\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :
Comparable<R>> UIntArray.minOf(selector: (UInt) -> R): R {\n  if (isEmpty()) throw
NoSuchElementException()\n  var minValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v =
selector(this[i])\n    if (minValue > v) {\n      minValue = v\n    }\n  }\n  return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the
array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :
Comparable<R>> ULongArray.minOf(selector: (ULong) -> R): R {\n  if (isEmpty()) throw
NoSuchElementException()\n  var minValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v =
selector(this[i])\n    if (minValue > v) {\n      minValue = v\n    }\n  }\n  return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the
array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :
Comparable<R>> UByteArray.minOf(selector: (UByte) -> R): R {\n  if (isEmpty()) throw
NoSuchElementException()\n  var minValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v =
selector(this[i])\n    if (minValue > v) {\n      minValue = v\n    }\n  }\n  return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the
array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*\n*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :
Comparable<R>> UShortArray.minOf(selector: (UShort) -> R): R {\n  if (isEmpty()) throw
NoSuchElementException()\n  var minValue = selector(this[0])\n  for (i in 1..lastIndex) {\n    val v =
selector(this[i])\n    if (minValue > v) {\n      minValue = v\n    }\n  }\n  return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array
or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned

```

result is `NaN`.\n

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nUIntArray.minOfOrNull(selector: (UInt) -> Double): Double? {\n    if (isEmpty()) return null\n    var minValue =\n    selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector]\n function is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nULongArray.minOfOrNull(selector: (ULong) -> Double): Double? {\n    if (isEmpty()) return null\n    var minValue\n    = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by\n [selector] function is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nUByteArray.minOfOrNull(selector: (UByte) -> Double): Double? {\n    if (isEmpty()) return null\n    var minValue\n    = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by\n [selector] function is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nUShortArray.minOfOrNull(selector: (UShort) -> Double): Double? {\n    if (isEmpty()) return null\n    var minValue\n    = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by\n [selector] function is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nUIntArray.minOfOrNull(selector: (UInt) -> Float): Float? {\n    if (isEmpty()) return null\n    var minValue =\n    selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector]\n function is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nULongArray.minOfOrNull(selector: (ULong) -> Float): Float? {\n    if (isEmpty()) return null\n    var minValue =\n    selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector]\n function is `NaN`, the returned result is `NaN`.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nUByteArray.minOfOrNull(selector: (UByte) -> Float): Float? {\n    if (isEmpty()) return null\n    var minValue =\n    selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector]\n function is `NaN`, the returned result is `NaN`.\n
```

applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nUShortArray.minOrNull(selector: (UShort) -> Float): Float? {\n    if (isEmpty()) return null\n    var minValue =\n    selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> UIntArray.minOrNull(selector: (UInt) -> R): R? {\n    if (isEmpty()) return null\n    var\n    minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (minValue > v) {\n            minValue = v\n        }\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all values\n * produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> ULongArray.minOrNull(selector: (ULong) -> R): R? {\n    if (isEmpty()) return null\n    var\n    minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (minValue > v) {\n            minValue = v\n        }\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all values\n * produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> UByteArray.minOrNull(selector: (UByte) -> R): R? {\n    if (isEmpty()) return null\n    var\n    minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (minValue > v) {\n            minValue = v\n        }\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all values\n * produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> UShortArray.minOrNull(selector: (UShort) -> R): R? {\n    if (isEmpty()) return null\n    var\n    minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (minValue > v) {\n            minValue = v\n        }\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value according to the\n * provided [comparator]\n * among all values produced by [selector] function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>\nUIntArray.minOfWith(comparator: Comparator<in R>, selector: (UInt) -> R): R {\n    if (isEmpty()) throw\n    NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v =\n        selector(this[i])\n        if (comparator.compare(minValue, v) > 0) {\n            minValue = v\n        }\n    }\n    return\n    minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values\n * produced by [selector] function applied to each element in the array.\n * \n * @throws NoSuchElementException if\n * the array is empty.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>\nULongArray.minOfWith(comparator: Comparator<in R>, selector: (ULong) -> R): R {\n    if (isEmpty()) throw\n    NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v =\n        selector(this[i])\n        if (comparator.compare(minValue, v) > 0) {\n            minValue = v\n        }\n    }\n    return\n    minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values\n * produced by [selector] function applied to each element in the array.\n * \n * @throws NoSuchElementException if\n
```

the array is empty.\n

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>\nUByteArray.minOfWith(comparator: Comparator<in R>, selector: (UByte) -> R): R {\n    if (isEmpty()) throw\n    NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v =\n        selector(this[i])\n        if (comparator.compare(minValue, v) > 0) {\n            minValue = v\n        }\n    }\n    return\n    minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values\n * produced by [selector] function applied to each element in the array.\n * \n * @throws NoSuchElementException if\n * the array is empty.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>\nUShortArray.minOfWith(comparator: Comparator<in R>, selector: (UShort) -> R): R {\n    if (isEmpty()) throw\n    NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v =\n        selector(this[i])\n        if (comparator.compare(minValue, v) > 0) {\n            minValue = v\n        }\n    }\n    return\n    minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values\n * produced by [selector] function applied to each element in the array or `null` if there are no elements.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>\nUIntArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (UInt) -> R): R? {\n    if (isEmpty()) return\n    null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if\n        (comparator.compare(minValue, v) > 0) {\n            minValue = v\n        }\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector]\n * function applied to each element in the array or `null` if there are no elements.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>\nULongArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (ULong) -> R): R? {\n    if (isEmpty())\n    return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if\n        (comparator.compare(minValue, v) > 0) {\n            minValue = v\n        }\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector]\n * function applied to each element in the array or `null` if there are no elements.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>\nUByteArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (UByte) -> R): R? {\n    if (isEmpty())\n    return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if\n        (comparator.compare(minValue, v) > 0) {\n            minValue = v\n        }\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector]\n * function applied to each element in the array or `null` if there are no elements.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>\nUShortArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (UShort) -> R): R? {\n    if (isEmpty())\n    return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if\n        (comparator.compare(minValue, v) > 0) {\n            minValue = v\n        }\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n
```

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.minOrNull(): UInt? {\n    if\n    (isEmpty()) return null\n    var min = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (min > e) min =\n        e\n    }\n    return min\n}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n
```

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.minOrNull(): ULong? {\n    if\n    (isEmpty()) return null\n    var min = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (min > e) min =\n
```



```

* @sample samples.collections.Collections.Aggregates.none\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.none(): Boolean {\n    return isEmpty()\n}\n\n/**\n * Returns `true` if no elements match the given
[predicate].\n * \n * @sample samples.collections.Collections.Aggregates.noneWithPredicate\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.none(predicate: (UInt) -> Boolean): Boolean {\n    for (element in this) if (predicate(element)) return
false\n    return true\n}\n\n/**\n * Returns `true` if no elements match the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.noneWithPredicate\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.none(predicate: (ULong) -> Boolean): Boolean {\n    for (element in this) if (predicate(element)) return
false\n    return true\n}\n\n/**\n * Returns `true` if no elements match the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.noneWithPredicate\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.none(predicate: (UByte) -> Boolean): Boolean {\n    for (element in this) if (predicate(element)) return
false\n    return true\n}\n\n/**\n * Returns `true` if no elements match the given [predicate].\n * \n * @sample
samples.collections.Collections.Aggregates.noneWithPredicate\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.none(predicate: (UShort) -> Boolean): Boolean {\n    for (element in this) if (predicate(element))
return false\n    return true\n}\n\n/**\n * Performs the given [action] on each element and returns the array itself
afterwards.\n * \n *\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun UIntArray.onEach(action: (UInt) -> Unit): UIntArray {\n    return apply { for (element in this)
action(element) }\n}\n\n/**\n * Performs the given [action] on each element and returns the array itself
afterwards.\n * \n *\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun ULongArray.onEach(action: (ULong) -> Unit): ULongArray {\n    return apply { for (element in this)
action(element) }\n}\n\n/**\n * Performs the given [action] on each element and returns the array itself
afterwards.\n * \n *\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun UByteArray.onEach(action: (UByte) -> Unit): UByteArray {\n    return apply { for (element in this)
action(element) }\n}\n\n/**\n * Performs the given [action] on each element and returns the array itself
afterwards.\n * \n *\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun UShortArray.onEach(action: (UShort) -> Unit): UShortArray {\n    return apply { for (element in this)
action(element) }\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the
element,\n * and returns the array itself afterwards.\n * \n * @param [action] function that takes the index of an element
and the element itself\n * and performs the action on the element.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.onEachIndexed(action: (index: Int, UInt) -> Unit): UIntArray {\n    return apply {
forEachIndexed(action) }\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index
with the element,\n * and returns the array itself afterwards.\n * \n * @param [action] function that takes the index of an
element and the element itself\n * and performs the action on the element.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.onEachIndexed(action: (index: Int, ULong) -> Unit): ULongArray {\n    return apply {
forEachIndexed(action) }\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index
with the element,\n * and returns the array itself afterwards.\n * \n * @param [action] function that takes the index of an
element and the element itself\n * and performs the action on the element.\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.onEachIndexed(action: (index: Int, UByte) -> Unit): UByteArray {\n    return apply {
forEachIndexed(action) }\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index
with the element,\n * and returns the array itself afterwards.\n * \n * @param [action] function that takes the index of an
element and the element itself\n * and performs the action on the element.\n

```

```

*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.onEachIndexed(action: (index: Int, UShort) -> Unit): UShortArray {\n    return apply {
forEachIndexed(action) }\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation]
from left to right\n * to current accumulator value and each element.\n * \n * Throws an exception if this array is
empty. If the array can be empty in an expected way,\n * please use [reduceOrNull] instead. It returns `null` when its
receiver is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n *
and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.reduce(operation: (acc: UInt, UInt) -> UInt): UInt {\n    if (isEmpty())\n        throw
UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = this[0]\n    for (index in
1..lastIndex) {\n        accumulator = operation(accumulator, this[index])\n    }\n    return accumulator\n}\n\n/**\n *
Accumulates value starting with the first element and applying [operation] from left to right\n * to current
accumulator value and each element.\n * \n * Throws an exception if this array is empty. If the array can be empty
in an expected way,\n * please use [reduceOrNull] instead. It returns `null` when its receiver is empty.\n * \n *
@param [operation] function that takes current accumulator value and an element,\n * and calculates the next
accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.reduce(operation: (acc: ULong, ULong) -> ULong): ULong {\n    if (isEmpty())\n        throw
UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = this[0]\n    for (index in
1..lastIndex) {\n        accumulator = operation(accumulator, this[index])\n    }\n    return accumulator\n}\n\n/**\n *
Accumulates value starting with the first element and applying [operation] from left to right\n * to current
accumulator value and each element.\n * \n * Throws an exception if this array is empty. If the array can be empty
in an expected way,\n * please use [reduceOrNull] instead. It returns `null` when its receiver is empty.\n * \n *
@param [operation] function that takes current accumulator value and an element,\n * and calculates the next
accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.reduce(operation: (acc: UByte, UByte) -> UByte): UByte {\n    if (isEmpty())\n        throw
UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = this[0]\n    for (index in
1..lastIndex) {\n        accumulator = operation(accumulator, this[index])\n    }\n    return accumulator\n}\n\n/**\n *
Accumulates value starting with the first element and applying [operation] from left to right\n * to current
accumulator value and each element.\n * \n * Throws an exception if this array is empty. If the array can be empty
in an expected way,\n * please use [reduceOrNull] instead. It returns `null` when its receiver is empty.\n * \n *
@param [operation] function that takes current accumulator value and an element,\n * and calculates the next
accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.reduce(operation: (acc: UShort, UShort) -> UShort): UShort {\n    if (isEmpty())\n        throw
UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = this[0]\n    for (index in
1..lastIndex) {\n        accumulator = operation(accumulator, this[index])\n    }\n    return accumulator\n}\n\n/**\n *
Accumulates value starting with the first element and applying [operation] from left to right\n * to current
accumulator value and each element with its index in the original array.\n * \n * Throws an exception if this array is
empty. If the array can be empty in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null`
when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, current
accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduce\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.reduceIndexed(operation: (index: Int, acc: UInt, UInt) -> UInt): UInt {\n    if (isEmpty())\n        throw
UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = this[0]\n    for (index in
1..lastIndex) {\n        accumulator = operation(index, accumulator, this[index])\n    }\n    return

```

accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nULongArray.reduceIndexed(operation: (index: Int, acc: ULong, ULong) -> ULong): ULong {\n    if (isEmpty())\n        throw UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = this[0]\n    for (index in 1..lastIndex) {\n        accumulator = operation(index, accumulator, this[index])\n    }\n    return\n    accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n
```

accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nUByteArray.reduceIndexed(operation: (index: Int, acc: UByte, UByte) -> UByte): UByte {\n    if (isEmpty())\n        throw UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = this[0]\n    for (index in 1..lastIndex) {\n        accumulator = operation(index, accumulator, this[index])\n    }\n    return\n    accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n
```

accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nUShortArray.reduceIndexed(operation: (index: Int, acc: UShort, UShort) -> UShort): UShort {\n    if (isEmpty())\n        throw UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = this[0]\n    for (index in 1..lastIndex) {\n        accumulator = operation(index, accumulator, this[index])\n    }\n    return\n    accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n
```

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nUIntArray.reduceIndexedOrNull(operation: (index: Int, acc: UInt, UInt) -> UInt): UInt? {\n    if (isEmpty())\n        return null\n    var accumulator = this[0]\n    for (index in 1..lastIndex) {\n        accumulator = operation(index, accumulator, this[index])\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n
```

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\nULongArray.reduceIndexedOrNull(operation: (index: Int, acc: ULong, ULong) -> ULong): ULong? {\n    if (isEmpty())\n        return null\n    var accumulator = this[0]\n    for (index in 1..lastIndex) {\n        accumulator =\n        operation(index, accumulator, this[index])\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element\n
```

with its index in the original array.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n * \n * @SinceKotlin("1.4")\n * @ExperimentalUnsignedTypes\n * @kotlin.internal.InlineOnly\n * public inline fun UByteArray.reduceIndexedOrNull(operation: (index: Int, acc: UByte, UByte) -> UByte): UByte? {\n * if (isEmpty())\n * return null\n * var accumulator = this[0]\n * for (index in 1..lastIndex) {\n * accumulator = operation(index, accumulator, this[index])\n * }\n * return accumulator\n * }\n * \n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n * \n * @SinceKotlin("1.4")\n * @ExperimentalUnsignedTypes\n * @WasExperimental(ExperimentalStdlibApi::class)\n * @kotlin.internal.InlineOnly\n * public inline fun UIntArray.reduceOrNull(operation: (acc: UInt, UInt) -> UInt): UInt? {\n * if (isEmpty())\n * return null\n * var accumulator = this[0]\n * for (index in 1..lastIndex) {\n * accumulator = operation(accumulator, this[index])\n * }\n * return accumulator\n * }\n * \n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n * \n * @SinceKotlin("1.4")\n * @ExperimentalUnsignedTypes\n * @WasExperimental(ExperimentalStdlibApi::class)\n * @kotlin.internal.InlineOnly\n * public inline fun ULongArray.reduceOrNull(operation: (acc: ULong, ULong) -> ULong): ULong? {\n * if (isEmpty())\n * return null\n * var accumulator = this[0]\n * for (index in 1..lastIndex) {\n * accumulator = operation(accumulator, this[index])\n * }\n * return accumulator\n * }\n * \n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n * \n * @SinceKotlin("1.4")\n * @ExperimentalUnsignedTypes\n * @WasExperimental(ExperimentalStdlibApi::class)\n * @kotlin.internal.InlineOnly\n * public inline fun UByteArray.reduceOrNull(operation: (acc: UByte, UByte) -> UByte): UByte? {\n * if (isEmpty())\n * return null\n * var accumulator = this[0]\n * for (index in 1..lastIndex) {\n * accumulator = operation(accumulator, this[index])\n * }\n * return accumulator\n * }\n * \n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n * \n * @SinceKotlin("1.4")\n * @ExperimentalUnsignedTypes\n * @WasExperimental(ExperimentalStdlibApi::class)\n * @kotlin.internal.InlineOnly\n * public inline fun UShortArray.reduceOrNull(operation: (acc: UShort, UShort) -> UShort): UShort? {\n * if (isEmpty())\n * return null\n * var accumulator = this[0]\n * for (index in 1..lastIndex) {\n * accumulator = operation(accumulator, this[index])\n * }\n * return accumulator\n * }\n * \n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element and current

accumulator value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.reduceRight(operation: (UInt, acc: UInt) -> UInt): UInt {\n    var index = lastIndex\n    if (index < 0)
throw UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = get(index--)\n    while (index >= 0) {\n        accumulator = operation(get(index--), accumulator)\n    }\n    return
accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to
left\n * to each element and current accumulator value.\n * \n * Throws an exception if this array is empty. If the
array can be empty in an expected way,\n * please use [reduceRightOrNull] instead. It returns `null` when its
receiver is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRight\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.reduceRight(operation: (ULong, acc: ULong) -> ULong): ULong {\n    var index = lastIndex\n    if
(index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator =
get(index--)\n    while (index >= 0) {\n        accumulator = operation(get(index--), accumulator)\n    }\n    return
accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to
left\n * to each element and current accumulator value.\n * \n * Throws an exception if this array is empty. If the
array can be empty in an expected way,\n * please use [reduceRightOrNull] instead. It returns `null` when its
receiver is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRight\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.reduceRight(operation: (UByte, acc: UByte) -> UByte): UByte {\n    var index = lastIndex\n    if (index
< 0) throw UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = get(index--)\n    while (index >= 0) {\n        accumulator = operation(get(index--), accumulator)\n    }\n    return
accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to
left\n * to each element and current accumulator value.\n * \n * Throws an exception if this array is empty. If the
array can be empty in an expected way,\n * please use [reduceRightOrNull] instead. It returns `null` when its
receiver is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRight\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.reduceRight(operation: (UShort, acc: UShort) -> UShort): UShort {\n    var index = lastIndex\n    if
(index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator =
get(index--)\n    while (index >= 0) {\n        accumulator = operation(get(index--), accumulator)\n    }\n    return
accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to
left\n * to each element with its index in the original array and current accumulator value.\n * \n * Throws an
exception if this array is empty. If the array can be empty in an expected way,\n * please use
[reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation]
function that takes the index of an element, the element itself and current accumulator value,\n * and calculates the
next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.reduceRightIndexed(operation: (index: Int, UInt, acc: UInt) -> UInt): UInt {\n    var index = lastIndex\n    if
(index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator =
get(index--)\n    while (index >= 0) {\n        accumulator = operation(index, get(index), accumulator)\n        --index\n    }\n    return
accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to
left\n * to each element with its index in the original array and current accumulator value.\n * \n * Throws an
exception if this array is empty. If the array can be empty in an expected way,\n * please use
[reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation]
function that takes the index of an element, the element itself and current accumulator value,\n * and calculates the
next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n
```

```

    }
    return accumulator
}

/** Accumulates value starting with the last element and applying [operation]
from right to left * to each element with its index in the original array and current accumulator value.
Throws an exception if this array is empty. If the array can be empty in an expected way, please use
[reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.
@param [operation] function that takes the index of an element, the element itself and current accumulator value,
and calculates the next accumulator value.
@sample samples.collections.Collections.Aggregates.reduceRight

*/
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UIntArray.reduceRightIndexed(operation: (index: Int, ULong, acc: ULong) -> ULong): ULong {
    var index =
    lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var
    accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index),
        accumulator)
        --index
    }
    return accumulator
}

/** Accumulates value starting with the last
element and applying [operation] from right to left * to each element with its index in the original array and
current accumulator value.
Throws an exception if this array is empty. If the array can be empty in an
expected way, please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.
@param [operation] function that takes the index of an element, the element itself and current accumulator
value,
and calculates the next accumulator value.
@sample
samples.collections.Collections.Aggregates.reduceRight

*/
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UByteArray.reduceRightIndexed(operation: (index: Int, UByte, acc: UByte) -> UByte): UByte {
    var index =
    lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var
    accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index),
        accumulator)
        --index
    }
    return accumulator
}

/** Accumulates value starting with the last
element and applying [operation] from right to left * to each element with its index in the original array and
current accumulator value.
Throws an exception if this array is empty. If the array can be empty in an
expected way, please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.
@param [operation] function that takes the index of an element, the element itself and current accumulator
value,
and calculates the next accumulator value.
@sample
samples.collections.Collections.Aggregates.reduceRight

*/
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UShortArray.reduceRightIndexed(operation: (index: Int, UShort, acc: UShort) -> UShort): UShort {
    var index =
    lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var
    accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index),
        accumulator)
        --index
    }
    return accumulator
}

/** Accumulates value starting with the last
element and applying [operation] from right to left * to each element with its index in the original array and
current accumulator value.
Returns `null` if the array is empty.
@param [operation] function that
takes the index of an element, the element itself and current accumulator value,
and calculates the next
accumulator value.
@sample samples.collections.Collections.Aggregates.reduceRightOrNull

*/
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UIntArray.reduceRightIndexedOrNull(operation: (index: Int, UInt, acc: UInt) -> UInt?): UInt? {
    var index =
    lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

/** Accumulates value starting with the last element and applying [operation] from right to left * to each element with
its index in the original array and current accumulator value.
Returns `null` if the array is empty.
@param [operation] function that takes the index of an element, the element itself and current accumulator value,
and calculates the next accumulator value.
@sample
samples.collections.Collections.Aggregates.reduceRightOrNull

*/
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
ULongArray.reduceRightIndexedOrNull(operation: (index: Int, ULong, acc: ULong) -> ULong?): ULong? {
    var

```

```

index = lastIndex\n  if (index < 0) return null\n  var accumulator = get(index--)\n  while (index >= 0) {\n
accumulator = operation(index, get(index), accumulator)\n    --index\n  }\n  return accumulator\n}\n\n/**\n *
Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with
its index in the original array and current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n *
@param [operation] function that takes the index of an element, the element itself and current accumulator value,\n
* and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n
*/\n\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.reduceRightIndexedOrNull(operation: (index: Int, UByte, acc: UByte) -> UByte): UByte? {\n  var
index = lastIndex\n  if (index < 0) return null\n  var accumulator = get(index--)\n  while (index >= 0) {\n
accumulator = operation(index, get(index), accumulator)\n    --index\n  }\n  return accumulator\n}\n\n/**\n *
Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with
its index in the original array and current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n *
@param [operation] function that takes the index of an element, the element itself and current accumulator value,\n
* and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n
*/\n\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.reduceRightIndexedOrNull(operation: (index: Int, UShort, acc: UShort) -> UShort): UShort? {\n  var
index = lastIndex\n  if (index < 0) return null\n  var accumulator = get(index--)\n  while (index >= 0) {\n
accumulator = operation(index, get(index), accumulator)\n    --index\n  }\n  return accumulator\n}\n\n/**\n *
Accumulates value starting with the last element and applying [operation] from right to left\n * to each element and
current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that
takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n
*/\n\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun UIntArray.reduceRightOrNull(operation: (UInt, acc: UInt) -> UInt):
UInt? {\n  var index = lastIndex\n  if (index < 0) return null\n  var accumulator = get(index--)\n  while (index
>= 0) {\n    accumulator = operation(get(index--), accumulator)\n  }\n  return accumulator\n}\n\n/**\n *
Accumulates value starting with the last element and applying [operation] from right to left\n * to each element and
current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that
takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n
*/\n\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun ULongArray.reduceRightOrNull(operation: (ULong, acc: ULong) ->
ULong): ULong? {\n  var index = lastIndex\n  if (index < 0) return null\n  var accumulator = get(index--)\n
while (index >= 0) {\n    accumulator = operation(get(index--), accumulator)\n  }\n  return
accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to
left\n * to each element and current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n * @param
[operation] function that takes an element and current accumulator value,\n * and calculates the next accumulator
value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRightOrNull\n
*/\n\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun UByteArray.reduceRightOrNull(operation: (UByte, acc: UByte) ->
UByte): UByte? {\n  var index = lastIndex\n  if (index < 0) return null\n  var accumulator = get(index--)\n
while (index >= 0) {\n    accumulator = operation(get(index--), accumulator)\n  }\n  return
accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to
left\n * to each element and current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n * @param
[operation] function that takes an element and current accumulator value,\n * and calculates the next accumulator
value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRightOrNull\n

```

```

*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun UShortArray.reduceRightOrNull(operation: (UShort, acc: UShort) ->
UShort): UShort? {\n    var index = lastIndex\n    if (index < 0) return null\n    var accumulator = get(index--)\n
while (index >= 0) {\n    accumulator = operation(get(index--), accumulator)\n    }\n    return
accumulator}\n\n/**\n * Returns a list containing successive accumulation values generated by applying
[operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n *
\n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the
previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an
element, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningFold\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UIntArray.runningFold(initial: R, operation: (acc: R, UInt) -> R): List<R> {\n    if (isEmpty()) return
listOf(initial)\n    val result = ArrayList<R>(size + 1).apply { add(initial) }\n    var accumulator = initial\n
for (element in this) {\n    accumulator = operation(accumulator, element)\n    result.add(accumulator)\n    }\n
return result}\n\n/**\n * Returns a list containing successive accumulation values generated by applying
[operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n *
\n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the
previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an
element, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningFold\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
ULongArray.runningFold(initial: R, operation: (acc: R, ULong) -> R): List<R> {\n    if (isEmpty()) return
listOf(initial)\n    val result = ArrayList<R>(size + 1).apply { add(initial) }\n    var accumulator = initial\n
for (element in this) {\n    accumulator = operation(accumulator, element)\n    result.add(accumulator)\n    }\n
return result}\n\n/**\n * Returns a list containing successive accumulation values generated by applying
[operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n *
\n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the
previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an
element, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningFold\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UByteArray.runningFold(initial: R, operation: (acc: R, UByte) -> R): List<R> {\n    if (isEmpty()) return
listOf(initial)\n    val result = ArrayList<R>(size + 1).apply { add(initial) }\n    var accumulator = initial\n
for (element in this) {\n    accumulator = operation(accumulator, element)\n    result.add(accumulator)\n    }\n
return result}\n\n/**\n * Returns a list containing successive accumulation values generated by applying
[operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n *
\n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the
previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an
element, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningFold\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UShortArray.runningFold(initial: R, operation: (acc: R, UShort) -> R): List<R> {\n    if (isEmpty()) return
listOf(initial)\n    val result = ArrayList<R>(size + 1).apply { add(initial) }\n    var accumulator = initial\n
for (element in this) {\n    accumulator = operation(accumulator, element)\n    result.add(accumulator)\n    }\n
return result}\n\n/**\n * Returns a list containing successive accumulation values generated by applying
[operation] from left to right\n * to each element, its index in the original array and current accumulator value that
starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n *
otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the

```

index of an element, current accumulator value and the element itself, and calculates the next accumulator value.

```

@sample samples.collections.Collections.Aggregates.runningFold
*/n@SinceKotlin("1.4")n@ExperimentalUnsignedTypesn@kotlin.internal.InlineOnlynpublic inline fun <R>
UIntArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, UInt) -> R): List<R> {n if (isEmpty())
return listOf(initial)n val result = ArrayList<R>(size + 1).apply { add(initial) }n var accumulator = initialn
for (index in indices) {n accumulator = operation(index, accumulator, this[index])n
result.add(accumulator)n }n return resultn}n/n/**n * Returns a list containing successive accumulation
values generated by applying [operation] from left to rightn * to each element, its index in the original array and
current accumulator value that starts with [initial] value.n * n * Note that `acc` value passed to [operation] function
should not be mutated;n * otherwise it would affect the previous value in resulting list.n * n * @param [operation]
function that takes the index of an element, current accumulator value and the element itself, and calculates the
next accumulator value.n * n * @sample samples.collections.Collections.Aggregates.runningFoldn
*/n@SinceKotlin("1.4")n@ExperimentalUnsignedTypesn@kotlin.internal.InlineOnlynpublic inline fun <R>
ULongArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, ULong) -> R): List<R> {n if
(isEmpty()) return listOf(initial)n val result = ArrayList<R>(size + 1).apply { add(initial) }n var accumulator =
initialn for (index in indices) {n accumulator = operation(index, accumulator, this[index])n
result.add(accumulator)n }n return resultn}n/n/n/**n * Returns a list containing successive accumulation
values generated by applying [operation] from left to rightn * to each element, its index in the original array and
current accumulator value that starts with [initial] value.n * n * Note that `acc` value passed to [operation] function
should not be mutated;n * otherwise it would affect the previous value in resulting list.n * n * @param [operation]
function that takes the index of an element, current accumulator value and the element itself, and calculates the
next accumulator value.n * n * @sample samples.collections.Collections.Aggregates.runningFoldn
*/n@SinceKotlin("1.4")n@ExperimentalUnsignedTypesn@kotlin.internal.InlineOnlynpublic inline fun <R>
UByteArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, UByte) -> R): List<R> {n if
(isEmpty()) return listOf(initial)n val result = ArrayList<R>(size + 1).apply { add(initial) }n var accumulator =
initialn for (index in indices) {n accumulator = operation(index, accumulator, this[index])n
result.add(accumulator)n }n return resultn}n/n/n/**n * Returns a list containing successive accumulation
values generated by applying [operation] from left to rightn * to each element, its index in the original array and
current accumulator value that starts with [initial] value.n * n * Note that `acc` value passed to [operation] function
should not be mutated;n * otherwise it would affect the previous value in resulting list.n * n * @param [operation]
function that takes the index of an element, current accumulator value and the element itself, and calculates the
next accumulator value.n * n * @sample samples.collections.Collections.Aggregates.runningFoldn
*/n@SinceKotlin("1.4")n@ExperimentalUnsignedTypesn@kotlin.internal.InlineOnlynpublic inline fun <R>
UShortArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, UShort) -> R): List<R> {n if
(isEmpty()) return listOf(initial)n val result = ArrayList<R>(size + 1).apply { add(initial) }n var accumulator =
initialn for (index in indices) {n accumulator = operation(index, accumulator, this[index])n
result.add(accumulator)n }n return resultn}n/n/n/**n * Returns a list containing successive accumulation
values generated by applying [operation] from left to rightn * to each element and current accumulator value that
starts with the first element of this array.n * n * Note that `acc` value passed to [operation] function should not be
mutated;n * otherwise it would affect the previous value in resulting list.n * n * @param [operation] function that
takes current accumulator value and an element, and calculates the next accumulator value.n * n * @sample
samples.collections.Collections.Aggregates.runningReducen
*/n@SinceKotlin("1.4")n@ExperimentalUnsignedTypesn@kotlin.internal.InlineOnlynpublic inline fun
UIntArray.runningReduce(operation: (acc: UInt, UInt) -> UInt): List<UInt> {n if (isEmpty()) return
emptyList()n var accumulator = this[0]n val result = ArrayList<UInt>(size).apply { add(accumulator) }n for
(index in 1 until size) {n accumulator = operation(accumulator, this[index])n result.add(accumulator)n
}n return resultn}n/n/n/**n * Returns a list containing successive accumulation values generated by applying
[operation] from left to rightn * to each element and current accumulator value that starts with the first element of

```

this array.

* Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.

* @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

* @sample

```

samples.collections.Collections.Aggregates.runningReduce
*/
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
ULongArray.runningReduce(operation: (acc: ULong, ULong) -> ULong): List<ULong> {
    if (isEmpty()) return
    emptyList()
    var accumulator = this[0]
    val result = ArrayList<ULong>(size).apply { add(accumulator) }
    for (index in 1 until size) {
        accumulator = operation(accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

```

* Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with the first element of this array.

* Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.

* @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

* @sample

```

samples.collections.Collections.Aggregates.runningReduce
*/
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UByteArray.runningReduce(operation: (acc: UByte, UByte) -> UByte): List<UByte> {
    if (isEmpty()) return
    emptyList()
    var accumulator = this[0]
    val result = ArrayList<UByte>(size).apply { add(accumulator) }
    for (index in 1 until size) {
        accumulator = operation(accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

```

* Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with the first element of this array.

* Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.

* @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

* @sample

```

samples.collections.Collections.Aggregates.runningReduce
*/
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UShortArray.runningReduce(operation: (acc: UShort, UShort) -> UShort): List<UShort> {
    if (isEmpty()) return
    emptyList()
    var accumulator = this[0]
    val result = ArrayList<UShort>(size).apply { add(accumulator) }
    for (index in 1 until size) {
        accumulator = operation(accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

```

* Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original array and current accumulator value that starts with the first element of this array.

* Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.

* @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.

* @sample

```

samples.collections.Collections.Aggregates.runningReduce
*/
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UIntArray.runningReduceIndexed(operation: (index: Int, acc: UInt, UInt) -> UInt): List<UInt> {
    if (isEmpty()) return
    emptyList()
    var accumulator = this[0]
    val result = ArrayList<UInt>(size).apply { add(accumulator) }
    for (index in 1 until size) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

```

* Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original array and current accumulator value that starts with the first element of this array.

* Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.

* @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.

* @sample

```

samples.collections.Collections.Aggregates.runningReduce
*/
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
ULongArray.runningReduceIndexed(operation: (index: Int, acc: ULong, ULong) -> ULong): List<ULong> {
    if (isEmpty()) return
    emptyList()
    var accumulator = this[0]
    val result = ArrayList<ULong>(size).apply {

```

```

add(accumulator) }\n  for (index in 1 until size) {\n    accumulator = operation(index, accumulator, this[index])\n    result.add(accumulator)\n  }\n  return result\n}\n\n/**\n * Returns a list containing successive accumulation
values generated by applying [operation] from left to right\n * to each element, its index in the original array and
current accumulator value that starts with the first element of this array.\n * \n * Note that `acc` value passed to
[operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n *
@param [operation] function that takes the index of an element, current accumulator value\n * and the element
itself, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningReduce\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.runningReduceIndexed(operation: (index: Int, acc: UByte, UByte) -> UByte): List<UByte> {\n  if
(isEmpty()) return emptyList()\n  var accumulator = this[0]\n  val result = ArrayList<UByte>(size).apply {\n
add(accumulator) }\n  for (index in 1 until size) {\n    accumulator = operation(index, accumulator, this[index])\n
    result.add(accumulator)\n  }\n  return result\n}\n\n/**\n * Returns a list containing successive accumulation
values generated by applying [operation] from left to right\n * to each element, its index in the original array and
current accumulator value that starts with the first element of this array.\n * \n * Note that `acc` value passed to
[operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n *
@param [operation] function that takes the index of an element, current accumulator value\n * and the element
itself, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.runningReduce\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.runningReduceIndexed(operation: (index: Int, acc: UShort, UShort) -> UShort): List<UShort> {\n  if
(isEmpty()) return emptyList()\n  var accumulator = this[0]\n  val result = ArrayList<UShort>(size).apply {\n
add(accumulator) }\n  for (index in 1 until size) {\n    accumulator = operation(index, accumulator, this[index])\n
    result.add(accumulator)\n  }\n  return result\n}\n\n/**\n * Returns a list containing successive accumulation
values generated by applying [operation] from left to right\n * to each element and current accumulator value that
starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n *
otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current
accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.scan\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun <R> UIntArray.scan(initial: R, operation: (acc: R, UInt) -> R):
List<R> {\n  return runningFold(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation
values generated by applying [operation] from left to right\n * to each element and current accumulator value that
starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n *
otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current
accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.scan\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun <R> ULongArray.scan(initial: R, operation: (acc: R, ULong) -> R):
List<R> {\n  return runningFold(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation
values generated by applying [operation] from left to right\n * to each element and current accumulator value that
starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n *
otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current
accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.scan\n
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun <R> UByteArray.scan(initial: R, operation: (acc: R, UByte) -> R):
List<R> {\n  return runningFold(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation

```

values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample

samples.collections.Collections.Aggregates.scan\n

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <R> UShortArray.scan(initial: R, operation: (acc: R, UShort) -> R): List<R> {\n    return runningFold(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample
```

```
samples.collections.Collections.Aggregates.scan\n\n*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <R> UIntArray.scanIndexed(initial: R, operation: (index: Int, acc: R, UInt) -> R): List<R> {\n    return runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample
```

```
samples.collections.Collections.Aggregates.scan\n\n*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <R> ULongArray.scanIndexed(initial: R, operation: (index: Int, acc: R, ULong) -> R): List<R> {\n    return runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample
```

```
samples.collections.Collections.Aggregates.scan\n\n*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <R> UByteArray.scanIndexed(initial: R, operation: (index: Int, acc: R, UByte) -> R): List<R> {\n    return runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample
```

```
samples.collections.Collections.Aggregates.scan\n\n*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <R> UShortArray.scanIndexed(initial: R, operation: (index: Int, acc: R, UShort) -> R): List<R> {\n    return runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n * \n * @Deprecated("Use sumOf instead.", ReplaceWith("this.sumOf(selector)"))\n * @DeprecatedSinceKotlin(warningSince = "1.5")\n * @SinceKotlin("1.3")\n * @ExperimentalUnsignedTypes\n * @kotlin.internal.InlineOnly\n * public inline fun UIntArray.sumBy(selector: (UInt) -> UInt): UInt {\n *     var sum: UInt = 0u\n *     for (element in this) {\n *         sum += selector(element)\n *     }\n *     return sum\n * }\n\n/**\n * Returns the sum of all values produced by [selector] function
```



```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@Suppress("INAPPLICABLE_JVM_NAME")\n@kotlin.jvm.JvmName("sumOfDouble")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.sumOf(selector: (UByte) -> Double): Double {\n    var sum: Double = 0.toDouble()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@Suppress("INAPPLICABLE_JVM_NAME")\n@kotlin.jvm.JvmName("sumOfDouble")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.sumOf(selector: (UShort) -> Double): Double {\n    var sum: Double = 0.toDouble()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@Suppress("INAPPLICABLE_JVM_NAME")\n@kotlin.jvm.JvmName("sumOfInt")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.sumOf(selector: (UInt) -> Int): Int {\n    var sum: Int = 0.toInt()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@Suppress("INAPPLICABLE_JVM_NAME")\n@kotlin.jvm.JvmName("sumOfInt")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.sumOf(selector: (ULong) -> Int): Int {\n    var sum: Int = 0.toInt()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@Suppress("INAPPLICABLE_JVM_NAME")\n@kotlin.jvm.JvmName("sumOfInt")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.sumOf(selector: (UByte) -> Int): Int {\n    var sum: Int = 0.toInt()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@Suppress("INAPPLICABLE_JVM_NAME")\n@kotlin.jvm.JvmName("sumOfInt")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.sumOf(selector: (UShort) -> Int): Int {\n    var sum: Int = 0.toInt()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@Suppress("INAPPLICABLE_JVM_NAME")\n@kotlin.jvm.JvmName("sumOfLong")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.sumOf(selector: (UInt) -> Long): Long {\n    var sum: Long = 0.toLong()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@Suppress("INAPPLICABLE_JVM_NAME")\n@kotlin.jvm.JvmName("sumOfLong")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.sumOf(selector: (ULong) -> Long): Long {\n    var sum: Long = 0.toLong()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@Suppress("INAPPLICABLE_JVM_NAME")\n@kotlin.jvm.JvmName("sumOfLong")\n\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.sumOf(selector:\n(UByte) -> Long): Long {\n    var sum: Long = 0.toLong()\n    for (element in this) {\n        sum +=\nselector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function\napplied to each element in the array.\n
```

```
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@Suppress("INAPPLICABLE_JVM_NAME")\n@kotlin.jvm.JvmName("sumOfLong")\n\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.sumOf(selector:\n(UShort) -> Long): Long {\n    var sum: Long = 0.toLong()\n    for (element in this) {\n        sum +=\nselector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function\napplied to each element in the array.\n
```

```
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@Suppress("INAPPLICABLE_JVM_NAME")\n@kotlin.jvm.JvmName("sumOfUInt")\n\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.Inline\nOnly\npublic inline fun UIntArray.sumOf(selector: (UInt) -> UInt): UInt {\n    var sum: UInt = 0.toUInt()\n    for\n(element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all values\nproduced by [selector] function applied to each element in the array.\n
```

```
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@Suppress("INAPPLICABLE_JVM_NAME")\n@kotlin.jvm.JvmName("sumOfUInt")\n\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.Inline\nOnly\npublic inline fun ULongArray.sumOf(selector: (ULong) -> UInt): UInt {\n    var sum: UInt = 0.toUInt()\n    for\n(element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all\nvalues produced by [selector] function applied to each element in the array.\n
```

```
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@Suppress("INAPPLICABLE_JVM_NAME")\n@kotlin.jvm.JvmName("sumOfUInt")\n\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.Inline\nOnly\npublic inline fun UByteArray.sumOf(selector: (UByte) -> UInt): UInt {\n    var sum: UInt = 0.toUInt()\n    for\n(element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all\nvalues produced by [selector] function applied to each element in the array.\n
```

```
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@Suppress("INAPPLICABLE_JVM_NAME")\n@kotlin.jvm.JvmName("sumOfUInt")\n\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.Inline\nOnly\npublic inline fun UShortArray.sumOf(selector: (UShort) -> UInt): UInt {\n    var sum: UInt = 0.toUInt()\n    for\n(element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all\nvalues produced by [selector] function applied to each element in the array.\n
```

```
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@Suppress("INAPPLICABLE_JVM_NAME")\n@kotlin.jvm.JvmName("sumOfULong")\n\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.Inli\nneOnly\npublic inline fun UIntArray.sumOf(selector: (UInt) -> ULong): ULong {\n    var sum: ULong =\n0.toULong()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns\nthe sum of all values produced by [selector] function applied to each element in the array.\n
```

```
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@Suppress("INAPPLICABLE_JVM_NAME")\n@kotlin.jvm.JvmName("sumOfULong")\n\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.Inli\nneOnly\npublic inline fun ULongArray.sumOf(selector: (ULong) -> ULong): ULong {\n    var sum: ULong =\n0.toULong()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n\n/**\n * Returns\nthe sum of all values produced by [selector] function applied to each element in the array.\n
```

```
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@Suppress("INAPPLICABLE_JVM_NAME")\n@kotlin.jvm.JvmName("sumOfULong")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.sumOf(selector: (UByte) -> ULong): ULong {\n    var sum: ULong = 0.toULong()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n
```

```
*\n@SinceKotlin("1.5")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@Suppress("INAPPLICABLE_JVM_NAME")\n@kotlin.jvm.JvmName("sumOfULong")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.sumOf(selector: (UShort) -> ULong): ULong {\n    var sum: ULong = 0.toULong()\n    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic infix fun <R> UIntArray.zip(other: Array<out R>): List<Pair<UInt, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic infix fun <R> ULongArray.zip(other: Array<out R>): List<Pair<ULong, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic infix fun <R> UByteArray.zip(other: Array<out R>): List<Pair<UByte, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic infix fun <R> UShortArray.zip(other: Array<out R>): List<Pair<UShort, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, V> UIntArray.zip(other: Array<out R>, transform: (a: UInt, b: R) -> V): List<V> {\n    val size = minOf(size, other.size)\n    val list = ArrayList<V>(size)\n    for (i in 0 until size) {\n        list.add(transform(this[i], other[i]))\n    }\n    return list\n}\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, V> ULongArray.zip(other: Array<out R>, transform: (a: ULong, b: R) -> V): List<V> {\n    val size = minOf(size, other.size)\n    val list = ArrayList<V>(size)\n    for (i in 0 until size) {\n        list.add(transform(this[i], other[i]))\n    }\n    return list\n}\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n
```

```
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, V> UByteArray.zip(other: Array<out R>, transform: (a: UByte, b: R) -> V): List<V> {\n    val size = minOf(size, other.size)\n    val list = ArrayList<V>(size)\n    for (i in 0 until size) {\n        list.add(transform(this[i], other[i]))\n    }\n    return list\n}\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array\n
```

with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n

```

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, V>
UShortArray.zip(other: Array<out R>, transform: (a: UShort, b: R) -> V): List<V> {\n    val size = minOf(size,
other.size)\n    val list = ArrayList<V>(size)\n    for (i in 0 until size) {\n        list.add(transform(this[i], other[i]))\n    }\n    return list\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` collection and [other] array with
the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterable\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic infix fun <R> UIntArray.zip(other:
Iterable<R>): List<Pair<UInt, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs
built from the elements of `this` collection and [other] array with the same index.\n * The returned list has length of
the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic infix fun <R> ULongArray.zip(other:
Iterable<R>): List<Pair<ULong, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs
built from the elements of `this` collection and [other] array with the same index.\n * The returned list has length of
the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic infix fun <R> UByteArray.zip(other:
Iterable<R>): List<Pair<UByte, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs
built from the elements of `this` collection and [other] array with the same index.\n * The returned list has length of
the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic infix fun <R> UShortArray.zip(other:
Iterable<R>): List<Pair<UShort, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of
values built from the elements of `this` array and the [other] collection with the same index\n * using the provided
[transform] function applied to each pair of elements.\n * The returned list has length of the shortest collection.\n *
\n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, V>
UIntArray.zip(other: Iterable<R>, transform: (a: UInt, b: R) -> V): List<V> {\n    val arraySize = size\n    val list =
ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))\n    var i = 0\n    for (element in other) {\n
if (i >= arraySize) break\n        list.add(transform(this[i++], element))\n    }\n    return list\n}\n\n/**\n * Returns a
list of values built from the elements of `this` array and the [other] collection with the same index\n * using the
provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest
collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, V>
ULongArray.zip(other: Iterable<R>, transform: (a: ULong, b: R) -> V): List<V> {\n    val arraySize = size\n    val
list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))\n    var i = 0\n    for (element in other)
{\n        if (i >= arraySize) break\n        list.add(transform(this[i++], element))\n    }\n    return list\n}\n\n/**\n *
Returns a list of values built from the elements of `this` array and the [other] collection with the same index\n *
using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the
shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, V>
UByteArray.zip(other: Iterable<R>, transform: (a: UByte, b: R) -> V): List<V> {\n    val arraySize = size\n    val
list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))\n    var i = 0\n    for (element in other)
{\n        if (i >= arraySize) break\n        list.add(transform(this[i++], element))\n    }\n    return list\n}\n\n/**\n *
Returns a list of values built from the elements of `this` array and the [other] collection with the same index\n *
using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the
shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, V>

```

```

UShortArray.zip(other: Iterable<R>, transform: (a: UShort, b: R) -> V): List<V> {
    val arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.
The returned list has length of the shortest collection.

@sample
samples.collections.Iterables.Operations.zipIterable

*\/n@SinceKotlin("1.3")n@ExperimentalUnsignedTypesnpublic infix fun UIntArray.zip(other: UIntArray):
List<Pair<UInt, UInt>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.
The returned list has length of the shortest collection.

@sample
samples.collections.Iterables.Operations.zipIterable

*\/n@SinceKotlin("1.3")n@ExperimentalUnsignedTypesnpublic infix fun ULongArray.zip(other: ULongArray):
List<Pair<ULong, ULong>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.
The returned list has length of the shortest collection.

@sample
samples.collections.Iterables.Operations.zipIterable

*\/n@SinceKotlin("1.3")n@ExperimentalUnsignedTypesnpublic infix fun UByteArray.zip(other: UByteArray):
List<Pair<UByte, UByte>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.
The returned list has length of the shortest collection.

@sample
samples.collections.Iterables.Operations.zipIterable

*\/n@SinceKotlin("1.3")n@ExperimentalUnsignedTypesnpublic infix fun UShortArray.zip(other: UShortArray):
List<Pair<UShort, UShort>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

Returns a list of values built from the elements of `this` array and the [other] array with the same index
using the provided [transform] function applied to each pair of elements.
The returned list has length of the shortest array.

@sample
samples.collections.Iterables.Operations.zipIterableWithTransform

*\/n@SinceKotlin("1.3")n@ExperimentalUnsignedTypesn@kotlin.internal.InlineOnlynpublic inline fun <V>
UIntArray.zip(other: UIntArray, transform: (a: UInt, b: UInt) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}

Returns a list of values built from the elements of `this` array and the [other] array with the same index
using the provided [transform] function applied to each pair of elements.
The returned list has length of the shortest array.

@sample
samples.collections.Iterables.Operations.zipIterableWithTransform

*\/n@SinceKotlin("1.3")n@ExperimentalUnsignedTypesn@kotlin.internal.InlineOnlynpublic inline fun <V>
ULongArray.zip(other: ULongArray, transform: (a: ULong, b: ULong) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}

Returns a list of values built from the elements of `this` array and the [other] array with the same index
using the provided [transform] function applied to each pair of elements.
The returned list has length of the shortest array.

@sample
samples.collections.Iterables.Operations.zipIterableWithTransform

*\/n@SinceKotlin("1.3")n@ExperimentalUnsignedTypesn@kotlin.internal.InlineOnlynpublic inline fun <V>
UByteArray.zip(other: UByteArray, transform: (a: UByte, b: UByte) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}

Returns a list of values built from the elements of `this` array and the [other] array with the same index
using the provided [transform] function applied to each pair of elements.
The returned list has length of the shortest array.

@sample
samples.collections.Iterables.Operations.zipIterableWithTransform

*\/n@SinceKotlin("1.3")n@ExperimentalUnsignedTypesn@kotlin.internal.InlineOnlynpublic inline fun <V>
UShortArray.zip(other: UShortArray, transform: (a: UShort, b: UShort) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}

Returns the sum of all elements in the array.

```

```

*\n@kotlin.jvm.JvmName("\sumOfUInt")\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun Array<out UInt>.sum(): UInt {\n    var sum: UInt = 0u\n    for (element in this) {\n        sum += element\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all elements in the array.\n */
*\n@kotlin.jvm.JvmName("\sumOfULong")\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun Array<out ULong>.sum(): ULong {\n    var sum: ULong = 0uL\n    for (element in this) {\n        sum += element\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all elements in the array.\n */
*\n@kotlin.jvm.JvmName("\sumOfUByte")\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun Array<out UByte>.sum(): UInt {\n    var sum: UInt = 0u\n    for (element in this) {\n        sum += element\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all elements in the array.\n */
*\n@kotlin.jvm.JvmName("\sumOfUShort")\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun Array<out UShort>.sum(): UInt {\n    var sum: UInt = 0u\n    for (element in this) {\n        sum += element\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all elements in the array.\n */
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.sum(): UInt {\n    return storage.sum().toUInt()\n}\n\n/**\n * Returns the sum of all elements in the array.\n */
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.sum(): ULong {\n    return storage.sum().toULong()\n}\n\n/**\n * Returns the sum of all elements in the array.\n */
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.sum(): UInt {\n    return sumOf { it.toUInt() }\n}\n\n/**\n * Returns the sum of all elements in the array.\n */
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.sum(): UInt {\n    return sumOf { it.toUInt() }\n}\n\n", "\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */
*\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("UCollectionsKt")\n\npackage kotlin.collections\n\n/\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n/\n\nimport kotlin.random.*\nimport kotlin.ranges.contains\nimport kotlin.ranges.reversed\n\n/**\n * Returns an array of UByte containing all of the elements of this collection.\n */
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun Collection<UByte>.toUByteArray(): UByteArray {\n    val result = UByteArray(size)\n    var index = 0\n    for (element in this) {\n        result[index++] = element\n    }\n    return result\n}\n\n/**\n * Returns an array of UInt containing all of the elements of this collection.\n */
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun Collection<UInt>.toUIntArray(): UIntArray {\n    val result = UIntArray(size)\n    var index = 0\n    for (element in this) {\n        result[index++] = element\n    }\n    return result\n}\n\n/**\n * Returns an array of ULong containing all of the elements of this collection.\n */
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun Collection<ULong>.toULongArray(): ULongArray {\n    val result = ULongArray(size)\n    var index = 0\n    for (element in this) {\n        result[index++] = element\n    }\n    return result\n}\n\n/**\n * Returns an array of UShort containing all of the elements of this collection.\n */
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun Collection<UShort>.toUShortArray(): UShortArray {\n    val result = UShortArray(size)\n    var index = 0\n    for (element in this) {\n        result[index++] = element\n    }\n    return result\n}\n\n/**\n * Returns the sum of all elements in the collection.\n */
*\n@kotlin.jvm.JvmName("\sumOfUInt")\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun Iterable<UInt>.sum(): UInt {\n    var sum: UInt = 0u\n    for (element in this) {\n        sum += element\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all elements in the collection.\n */
*\n@kotlin.jvm.JvmName("\sumOfULong")\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun Iterable<ULong>.sum(): ULong {\n    var sum: ULong = 0uL\n    for (element in this) {\n        sum += element\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all elements in the collection.\n */
*\n@kotlin.jvm.JvmName("\sumOfUByte")\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun Iterable<UByte>.sum(): UInt {\n    var sum: UInt = 0u\n    for (element in this) {\n        sum += element\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all elements in the collection.\n */

```

```

*^@kotlin.jvm.JvmName("\sumOfUShort")^@SinceKotlin("1.5")^@WasExperimental(ExperimentalUnsigned
Types::class)^npublic fun Iterable<UShort>.sum(): UInt {\n  var sum: UInt = 0u\n  for (element in this) {\n
sum += element\n  }\n  return sum\n}\n\n"/^ * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n
*^@n^@file:kotlin.jvm.JvmMultifileClass^@file:kotlin.jvm.JvmName("\UComparisonsKt")^n^npackage
kotlin.comparisons^n/^n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt^n// See:
https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib^n/^nimport kotlin.random.*^n/^n/**^n * Returns the
greater of two values.\n
*^@n^@SinceKotlin("1.5")^n^@WasExperimental(ExperimentalUnsignedTypes::class)^npublic fun maxOf(a: UInt, b:
UInt): UInt {\n  return if (a >= b) a else b\n}\n\n/**^n * Returns the greater of two values.\n
*^@n^@SinceKotlin("1.5")^n^@WasExperimental(ExperimentalUnsignedTypes::class)^npublic fun maxOf(a: ULong,
b: ULong): ULong {\n  return if (a >= b) a else b\n}\n\n/**^n * Returns the greater of two values.\n
*^@n^@SinceKotlin("1.5")^n^@WasExperimental(ExperimentalUnsignedTypes::class)^npublic fun maxOf(a: UByte,
b: UByte): UByte {\n  return if (a >= b) a else b\n}\n\n/**^n * Returns the greater of two values.\n
*^@n^@SinceKotlin("1.5")^n^@WasExperimental(ExperimentalUnsignedTypes::class)^npublic fun maxOf(a: UShort,
b: UShort): UShort {\n  return if (a >= b) a else b\n}\n\n/**^n * Returns the greater of three values.\n
*^@n^@SinceKotlin("1.5")^n^@WasExperimental(ExperimentalUnsignedTypes::class)^n^@kotlin.internal.InlineOnly^
npublic inline fun maxOf(a: UInt, b: UInt, c: UInt): UInt {\n  return maxOf(a, maxOf(b, c))\n}\n\n/**^n * Returns
the greater of three values.\n
*^@n^@SinceKotlin("1.5")^n^@WasExperimental(ExperimentalUnsignedTypes::class)^n^@kotlin.internal.InlineOnly^
npublic inline fun maxOf(a: ULong, b: ULong, c: ULong): ULong {\n  return maxOf(a, maxOf(b, c))\n}\n\n/**^n *
Returns the greater of three values.\n
*^@n^@SinceKotlin("1.5")^n^@WasExperimental(ExperimentalUnsignedTypes::class)^n^@kotlin.internal.InlineOnly^
npublic inline fun maxOf(a: UByte, b: UByte, c: UByte): UByte {\n  return maxOf(a, maxOf(b, c))\n}\n\n/**^n *
Returns the greater of three values.\n
*^@n^@SinceKotlin("1.5")^n^@WasExperimental(ExperimentalUnsignedTypes::class)^n^@kotlin.internal.InlineOnly^
npublic inline fun maxOf(a: UShort, b: UShort, c: UShort): UShort {\n  return maxOf(a, maxOf(b, c))\n}\n\n/**^n
* Returns the greater of the given values.\n
*^@n^@SinceKotlin("1.4")^n^@ExperimentalUnsignedTypes^npublic fun
maxOf(a: UInt, vararg other: UInt): UInt {\n  var max = a\n  for (e in other) max = maxOf(max, e)\n  return
max\n}\n\n/**^n * Returns the greater of the given values.\n
*^@n^@SinceKotlin("1.4")^n^@ExperimentalUnsignedTypes^npublic fun maxOf(a: ULong, vararg other: ULong):
ULong {\n  var max = a\n  for (e in other) max = maxOf(max, e)\n  return max\n}\n\n/**^n * Returns the greater
of the given values.\n
*^@n^@SinceKotlin("1.4")^n^@ExperimentalUnsignedTypes^npublic fun maxOf(a: UByte,
vararg other: UByte): UByte {\n  var max = a\n  for (e in other) max = maxOf(max, e)\n  return max\n}\n\n/**^n
* Returns the greater of the given values.\n
*^@n^@SinceKotlin("1.4")^n^@ExperimentalUnsignedTypes^npublic fun
maxOf(a: UShort, vararg other: UShort): UShort {\n  var max = a\n  for (e in other) max = maxOf(max, e)\n
return max\n}\n\n/**^n * Returns the smaller of two values.\n
*^@n^@SinceKotlin("1.5")^n^@WasExperimental(ExperimentalUnsignedTypes::class)^npublic fun minOf(a: UInt, b:
UInt): UInt {\n  return if (a <= b) a else b\n}\n\n/**^n * Returns the smaller of two values.\n
*^@n^@SinceKotlin("1.5")^n^@WasExperimental(ExperimentalUnsignedTypes::class)^npublic fun minOf(a: ULong,
b: ULong): ULong {\n  return if (a <= b) a else b\n}\n\n/**^n * Returns the smaller of two values.\n
*^@n^@SinceKotlin("1.5")^n^@WasExperimental(ExperimentalUnsignedTypes::class)^npublic fun minOf(a: UByte,
b: UByte): UByte {\n  return if (a <= b) a else b\n}\n\n/**^n * Returns the smaller of two values.\n
*^@n^@SinceKotlin("1.5")^n^@WasExperimental(ExperimentalUnsignedTypes::class)^npublic fun minOf(a: UShort,
b: UShort): UShort {\n  return if (a <= b) a else b\n}\n\n/**^n * Returns the smaller of three values.\n
*^@n^@SinceKotlin("1.5")^n^@WasExperimental(ExperimentalUnsignedTypes::class)^n^@kotlin.internal.InlineOnly^
npublic inline fun minOf(a: UInt, b: UInt, c: UInt): UInt {\n  return minOf(a, minOf(b, c))\n}\n\n/**^n * Returns

```

the smaller of three values.\n

```
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun minOf(a: ULong, b: ULong, c: ULong): ULong {\n    return minOf(a, minOf(b, c))\n}\n\n/**\n * Returns the smaller of three values.\n */
```

```
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun minOf(a: UByte, b: UByte, c: UByte): UByte {\n    return minOf(a, minOf(b, c))\n}\n\n/**\n * Returns the smaller of three values.\n */
```

```
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun minOf(a: UShort, b: UShort, c: UShort): UShort {\n    return minOf(a, minOf(b, c))\n}\n\n/**\n * Returns the smaller of the given values.\n */\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun\nminOf(a: UInt, vararg other: UInt): UInt {\n    var min = a\n    for (e in other) min = minOf(min, e)\n    return\n    min\n}\n\n/**\n * Returns the smaller of the given values.\n */
```

```
*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun minOf(a: ULong, vararg other: ULong):\n    ULong {\n    var min = a\n    for (e in other) min = minOf(min, e)\n    return min\n}\n\n/**\n * Returns the smaller\n * of the given values.\n */\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun minOf(a: UByte,\n    vararg other: UByte): UByte {\n    var min = a\n    for (e in other) min = minOf(min, e)\n    return min\n}\n\n/**\n * Returns the smaller of the given values.\n */\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun\nminOf(a: UShort, vararg other: UShort): UShort {\n    var min = a\n    for (e in other) min = minOf(min, e)\n    return min\n}\n}\n\n"/**\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */
```

```
*\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("URangesKt")\n\npackage
```

```
kotlin.ranges\n\n/\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See:
```

```
https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\nimport kotlin.random.*\n\n/**\n * Returns a\n * random element from this range.\n * \n * @throws IllegalArgumentException if this range is empty.\n */
```

```
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun UIntRange.random(): UInt {\n    return random(Random)\n}\n\n/**\n * Returns a random element\n * from this range.\n * \n * @throws IllegalArgumentException if this range is empty.\n */
```

```
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun ULongRange.random(): ULong {\n    return random(Random)\n}\n\n/**\n * Returns a random\n * element from this range using the specified source of randomness.\n * \n * @throws IllegalArgumentException if\n * this range is empty.\n */\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun\nUIntRange.random(random: Random): UInt {\n    try {\n        return random.nextUInt(this)\n    } catch(e:\n        IllegalArgumentException) {\n        throw NoSuchElementException(e.message)\n    }\n}\n\n/**\n * Returns a\n * random element from this range using the specified source of randomness.\n * \n * @throws\n * IllegalArgumentException if this range is empty.\n */
```

```
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun\nULongRange.random(random: Random): ULong {\n    try {\n        return random.nextULong(this)\n    } catch(e:\n        IllegalArgumentException) {\n        throw NoSuchElementException(e.message)\n    }\n}\n\n/**\n * Returns a\n * random element from this range, or `null` if this range is empty.\n */
```

```
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class,\n    ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun UIntRange.randomOrNull():\n    UInt? {\n    return randomOrNull(Random)\n}\n\n/**\n * Returns a random element from this range, or `null` if this\n * range is empty.\n */\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class,\n    ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun ULongRange.randomOrNull():\n    ULong? {\n    return randomOrNull(Random)\n}\n\n/**\n * Returns a random element from this range using the\n * specified source of randomness, or `null` if this range is empty.\n */
```

```
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class,\n    ExperimentalUnsignedTypes::class)\npublic fun UIntRange.randomOrNull(random: Random): UInt? {\n    if
```

```

(isEmpty())\n    return null\n    return random.nextUInt(this)\n}\n\n/**\n * Returns a random element from this
range using the specified source of randomness, or `null` if this range is empty.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class,
ExperimentalUnsignedTypes::class)\npublic fun ULongRange.randomOrNull(random: Random): ULong? {\n    if
(isEmpty())\n        return null\n        return random.nextULong(this)\n}\n\n/**\n * Returns `true` if this range contains
the specified [element].\n * \n * Always returns `false` if the [element] is `null`.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\n
public inline operator fun UIntRange.contains(element: UInt?): Boolean {\n    return element != null &&
contains(element)\n}\n\n/**\n * Returns `true` if this range contains the specified [element].\n * \n * Always returns
`false` if the [element] is `null`.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\n
public inline operator fun ULongRange.contains(element: ULong?): Boolean {\n    return element != null &&
contains(element)\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic operator fun
UIntRange.contains(value: UByte): Boolean {\n    return contains(value.toUInt())\n}\n\n/**\n * Checks if the
specified [value] belongs to this range.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic operator fun
ULongRange.contains(value: UByte): Boolean {\n    return contains(value.toULong())\n}\n\n/**\n * Checks if the
specified [value] belongs to this range.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic operator fun
ULongRange.contains(value: UInt): Boolean {\n    return contains(value.toULong())\n}\n\n/**\n * Checks if the
specified [value] belongs to this range.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic operator fun
UIntRange.contains(value: ULong): Boolean {\n    return (value shr UInt.SIZE_BITS) == 0uL &&
contains(value.toUInt())\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic operator fun
UIntRange.contains(value: UShort): Boolean {\n    return contains(value.toUInt())\n}\n\n/**\n * Checks if the
specified [value] belongs to this range.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic operator fun
ULongRange.contains(value: UShort): Boolean {\n    return contains(value.toULong())\n}\n\n/**\n * Returns a
progression from this value down to the specified [to] value with the step -1.\n * \n * The [to] value should be less
than or equal to `this` value.\n * If the [to] value is greater than `this` value the returned progression is empty.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic infix fun
UByte.downTo(to: UByte): UIntProgression {\n    return UIntProgression.fromClosedRange(this.toUInt(),
to.toUInt(), -1)\n}\n\n/**\n * Returns a progression from this value down to the specified [to] value with the step -
1.\n * \n * The [to] value should be less than or equal to `this` value.\n * If the [to] value is greater than `this`
value the returned progression is empty.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic infix fun
UInt.downTo(to: UInt): UIntProgression {\n    return UIntProgression.fromClosedRange(this, to, -1)\n}\n\n/**\n *
Returns a progression from this value down to the specified [to] value with the step -1.\n * \n * The [to] value should
be less than or equal to `this` value.\n * If the [to] value is greater than `this` value the returned progression is
empty.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic infix fun
ULong.downTo(to: ULong): ULongProgression {\n    return ULongProgression.fromClosedRange(this, to, -
1L)\n}\n\n/**\n * Returns a progression from this value down to the specified [to] value with the step -1.\n * \n *
The [to] value should be less than or equal to `this` value.\n * If the [to] value is greater than `this` value the
returned progression is empty.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic infix fun
UShort.downTo(to: UShort): UIntProgression {\n    return UIntProgression.fromClosedRange(this.toUInt(),

```

```

to.toUInt(), -1)\n\n/**\n * Returns a progression that goes over the same range in the opposite direction with the
same step.\n * \n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
UIntProgression.reversed(): UIntProgression {\n    return UIntProgression.fromClosedRange(last, first, -
step)\n}\n\n/**\n * Returns a progression that goes over the same range in the opposite direction with the same
step.\n * \n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
ULongProgression.reversed(): ULongProgression {\n    return ULongProgression.fromClosedRange(last, first, -
step)\n}\n\n/**\n * Returns a progression that goes over the same range with the given step.\n
*\n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic infix fun
UIntProgression.step(step: Int): UIntProgression {\n    checkStepIsPositive(step > 0, step)\n    return
UIntProgression.fromClosedRange(first, last, if (this.step > 0) step else -step)\n}\n\n/**\n * Returns a progression
that goes over the same range with the given step.\n
*\n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic infix fun
ULongProgression.step(step: Long): ULongProgression {\n    checkStepIsPositive(step > 0, step)\n    return
ULongProgression.fromClosedRange(first, last, if (this.step > 0) step else -step)\n}\n\n/**\n * Returns a range from
this value up to but excluding the specified [to] value.\n * \n * \n * If the [to] value is less than or equal to `this` value,
then the returned range is empty.\n
*\n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic infix fun
UByte.until(to: UByte): UIntRange {\n    if (to <= UByte.MIN_VALUE) return UIntRange.EMPTY\n    return
this.toUInt() .. (to - 1u).toUInt()\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to]
value.\n * \n * \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n
*\n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic infix fun
UInt.until(to: UInt): UIntRange {\n    if (to <= UInt.MIN_VALUE) return UIntRange.EMPTY\n    return this .. (to -
1u).toUInt()\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * \n * If the
[to] value is less than or equal to `this` value, then the returned range is empty.\n
*\n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic infix fun
ULong.until(to: ULong): ULongRange {\n    if (to <= ULong.MIN_VALUE) return ULongRange.EMPTY\n    return
this .. (to - 1u).toULong()\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to]
value.\n * \n * \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n
*\n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic infix fun
UShort.until(to: UShort): UIntRange {\n    if (to <= UShort.MIN_VALUE) return UIntRange.EMPTY\n    return
this.toUInt() .. (to - 1u).toUInt()\n}\n\n/**\n * Ensures that this value is not less than the specified
[minimumValue].\n * \n * \n * @return this value if it's greater than or equal to the [minimumValue] or the
[minimumValue] otherwise.\n * \n * \n * @sample samples.comparisons.ComparableOps.coerceAtLeastUnsigned\n
*\n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
UInt.coerceAtLeast(minimumValue: UInt): UInt {\n    return if (this < minimumValue) minimumValue else
this\n}\n\n/**\n * Ensures that this value is not less than the specified [minimumValue].\n * \n * \n * @return this value
if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.\n * \n * \n * @sample
samples.comparisons.ComparableOps.coerceAtLeastUnsigned\n
*\n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
ULong.coerceAtLeast(minimumValue: ULong): ULong {\n    return if (this < minimumValue) minimumValue else
this\n}\n\n/**\n * Ensures that this value is not less than the specified [minimumValue].\n * \n * \n * @return this value
if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.\n * \n * \n * @sample
samples.comparisons.ComparableOps.coerceAtLeastUnsigned\n
*\n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
UByte.coerceAtLeast(minimumValue: UByte): UByte {\n    return if (this < minimumValue) minimumValue else
this\n}\n\n/**\n * Ensures that this value is not less than the specified [minimumValue].\n * \n * \n * @return this value
if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.\n * \n * \n * @sample
samples.comparisons.ComparableOps.coerceAtLeastUnsigned\n

```

```

*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
UShort.coerceAtLeast(minimumValue: UShort): UShort {\n    return if (this < minimumValue) minimumValue else
this\n}\n\n/**\n * Ensures that this value is not greater than the specified [maximumValue].\n * \n * @return this
value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise.\n * \n * @sample
samples.comparisons.ComparableOps.coerceAtMostUnsigned\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
UInt.coerceAtMost(maximumValue: UInt): UInt {\n    return if (this > maximumValue) maximumValue else
this\n}\n\n/**\n * Ensures that this value is not greater than the specified [maximumValue].\n * \n * @return this
value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise.\n * \n * @sample
samples.comparisons.ComparableOps.coerceAtMostUnsigned\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
ULong.coerceAtMost(maximumValue: ULong): ULong {\n    return if (this > maximumValue) maximumValue else
this\n}\n\n/**\n * Ensures that this value is not greater than the specified [maximumValue].\n * \n * @return this
value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise.\n * \n * @sample
samples.comparisons.ComparableOps.coerceAtMostUnsigned\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
UByte.coerceAtMost(maximumValue: UByte): UByte {\n    return if (this > maximumValue) maximumValue else
this\n}\n\n/**\n * Ensures that this value is not greater than the specified [maximumValue].\n * \n * @return this
value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise.\n * \n * @sample
samples.comparisons.ComparableOps.coerceAtMostUnsigned\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
UShort.coerceAtMost(maximumValue: UShort): UShort {\n    return if (this > maximumValue) maximumValue
else this\n}\n\n/**\n * Ensures that this value lies in the specified range [minimumValue]..[maximumValue].\n * \n
* @return this value if it's in the range, or [minimumValue] if this value is less than [minimumValue], or
[maximumValue] if this value is greater than [maximumValue].\n * \n * @sample
samples.comparisons.ComparableOps.coerceInUnsigned\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
UInt.coerceIn(minimumValue: UInt, maximumValue: UInt): UInt {\n    if (minimumValue > maximumValue)
throw IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than
minimum $minimumValue.")\n    if (this < minimumValue) return minimumValue\n    if (this > maximumValue)
return maximumValue\n    return this\n}\n\n/**\n * Ensures that this value lies in the specified range
[minimumValue]..[maximumValue].\n * \n * @return this value if it's in the range, or [minimumValue] if this
value is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue].\n * \n *
@sample
samples.comparisons.ComparableOps.coerceInUnsigned\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
ULong.coerceIn(minimumValue: ULong, maximumValue: ULong): ULong {\n    if (minimumValue >
maximumValue) throw IllegalArgumentException("Cannot coerce value to an empty range: maximum
$maximumValue is less than minimum $minimumValue.")\n    if (this < minimumValue) return minimumValue\n
if (this > maximumValue) return maximumValue\n    return this\n}\n\n/**\n * Ensures that this value lies in the
specified range [minimumValue]..[maximumValue].\n * \n * @return this value if it's in the range, or
[minimumValue] if this value is less than [minimumValue], or [maximumValue] if this value is greater than
[maximumValue].\n * \n * @sample
samples.comparisons.ComparableOps.coerceInUnsigned\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
UByte.coerceIn(minimumValue: UByte, maximumValue: UByte): UByte {\n    if (minimumValue >
maximumValue) throw IllegalArgumentException("Cannot coerce value to an empty range: maximum
$maximumValue is less than minimum $minimumValue.")\n    if (this < minimumValue) return minimumValue\n
if (this > maximumValue) return maximumValue\n    return this\n}\n\n/**\n * Ensures that this value lies in the
specified range [minimumValue]..[maximumValue].\n * \n * @return this value if it's in the range, or

```

```

[minimumValue] if this value is less than [minimumValue], or [maximumValue] if this value is greater than
[maximumValue].\n * \n * @sample samples.comparisons.ComparableOps.coerceInUnsigned\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
UShort.coerceIn(minimumValue: UShort, maximumValue: UShort): UShort {\n if (minimumValue >
maximumValue) throw IllegalArgumentException("Cannot coerce value to an empty range: maximum
$maximumValue is less than minimum $minimumValue.")\n if (this < minimumValue) return minimumValue\n
if (this > maximumValue) return maximumValue\n return this\n}\n\n/**\n * Ensures that this value lies in the
specified [range].\n * \n * @return this value if it's in the [range], or `range.start` if this value is less than
`range.start`, or `range.endInclusive` if this value is greater than `range.endInclusive`.\n * \n * @sample
samples.comparisons.ComparableOps.coerceInUnsigned\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
UInt.coerceIn(range: ClosedRange<UInt>): UInt {\n if (range is ClosedFloatingPointRange) {\n return
this.coerceIn<UInt>(range)\n }\n if (range.isEmpty()) throw IllegalArgumentException("Cannot coerce value to
an empty range: $range.")\n return when {\n this < range.start -> range.start\n this > range.endInclusive -
> range.endInclusive\n else -> this\n }\n}\n\n/**\n * Ensures that this value lies in the specified [range].\n * \n
* @return this value if it's in the [range], or `range.start` if this value is less than `range.start`, or
`range.endInclusive` if this value is greater than `range.endInclusive`.\n * \n * @sample
samples.comparisons.ComparableOps.coerceInUnsigned\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
ULong.coerceIn(range: ClosedRange<ULong>): ULong {\n if (range is ClosedFloatingPointRange) {\n return
this.coerceIn<ULong>(range)\n }\n if (range.isEmpty()) throw IllegalArgumentException("Cannot coerce value
to an empty range: $range.")\n return when {\n this < range.start -> range.start\n this >
range.endInclusive -> range.endInclusive\n else -> this\n }\n}\n\n"/\n * Copyright 2010-2021 JetBrains
s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0
license that can be found in the license/LICENSE.txt file.\n
*\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("USequencesKt")\n\npackage
kotlin.sequences\n\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See:
https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\nimport kotlin.random.*\n\n/**\n * Returns the
sum of all elements in the sequence.\n * \n * The operation is _terminal_.\n
*\n@kotlin.jvm.JvmName("sumOfUInt")\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedT
ypes::class)\npublic fun Sequence<UInt>.sum(): UInt {\n var sum: UInt = 0u\n for (element in this) {\n sum
+= element\n }\n return sum\n}\n\n/**\n * Returns the sum of all elements in the sequence.\n * \n * The
operation is _terminal_.\n
*\n@kotlin.jvm.JvmName("sumOfULong")\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsigned
Types::class)\npublic fun Sequence<ULong>.sum(): ULong {\n var sum: ULong = 0uL\n for (element in this)
{\n sum += element\n }\n return sum\n}\n\n/**\n * Returns the sum of all elements in the sequence.\n * \n *
The operation is _terminal_.\n
*\n@kotlin.jvm.JvmName("sumOfUByte")\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsigned
Types::class)\npublic fun Sequence<UByte>.sum(): UInt {\n var sum: UInt = 0u\n for (element in this) {\n
sum += element\n }\n return sum\n}\n\n/**\n * Returns the sum of all elements in the sequence.\n * \n * The
operation is _terminal_.\n
*\n@kotlin.jvm.JvmName("sumOfUShort")\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsigned
Types::class)\npublic fun Sequence<UShort>.sum(): UInt {\n var sum: UInt = 0u\n for (element in this) {\n
sum += element\n }\n return sum\n}\n\n"/\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n *\n\npackage kotlin\n\npublic expect open class Error : Throwable {\n
constructor()\n constructor(message: String?)\n constructor(message: String?, cause: Throwable?)\n
constructor(cause: Throwable?)\n}\n\npublic expect open class Exception : Throwable {\n
constructor()\n

```

```

constructor(message: String?)\n  constructor(message: String?, cause: Throwable?)\n  constructor(cause:
Throwable?)\n}\n\npublic expect open class RuntimeException : Exception {\n  constructor()\n
constructor(message: String?)\n  constructor(message: String?, cause: Throwable?)\n  constructor(cause:
Throwable?)\n}\n\npublic expect open class IllegalArgumentException : RuntimeException {\n  constructor()\n
constructor(message: String?)\n  constructor(message: String?, cause: Throwable?)\n  constructor(cause:
Throwable?)\n}\n\npublic expect open class IllegalStateException : RuntimeException {\n  constructor()\n
constructor(message: String?)\n  constructor(message: String?, cause: Throwable?)\n  constructor(cause:
Throwable?)\n}\n\npublic expect open class IndexOutOfBoundsException : RuntimeException {\n  constructor()\n
constructor(message: String?)\n}\n\npublic expect open class ConcurrentModificationException :
RuntimeException {\n  constructor()\n  constructor(message: String?)\n  @Deprecated(\\"The constructor is not
supported on all platforms and will be removed from kotlin-stdlib-common soon.\", level =
DeprecationLevel.ERROR)\n  constructor(message: String?, cause: Throwable?)\n  @Deprecated(\\"The
constructor is not supported on all platforms and will be removed from kotlin-stdlib-common soon.\", level =
DeprecationLevel.ERROR)\n  constructor(cause: Throwable?)\n}\n\npublic expect open class
UnsupportedOperationException : RuntimeException {\n  constructor()\n  constructor(message: String?)\n
constructor(message: String?, cause: Throwable?)\n  constructor(cause: Throwable?)\n}\n\npublic expect open
class NumberFormatException : IllegalArgumentException {\n  constructor()\n  constructor(message:
String?)\n}\n\npublic expect open class NullPointerException : RuntimeException {\n  constructor()\n
constructor(message: String?)\n}\n\npublic expect open class ClassCastException : RuntimeException {\n
constructor()\n  constructor(message: String?)\n}\n\npublic expect open class AssertionError : Error {\n
constructor()\n  constructor(message: Any?)\n}\n\npublic expect open class NoSuchElementException :
RuntimeException {\n  constructor()\n  constructor(message: String?)\n}\n\n@SinceKotlin(\\"1.3\")\npublic
expect open class ArithmeticException : RuntimeException {\n  constructor()\n  constructor(message:
String?)\n}\n\n@Deprecated(\\"This exception type is not supposed to be thrown or caught in common code and will
be removed from kotlin-stdlib-common soon.\", level = DeprecationLevel.ERROR)\npublic expect open class
NoWhenBranchMatchedException : RuntimeException {\n  constructor()\n  constructor(message: String?)\n
constructor(message: String?, cause: Throwable?)\n  constructor(cause: Throwable?)\n}\n\n@Deprecated(\\"This
exception type is not supposed to be thrown or caught in common code and will be removed from kotlin-stdlib-
common soon.\", level = DeprecationLevel.ERROR)\npublic expect class UninitializedPropertyAccessException :
RuntimeException {\n  constructor()\n  constructor(message: String?)\n  constructor(message: String?, cause:
Throwable?)\n  constructor(cause: Throwable?)\n}\n\n/**\n * Thrown after invocation of a function or property
that was expected to return `Nothing`, but returned something instead.\n
*\n *\n * @SinceKotlin(\\"1.4\")\n * @PublishedApi\n * internal class KotlinNothingValueException : RuntimeException {\n
constructor() : super()\n  constructor(message: String?) : super(message)\n  constructor(message: String?, cause:
Throwable?) : super(message, cause)\n  constructor(cause: Throwable?) : super(cause)\n}\n\n/**\n * Returns the
detailed description of this throwable with its stack trace.\n * The detailed description includes:\n * - the short
description (see [Throwable.toString]) of this throwable;\n * - the complete stack trace;\n * - detailed descriptions
of the exceptions that were [suppressed][suppressedExceptions] in order to deliver this exception;\n * - the detailed
description of each throwable in the [Throwable.cause] chain.\n * @SinceKotlin(\\"1.4\")\n * public expect fun
Throwable.stackTraceToString(): String\n\n/**\n * Prints the [detailed description][Throwable.stackTraceToString]
of this throwable to the standard output or standard error output.\n
*\n *\n * @SinceKotlin(\\"1.4\")\n * @Suppress(\\"EXTENSION_SHADOWED_BY_MEMBER\")\n * public expect fun
Throwable.printStackTrace(): Unit\n\n/**\n * When supported by the platform, adds the specified exception to the
list of exceptions that were\n * suppressed in order to deliver this exception.\n
*\n *\n * @SinceKotlin(\\"1.4\")\n * @Suppress(\\"EXTENSION_SHADOWED_BY_MEMBER\")\n * public expect fun
Throwable.addSuppressed(exception: Throwable)\n\n/**\n * Returns a list of all exceptions that were suppressed in
order to deliver this exception.\n * The list can be empty:\n * - if no exceptions were suppressed;\n * - if the
platform doesn't support suppressed exceptions;\n * - if this [Throwable] instance has disabled the suppression.\n

```

```

*  

@SinceKotlin("1.4")  

public expect val Throwable.suppressedExceptions: List<Throwable>  

"/  

Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.  

Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.  

package  

kotlin.js  

import kotlin.annotation.AnnotationTarget.  

/*  

*/  

/* Gives a declaration (a function, a property or a class) specific name in JavaScript.  

*/  

@Target(CLASS, FUNCTION, PROPERTY, CONSTRUCTOR, PROPERTY_GETTER, PROPERTY_SETTER)  

@OptionalExpectation  

public expect annotation class JsName(val name: String)  

/*  

*/  

/* Marks experimental JS export annotations.  

*/  

/* Note that behavior of these annotations will likely be changed in the future.  

*/  

/* Usages of such annotations will be reported as warnings unless an explicit opt-in with  

the [OptIn] annotation, e.g. `@OptIn(ExperimentalJsExport::class)`,  

or with the -opt-in=kotlin.js.ExperimentalJsExport compiler option is given.  

*/  

@RequiresOptIn(level = RequiresOptIn.Level.WARNING)  

@MustBeDocumented  

@Retention(AnnotationRetention.BINARY)  

@SinceKotlin("1.4")  

public annotation class ExperimentalJsExport  

/*  

*/  

/* Exports top-level declaration on JS platform.  

*/  

/* Compiled module exposes declarations that are marked with this annotation without name mangling.  

*/  

/* This annotation can be applied to either files or top-level declarations.  

*/  

/* It is currently prohibited to export the following kinds of declarations:  

*/  

/* * `expect` declarations  

*/  

/* * inline functions with reified type parameters  

*/  

/* * suspend functions  

*/  

/* * secondary constructors without `@JsName`  

*/  

/* * extension properties  

*/  

/* * enum classes  

*/  

/* * annotation classes  

*/  

/* Signatures of exported declarations must only contain "exportable" types:  

*/  

/* * `dynamic`, `Any`, `String`, `Boolean`, `Byte`, `Short`, `Int`, `Float`,  

*/  

/* * `Double`  

*/  

/* * `BooleanArray`, `ByteArray`, `ShortArray`, `IntArray`, `FloatArray`, `DoubleArray`  

*/  

/* * `Array<exportable-type>`  

*/  

/* * Function types with exportable parameters and return types  

*/  

/* * `external` or `@JsExport` classes and interfaces  

*/  

/* * Nullable counterparts of types above  

*/  

/* * Unit return type. Must not be nullable  

*/  

/* This annotation is experimental, meaning that restrictions mentioned above are subject to change.  

*/  

@ExperimentalJsExport  

@Retention(AnnotationRetention.BINARY)  

@Target(CLASS, PROPERTY, FUNCTION, FILE)  

@SinceKotlin("1.4")  

@OptionalExpectation  

public expect annotation class JsExport()  

"/  

Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.  

Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.  

package kotlin.io  

/*  

*/  

/* Prints the line separator to the standard output stream.  

*/  

public expect fun println()  

/*  

*/  

/* Prints the given [message] and the line separator to the standard output stream.  

*/  

public expect fun println(message: Any?)  

/*  

*/  

/* Prints the given [message] to the standard output stream.  

*/  

public expect fun print(message: Any?)  

/*  

*/  

/* Reads a line of input from the standard input stream and returns it,  

or throws a [RuntimeException] if EOF has already been reached when [readLn] is called.  

*/  

/* LF or CRLF is treated as the line terminator. Line terminator is not included in the returned string.  

*/  

/* Currently this function is not supported in Kotlin/JS and throws [UnsupportedOperationException].  

*/  

@SinceKotlin("1.6")  

public expect fun readLn(): String  

/*  

*/  

/* Reads a line of input from the standard input stream and returns it,  

or return `null` if EOF has already been reached when [readLnOrNull] is called.  

*/  

/* LF or CRLF is treated as the line terminator. Line terminator is not included in the returned string.  

*/  

/* Currently this function is not supported in Kotlin/JS and throws [UnsupportedOperationException].  

*/  

@SinceKotlin("1.6")  

public expect fun readLnOrNull(): String?  

/*  

*/  

internal class ReadAfterEOFException(message: String?) : RuntimeException(message)  

/*  

*/  

internal expect interface Serializable  

"/  

Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.  

Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.  

package kotlin.collections  

import kotlin.internal.PlatformDependent  

/*  

*/  

/* Classes that inherit from this interface can be represented as a sequence of elements that can  

be iterated over.  

*/  

/* @param T the type of element being iterated over. The iterator is covariant in its element type.  

*/  

public interface Iterable<out T> {  

/*  

*/  

/* Returns an iterator over the elements of this object.  

*/  

public operator fun iterator(): Iterator<T>  

}  

/*  

*/  

/* Classes that inherit from this interface can be represented as a sequence of elements that can  

be iterated over and that supports removing elements during iteration.  

*/  

/* @param T the type of element being iterated over. The mutable iterator is invariant in its element type.  

*/  

public interface MutableIterable<out T> : Iterable<T> {  

/*  

*/  

/* Returns an iterator over the elements of this sequence that

```

supports removing elements during iteration.

```

    override fun iterator(): MutableIterator<T>

```

A generic collection of elements. Methods in this interface support only read-only access to the collection; read/write access is supported through the [MutableCollection] interface.

@param E the type of elements contained in the collection. The collection is covariant in its element type.

```

public interface Collection<out E> :
    Iterable<E> {
    // Query Operations
    /**
     * Returns the size of the collection.
     */
    public val size: Int
    /**
     * Returns `true` if the collection is empty (contains no elements), `false` otherwise.
     */
    public fun isEmpty(): Boolean
    /**
     * Checks if the specified element is contained in this collection.
     */
    public operator fun contains(element: @UnsafeVariance E): Boolean
    override fun iterator():
        Iterator<E>
    // Bulk Operations
    /**
     * Checks if all elements in the specified collection are contained in
     * this collection.
     */
    public fun containsAll(elements: Collection<@UnsafeVariance E>): Boolean
}

```

A generic collection of elements that supports adding and removing elements.

@param E the type of elements contained in the collection. The mutable collection is invariant in its element type.

```

public interface
    MutableCollection<E> : Collection<E>, MutableIterable<E> {
    // Query Operations
    override fun iterator():
        MutableIterator<E>
    // Modification Operations
    /**
     * Adds the specified element to the collection.
     *
     * @return `true` if the element has been added, `false` if the collection does not support
     * duplicates and the element is already contained in the collection.
     */
    public fun add(element: E): Boolean
    /**
     * Removes a single instance of the specified element from this collection, if it is present.
     *
     * @return `true` if the element has been successfully removed; `false` if it was not present
     * in the collection.
     */
    public fun remove(element: E): Boolean
    // Bulk Modification Operations
    /**
     * Adds all of the elements of the specified collection to this collection.
     *
     * @return `true` if any of the specified elements was added to the collection, `false` if
     * the collection was not modified.
     */
    public fun addAll(elements: Collection<E>): Boolean
    /**
     * Removes all of this collection's elements that are also contained in the specified
     * collection.
     *
     * @return `true` if any of the specified elements was removed from the collection, `false` if
     * the collection was not modified.
     */
    public fun removeAll(elements: Collection<E>): Boolean
    /**
     * Retains only the elements in this collection that are contained in the specified collection.
     *
     * @return `true` if any element was removed from the collection, `false` if the collection
     * was not modified.
     */
    public fun retainAll(elements: Collection<E>): Boolean
    /**
     * Removes all elements from this collection.
     */
    public fun clear(): Unit
}

```

A generic ordered collection of elements. Methods in this interface support only read-only access to the list; read/write access is supported through the [MutableList] interface.

@param E the type of elements contained in the list. The list is covariant in its element type.

```

public interface
    List<out E> : Collection<E> {
    // Query Operations
    override val size: Int
    override fun isEmpty(): Boolean
    override fun contains(element: @UnsafeVariance E): Boolean
    override fun iterator():
        Iterator<E>
    // Bulk Operations
    override fun containsAll(elements: Collection<@UnsafeVariance E>):
        Boolean
    // Positional Access Operations
    /**
     * Returns the element at the specified index in the list.
     */
    public operator fun get(index: Int): E
    // Search Operations
    /**
     * Returns the index of the first occurrence of the specified element in the list, or -1 if
     * the specified element is not contained in the list.
     */
    public fun indexOf(element: @UnsafeVariance E): Int
    /**
     * Returns the index of the last occurrence of the specified element in the list, or -1 if
     * the specified element is not contained in the list.
     */
    public fun lastIndexOf(element: @UnsafeVariance E): Int
    // List Iterators
    /**
     * Returns a list iterator over the elements in this list (in proper sequence).
     */
    public fun listIterator(): ListIterator<E>
    /**
     * Returns a list iterator over the elements in this list (in proper sequence), starting at
     * the specified [index].
     */
    public fun listIterator(index: Int): ListIterator<E>
    // View
    /**
     * Returns a view of the portion of this list between the specified [fromIndex] (inclusive)
     * and [toIndex] (exclusive). The returned list is backed by this list, so non-structural
     * changes in the returned list are reflected in this list, and vice-versa. Structural
     * changes in the base list make the behavior of the view undefined.
     */
    public fun
        subList(fromIndex: Int, toIndex: Int): List<E>
}

```

A generic ordered collection of elements that supports adding and removing elements.

@param E the type of elements contained in the list. The mutable list is invariant in its element type.

```

public interface MutableList<E> : List<E>, MutableCollection<E> {
    // Modification

```

```

Operations\n /**\n * Adds the specified element to the end of this list.\n *\n * @return `true` because the
list is always modified as the result of this operation.\n *\n override fun add(element: E): Boolean\n\n
override fun remove(element: E): Boolean\n\n // Bulk Modification Operations\n /**\n * Adds all of the
elements of the specified collection to the end of this list.\n *\n * The elements are appended in the order they
appear in the [elements] collection.\n *\n * @return `true` if the list was changed as the result of the
operation.\n *\n override fun addAll(elements: Collection<E>): Boolean\n\n /**\n * Inserts all of the
elements of the specified collection [elements] into this list at the specified [index].\n *\n * @return `true` if the
list was changed as the result of the operation.\n *\n public fun addAll(index: Int, elements: Collection<E>):
Boolean\n\n override fun removeAll(elements: Collection<E>): Boolean\n override fun retainAll(elements:
Collection<E>): Boolean\n override fun clear(): Unit\n\n // Positional Access Operations\n /**\n * Replaces
the element at the specified position in this list with the specified element.\n *\n * @return the element
previously at the specified position.\n *\n public operator fun set(index: Int, element: E): E\n\n /**\n *
Inserts an element into the list at the specified [index].\n *\n public fun add(index: Int, element: E): Unit\n\n
/**\n * Removes an element at the specified [index] from the list.\n *\n * @return the element that has been
removed.\n *\n public fun removeAt(index: Int): E\n\n // List Iterators\n override fun listIterator():
MutableListIterator<E>\n\n override fun listIterator(index: Int): MutableListIterator<E>\n\n // View\n override
fun subList(fromIndex: Int, toIndex: Int): MutableList<E>\n}\n\n/**\n * A generic unordered collection of elements
that does not support duplicate elements.\n * Methods in this interface support only read-only access to the set;\n *
read/write access is supported through the [MutableSet] interface.\n * @param E the type of elements contained in
the set. The set is covariant in its element type.\n */\npublic interface Set<out E> : Collection<E> {\n // Query
Operations\n\n override val size: Int\n override fun isEmpty(): Boolean\n override fun contains(element:
@UnsafeVariance E): Boolean\n override fun iterator(): Iterator<E>\n\n // Bulk Operations\n override fun
containsAll(elements: Collection<@UnsafeVariance E>): Boolean\n}\n\n/**\n * A generic unordered collection of
elements that does not support duplicate elements, and supports\n * adding and removing elements.\n * @param E
the type of elements contained in the set. The mutable set is invariant in its element type.\n */\npublic interface
MutableSet<E> : Set<E>, MutableCollection<E> {\n // Query Operations\n override fun iterator():
MutableIterator<E>\n\n // Modification Operations\n\n /**\n * Adds the specified element to the set.\n *\n
* @return `true` if the element has been added, `false` if the element is already contained in the set.\n *\n
override fun add(element: E): Boolean\n\n override fun remove(element: E): Boolean\n\n // Bulk Modification
Operations\n\n override fun addAll(elements: Collection<E>): Boolean\n override fun removeAll(elements:
Collection<E>): Boolean\n override fun retainAll(elements: Collection<E>): Boolean\n override fun clear():
Unit\n}\n\n/**\n * A collection that holds pairs of objects (keys and values) and supports efficiently retrieving\n *
the value corresponding to each key. Map keys are unique; the map holds only one value for each key.\n * Methods
in this interface support only read-only access to the map; read-write access is supported through\n * the
[MutableMap] interface.\n * @param K the type of map keys. The map is invariant in its key type, as it\n * can
accept key as a parameter (of [containsKey] for example) and return it in [keys] set.\n * @param V the type of map
values. The map is covariant in its value type.\n */\npublic interface Map<K, out V> {\n // Query Operations\n
/**\n * Returns the number of key/value pairs in the map.\n *\n public val size: Int\n\n /**\n * Returns
`true` if the map is empty (contains no elements), `false` otherwise.\n *\n public fun isEmpty(): Boolean\n\n
/**\n * Returns `true` if the map contains the specified [key].\n *\n public fun containsKey(key: K):
Boolean\n\n /**\n * Returns `true` if the map maps one or more keys to the specified [value].\n *\n public
fun containsValue(value: @UnsafeVariance V): Boolean\n\n /**\n * Returns the value corresponding to the
given [key], or `null` if such a key is not present in the map.\n *\n public operator fun get(key: K): V?\n\n
/**\n * Returns the value corresponding to the given [key], or [defaultValue] if such a key is not present in the
map.\n *\n * @since JDK 1.8\n *\n @SinceKotlin("1.1")\n @PlatformDependent\n public fun
getOrDefault(key: K, defaultValue: @UnsafeVariance V): V {\n // See default implementation in JDK
sources\n throw NotImplementedError()\n }\n\n // Views\n /**\n * Returns a read-only [Set] of all keys
in this map.\n *\n public val keys: Set<K>\n\n /**\n * Returns a read-only [Collection] of all values in this

```

```

map. Note that this collection may contain duplicate values.\n    *\n    public val values: Collection<V>\n\n    /**\n    * Returns a read-only [Set] of all key/value pairs in this map.\n    *\n    public val entries: Set<Map.Entry<K,\nV>>\n\n    /**\n    * Represents a key/value pair held by a [Map].\n    *\n    public interface Entry<out K, out V>\n{\n    /**\n    * Returns the key of this key/value pair.\n    *\n    public val key: K\n\n    /**\n    * Returns the value of this key/value pair.\n    *\n    public val value: V\n    }\n\n    /**\n    * A modifiable collection that holds pairs of objects (keys and values) and supports efficiently retrieving\n    * the value corresponding to each key. Map keys are unique; the map holds only one value for each key.\n    * @param K the type of map keys. The map is invariant in its key type.\n    * @param V the type of map values. The mutable map is invariant in its value type.\n    *\n    public interface MutableMap<K, V> : Map<K, V> {\n    // Modification Operations\n    /**\n    * Associates the specified [value] with the specified [key] in the map.\n    *\n    * @return the previous value associated with the key, or `null` if the key was not present in the map.\n    *\n    public fun put(key: K, value: V): V?\n\n    /**\n    * Removes the specified key and its corresponding value from this map.\n    *\n    * @return the previous value associated with the key, or `null` if the key was not present in the map.\n    *\n    public fun remove(key: K): V?\n\n    /**\n    * Removes the entry for the specified key only if it is mapped to the specified value.\n    *\n    * @return true if entry was removed\n    *\n    @SinceKotlin("1.1")\n    @PlatformDependent\n    public fun remove(key: K, value: V): Boolean {\n    // See default implementation in JDK sources\n    return true\n    }\n\n    // Bulk Modification Operations\n    /**\n    * Updates this map with key/value pairs from the specified map [from].\n    *\n    public fun putAll(from: Map<out K, V>): Unit\n\n    /**\n    * Removes all elements from this map.\n    *\n    public fun clear(): Unit\n\n    // Views\n    /**\n    * Returns a [MutableSet] of all keys in this map.\n    *\n    override val keys: MutableSet<K>\n\n    /**\n    * Returns a [MutableCollection] of all values in this map. Note that this collection may contain duplicate values.\n    *\n    override val values: MutableCollection<V>\n\n    /**\n    * Returns a [MutableSet] of all key/value pairs in this map.\n    *\n    override val entries: MutableSet<MutableMap.MutableEntry<K, V>>\n\n    /**\n    * Represents a key/value pair held by a [MutableMap].\n    *\n    public interface MutableEntry<K, V> : Map.Entry<K, V> {\n    /**\n    * Changes the value associated with the key of this entry.\n    *\n    * @return the previous value corresponding to the key.\n    *\n    public fun setValue(newValue: V): V\n    }\n\n    * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n    * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n    *\n    // Auto-generated file. DO NOT EDIT!\n\n    package kotlin.collections\n\n    /** An iterator over a sequence of values of type `Byte`. *\n    public abstract class ByteIterator : Iterator<Byte> {\n    override final fun next() = nextByte()\n\n    /** Returns the next value in the sequence without boxing. *\n    public abstract fun nextByte(): Byte\n    }\n\n    /** An iterator over a sequence of values of type `Char`. *\n    public abstract class CharIterator : Iterator<Char> {\n    override final fun next() = nextChar()\n\n    /** Returns the next value in the sequence without boxing. *\n    public abstract fun nextChar(): Char\n    }\n\n    /** An iterator over a sequence of values of type `Short`. *\n    public abstract class ShortIterator : Iterator<Short> {\n    override final fun next() = nextShort()\n\n    /** Returns the next value in the sequence without boxing. *\n    public abstract fun nextShort(): Short\n    }\n\n    /** An iterator over a sequence of values of type `Int`. *\n    public abstract class IntIterator : Iterator<Int> {\n    override final fun next() = nextInt()\n\n    /** Returns the next value in the sequence without boxing. *\n    public abstract fun nextInt(): Int\n    }\n\n    /** An iterator over a sequence of values of type `Long`. *\n    public abstract class LongIterator : Iterator<Long> {\n    override final fun next() = nextLong()\n\n    /** Returns the next value in the sequence without boxing. *\n    public abstract fun nextLong(): Long\n    }\n\n    /** An iterator over a sequence of values of type `Float`. *\n    public abstract class FloatIterator : Iterator<Float> {\n    override final fun next() = nextFloat()\n\n    /** Returns the next value in the sequence without boxing. *\n    public abstract fun nextFloat(): Float\n    }\n\n    /** An iterator over a sequence of values of type `Double`. *\n    public abstract class DoubleIterator : Iterator<Double> {\n    override final fun next() = nextDouble()\n\n    /** Returns the next value in the sequence without boxing. *\n    public abstract fun nextDouble(): Double\n    }\n\n    /** An iterator over a sequence of values of type `Boolean`. *\n    public abstract class BooleanIterator : Iterator<Boolean> {\n    override final fun next() = nextBoolean()\n\n    /** Returns the next value in the sequence without boxing. *\n    public abstract fun nextBoolean(): Boolean\n    }\n\n    * Copyright 2010-

```

```

2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n *\n// Auto-generated file. DO NOT
EDIT!\n\npackage kotlin.ranges\n\n/**\n * An iterator over a progression of values of type `Char`.\n * @property
step the number by which the value is incremented on each step.\n */\n\ninternal class CharProgressionIterator(first:
Char, last: Char, val step: Int) : CharIterator() {\n    private val finalElement: Int = last.code\n    private var hasNext:
Boolean = if (step > 0) first <= last else first >= last\n    private var next: Int = if (hasNext) first.code else
finalElement\n\n    override fun hasNext(): Boolean = hasNext\n\n    override fun nextChar(): Char {\n        val value
= next\n        if (value == finalElement) {\n            if (!hasNext) throw kotlin.NoSuchElementException()\n
hasNext = false\n        }\n        else {\n            next += step\n        }\n        return value.toChar()\n    }\n}\n\n/**\n *
An iterator over a progression of values of type `Int`.\n * @property step the number by which the value is
incremented on each step.\n */\n\ninternal class IntProgressionIterator(first: Int, last: Int, val step: Int) : IntIterator()
{\n    private val finalElement: Int = last\n    private var hasNext: Boolean = if (step > 0) first <= last else first >=
last\n    private var next: Int = if (hasNext) first else finalElement\n\n    override fun hasNext(): Boolean =
hasNext\n\n    override fun nextInt(): Int {\n        val value = next\n        if (value == finalElement) {\n            if
(!hasNext) throw kotlin.NoSuchElementException()\n            hasNext = false\n        }\n        else {\n            next +=
step\n        }\n        return value\n    }\n}\n\n/**\n * A progression of values of type `Long`.\n *
@property step the number by which the value is incremented on each step.\n */\n\ninternal class
LongProgressionIterator(first: Long, last: Long, val step: Long) : LongIterator() {\n    private val finalElement: Long
= last\n    private var hasNext: Boolean = if (step > 0) first <= last else first >= last\n    private var next: Long = if
(hasNext) first else finalElement\n\n    override fun hasNext(): Boolean = hasNext\n\n    override fun nextLong():
Long {\n        val value = next\n        if (value == finalElement) {\n            if (!hasNext) throw
kotlin.NoSuchElementException()\n            hasNext = false\n        }\n        else {\n            next += step\n        }\n
return value\n    }\n}\n\n", /*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\n// Auto-generated file. DO NOT EDIT!\n\npackage kotlin.ranges\n\nimport
kotlin.internal.getProgressionLastElement\n\n/**\n * A progression of values of type `Char`.\n */\n\npublic open class
CharProgression(\n    internal constructor(\n        start: Char,\n        endInclusive: Char,\n        step: Int\n    ) :
Iterable<Char> {\n    init {\n        if (step == 0) throw kotlin.IllegalArgumentException("Step must be non-zero.")\n        if
(step == Int.MIN_VALUE) throw kotlin.IllegalArgumentException("Step must be greater than Int.MIN_VALUE to avoid overflow on
negation.")\n    }\n\n    /**\n     * The first element in the progression.\n     */\n    public val first: Char = start\n\n    /**\n     * The last element in the progression.\n     */\n    public val last:
Char = getProgressionLastElement(start.code, endInclusive.code, step).toChar()\n\n    /**\n     * The step of the
progression.\n     */\n    public val step: Int = step\n\n    override fun iterator(): CharIterator =
CharProgressionIterator(first, last, step)\n\n    /**\n     * Checks if the progression is empty.\n     */\n    * Progression
with a positive step is empty if its first element is greater than the last element.\n     * Progression with a negative
step is empty if its first element is less than the last element.\n     */\n    public open fun isEmpty(): Boolean = if
(step > 0) first > last else first < last\n\n    override fun equals(other: Any?): Boolean =\n        other is
CharProgression && (isEmpty() && other.isEmpty()) ||\n        first == other.first && last == other.last && step ==
other.step)\n\n    override fun hashCode(): Int =\n        if (isEmpty()) -1 else (31 * (31 * first.code + last.code) +
step)\n\n    override fun toString(): String = if (step > 0) "$first..$last step $step" else "$first downTo $last step $-
step"\n\n    companion object {\n        /**\n         * Creates CharProgression within the specified bounds of a
closed range.\n         */\n        * The progression starts with the [rangeStart] value and goes toward the [rangeEnd]
value not excluding it, with the specified [step].\n         * In order to go backwards the [step] must be negative.\n
*/\n        * [step] must be greater than `Int.MIN_VALUE` and not equal to zero.\n         */\n        public fun
fromClosedRange(rangeStart: Char, rangeEnd: Char, step: Int): CharProgression = CharProgression(rangeStart,
rangeEnd, step)\n    }\n}\n\n/**\n * A progression of values of type `Int`.\n */\n\npublic open class IntProgression(\n
internal constructor(\n        start: Int,\n        endInclusive: Int,\n        step: Int\n    ) : Iterable<Int> {\n    init {\n
        if (step == 0) throw kotlin.IllegalArgumentException("Step must be non-zero.")\n        if (step ==

```

```

Int.MIN_VALUE) throw kotlin.IllegalArgumentException("Step must be greater than Int.MIN_VALUE to avoid
overflow on negation.")\n }\n\n /**\n * The first element in the progression.\n */\n public val first: Int =
start\n\n /**\n * The last element in the progression.\n */\n public val last: Int =
getProgressionLastElement(start, endInclusive, step)\n\n /**\n * The step of the progression.\n */\n public
val step: Int = step\n\n override fun iterator(): IntIterator = IntProgressionIterator(first, last, step)\n\n /**\n *
Checks if the progression is empty.\n */\n * Progression with a positive step is empty if its first element is
greater than the last element.\n * Progression with a negative step is empty if its first element is less than the last
element.\n */\n public open fun isEmpty(): Boolean = if (step > 0) first > last else first < last\n\n override fun
equals(other: Any?): Boolean =\n other is IntProgression && (isEmpty() && other.isEmpty()) ||\n first ==
other.first && last == other.last && step == other.step)\n\n override fun hashCode(): Int =\n if (isEmpty()) -1
else (31 * (31 * first + last) + step)\n\n override fun toString(): String = if (step > 0) \"$first..$last step $step\" else
\"$first downTo $last step $-step\"\n\n companion object {\n /**\n * Creates IntProgression within the
specified bounds of a closed range.\n */\n * The progression starts with the [rangeStart] value and goes
toward the [rangeEnd] value not excluding it, with the specified [step].\n * In order to go backwards the [step]
must be negative.\n */\n * [step] must be greater than `Int.MIN_VALUE` and not equal to zero.\n */\n
public fun fromClosedRange(rangeStart: Int, rangeEnd: Int, step: Int): IntProgression =
IntProgression(rangeStart, rangeEnd, step)\n }\n\n /**\n * A progression of values of type `Long`.\n */\n public
open class LongProgression\n internal constructor\n (start: Long,\n endInclusive: Long,\n
step: Long\n ): Iterable<Long> {\n init {\n if (step == 0L) throw kotlin.IllegalArgumentException("Step
must be non-zero.")\n if (step == Long.MIN_VALUE) throw kotlin.IllegalArgumentException("Step must be
greater than Long.MIN_VALUE to avoid overflow on negation.")\n }\n\n /**\n * The first element in the
progression.\n */\n public val first: Long = start\n\n /**\n * The last element in the progression.\n */\n
public val last: Long = getProgressionLastElement(start, endInclusive, step)\n\n /**\n * The step of the
progression.\n */\n public val step: Long = step\n\n override fun iterator(): LongIterator =
LongProgressionIterator(first, last, step)\n\n /**\n * Checks if the progression is empty.\n */\n *
Progression with a positive step is empty if its first element is greater than the last element.\n * Progression with a
negative step is empty if its first element is less than the last element.\n */\n public open fun isEmpty(): Boolean
= if (step > 0) first > last else first < last\n\n override fun equals(other: Any?): Boolean =\n other is
LongProgression && (isEmpty() && other.isEmpty()) ||\n first == other.first && last == other.last && step ==
other.step)\n\n override fun hashCode(): Int =\n if (isEmpty()) -1 else (31 * (31 * (first xor (first ushr 32)) +
(last xor (last ushr 32))) + (step xor (step ushr 32))).toInt()\n\n override fun toString(): String = if (step > 0)
\"$first..$last step $step\" else \"$first downTo $last step $-step\"\n\n companion object {\n /**\n *
Creates LongProgression within the specified bounds of a closed range.\n */\n * The progression starts
with the [rangeStart] value and goes toward the [rangeEnd] value not excluding it, with the specified [step].\n *
In order to go backwards the [step] must be negative.\n */\n * [step] must be greater than
`Long.MIN_VALUE` and not equal to zero.\n */\n public fun fromClosedRange(rangeStart: Long,
rangeEnd: Long, step: Long): LongProgression = LongProgression(rangeStart, rangeEnd, step)\n }\n\n", /*\n *
Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is
governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage
kotlin.ranges\n\n /**\n * Represents a range of values (for example, numbers or characters).\n * See the [Kotlin
language documentation](https://kotlinlang.org/docs/reference/ranges.html) for more information.\n */\n public
interface ClosedRange<T>: Comparable<T>> {\n /**\n * The minimum value in the range.\n */\n public val
start: T\n\n /**\n * The maximum value in the range (inclusive).\n */\n public val endInclusive: T\n\n
/**\n * Checks whether the specified [value] belongs to the range.\n */\n public operator fun contains(value:
T): Boolean = value >= start && value <= endInclusive\n\n /**\n * Checks whether the range is empty.\n */\n
* The range is empty if its start value is greater than the end value.\n */\n public fun isEmpty(): Boolean = start
> endInclusive\n }\n\n", /*\n * Copyright 2010-2015 JetBrains s.r.o.\n * Licensed under the Apache License,
Version 2.0 (the "License");\n * you may not use this file except in compliance with the License.\n * You may

```

```

obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by
applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS"
BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
License for the specific language governing permissions and limitations under the License.
package
kotlin
/** The type with only one value: the `Unit` object. This type corresponds to the `void` type in Java.
 * \npublic object Unit {
 *     override fun toString() = "kotlin.Unit"
 * }
 * Copyright 2010-2015 JetBrains
 * s.r.o. Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 * Unless required by applicable law or agreed to in writing,
 * software distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES
 * OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language
 * governing permissions and limitations under the License.
 */
package kotlin.annotation
import
kotlin.annotation.AnnotationTarget
/** Contains the list of code elements which are the possible annotation
 * targets
 * \npublic enum class AnnotationTarget {
 *     /** Class, interface or object, annotation class is also
 *     included
 *     \n CLASS,
 *     /** Annotation class only
 *     \n ANNOTATION_CLASS,
 *     /** Generic type
 *     parameter
 *     \n TYPE_PARAMETER,
 *     /** Property
 *     \n PROPERTY,
 *     /** Field, including property's
 *     backing field
 *     \n FIELD,
 *     /** Local variable
 *     \n LOCAL_VARIABLE,
 *     /** Value parameter of a
 *     function or a constructor
 *     \n VALUE_PARAMETER,
 *     /** Constructor only (primary or secondary)
 *     \n
 *     CONSTRUCTOR,
 *     /** Function (constructors are not included)
 *     \n FUNCTION,
 *     /** Property getter only
 *     \n
 *     PROPERTY_GETTER,
 *     /** Property setter only
 *     \n PROPERTY_SETTER,
 *     /** Type usage
 *     \n
 *     TYPE,
 *     /** Any expression
 *     \n EXPRESSION,
 *     /** File
 *     \n FILE,
 *     /** Type alias
 *     \n
 *     @SinceKotlin("1.1")
 *     TYPEALIAS
 * }
 * Contains the list of possible annotation's retentions.
 * Determines how an annotation is stored in binary output.
 * \npublic enum class AnnotationRetention {
 *     /**
 *     Annotation isn't stored in binary output
 *     \n SOURCE,
 *     /** Annotation is stored in binary output, but invisible
 *     for reflection
 *     \n
 *     BINARY,
 *     /** Annotation is stored in binary output and visible for reflection (default
 *     retention)
 *     \n
 *     RUNTIME
 * }
 * This meta-annotation indicates the kinds of code elements which are
 * possible targets of an annotation.
 * If the target meta-annotation is not present on an annotation declaration,
 * the annotation is applicable to the following elements: [CLASS], [PROPERTY], [FIELD],
 * [LOCAL_VARIABLE], [VALUE_PARAMETER], [CONSTRUCTOR], [FUNCTION], [PROPERTY_GETTER],
 * [PROPERTY_SETTER].
 * @property allowedTargets list of allowed annotation targets
 * \n@Target(AnnotationTarget.ANNOTATION_CLASS)\n@MustBeDocumented\npublic annotation class
 * Target(vararg val allowedTargets: AnnotationTarget)
 * This meta-annotation determines whether an
 * annotation is stored in binary output and visible for reflection. By default, both are true.
 * @property value
 * necessary annotation retention (RUNTIME, BINARY or SOURCE)
 * \n@Target(AnnotationTarget.ANNOTATION_CLASS)\npublic annotation class Retention(val value:
 * AnnotationRetention = AnnotationRetention.RUNTIME)
 * This meta-annotation determines that an
 * annotation is applicable twice or more on a single code element
 * \n@Target(AnnotationTarget.ANNOTATION_CLASS)\npublic annotation class Repeatable
 * This
 * meta-annotation determines that an annotation is a part of public API and therefore should be included in the
 * generated
 * documentation for the element to which the annotation is applied.
 * \n@Target(AnnotationTarget.ANNOTATION_CLASS)\npublic annotation class MustBeDocumented
 * Copyright 2010-2016 JetBrains s.r.o. Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License. You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 * Unless required by applicable law or agreed to in writing,
 * software distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES
 * OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language
 * governing permissions and limitations under the License.
 */
package kotlin.internal
/** Specifies
 * that the corresponding type parameter is not used for unsafe operations such as casts or 'is' checks
 * That means

```

it's completely safe to use generic types as argument for such parameter.

```
*\n@Target(AnnotationTarget.TYPE_PARAMETER)\n@Retention(AnnotationRetention.BINARY)\ninternal annotation class PureReifiable\n * Specifies that the corresponding built-in method exists depending on platform.\n * Current implementation for JVM looks whether method with same JVM descriptor exists in the module JDK.\n * For example MutableMap.remove(K, V) available only if corresponding\n * method 'java/util/Map.remove(Ljava/lang/Object;Ljava/lang/Object;)Z' is defined in JDK (i.e. for major versions >= 8)\n*\n@Target(AnnotationTarget.FUNCTION)\n@Retention(AnnotationRetention.BINARY)\ninternal annotation class PlatformDependent\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n *\npackage kotlin.internal\n// a mod b (in arithmetical sense)\nprivate fun mod(a: Int, b: Int): Int {\n    val mod = a % b\n    return if (mod >= 0) mod else mod + b\n}\nprivate fun mod(a: Long, b: Long): Long {\n    val mod = a % b\n    return if (mod >= 0) mod else mod + b\n}\n(a - b) mod c\nprivate fun differenceModulo(a: Int, b: Int, c: Int): Int {\n    return mod(mod(a, c) - mod(b, c), c)\n}\nprivate fun differenceModulo(a: Long, b: Long, c: Long): Long {\n    return mod(mod(a, c) - mod(b, c), c)\n}\n*\n * Calculates the final element of a bounded arithmetic progression, i.e. the last element of the progression which is in the range\n * from [start] to [end] in case of a positive [step], or from [end] to [start] in case of a negative\n * [step].\n * No validation on passed parameters is performed. The given parameters should satisfy the condition:\n * - either `step > 0` and `start <= end`,\n * - or `step < 0` and `start >= end`.\n * @param start first element of the progression\n * @param end ending bound for the progression\n * @param step increment, or difference of successive elements in the progression\n * @return the final element of the progression\n *\n@suppress\n *\n@PublishedApi\ninternal fun getProgressionLastElement(start: Int, end: Int, step: Int): Int = when {\n    step > 0 -> if (start >= end) end else end - differenceModulo(end, start, step)\n    step < 0 -> if (start <= end) end else end + differenceModulo(start, end, -step)\n    else -> throw kotlin.IllegalArgumentException("Step is zero.")\n}\n *\n * Calculates the final element of a bounded arithmetic progression, i.e. the last element of the progression which is in the range\n * from [start] to [end] in case of a positive [step], or from [end] to [start] in case of a negative\n * [step].\n * No validation on passed parameters is performed. The given parameters should satisfy the condition:\n * - either `step > 0` and `start <= end`,\n * - or `step < 0` and `start >= end`.\n * @param start first element of the progression\n * @param end ending bound for the progression\n * @param step increment, or difference of successive elements in the progression\n * @return the final element of the progression\n *\n@suppress\n *\n@PublishedApi\ninternal fun getProgressionLastElement(start: Long, end: Long, step: Long): Long = when {\n    step > 0 -> if (start >= end) end else end - differenceModulo(end, start, step)\n    step < 0 -> if (start <= end) end else end + differenceModulo(start, end, -step)\n    else -> throw\n\nkotlin.IllegalArgumentException("Step is zero.")\n}\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n *\n@JsName("arrayIterator")\ninternal fun arrayIterator(array: dynamic, type: String?) = when (type) {\n    null -> {\n        val arr: Array<dynamic> = array\n        object : Iterator<dynamic> {\n            var index = 0\n            override fun hasNext() = index < arr.size\n            override fun next() = if (index < arr.size) arr[index++] else throw NoSuchElementException("$index")\n        }\n    }\n    "BooleanArray" -> booleanArrayIterator(array)\n    "ByteArray" -> byteArrayIterator(array)\n    "ShortArray" -> shortArrayIterator(array)\n    "CharArray" -> charArrayIterator(array)\n    "IntArray" -> intArrayIterator(array)\n    "LongArray" -> longArrayIterator(array)\n    "FloatArray" -> floatArrayIterator(array)\n    "DoubleArray" -> doubleArrayIterator(array)\n    else -> throw\n\nIllegalStateException("Unsupported type argument for arrayIterator:\n\n$type")\n}\n\n@JsName("booleanArrayIterator")\ninternal fun booleanArrayIterator(array: BooleanArray) = object : BooleanIterator() {\n    var index = 0\n    override fun hasNext() = index < array.size\n    override fun nextBoolean() = if (index < array.size) array[index++] else throw\n\nNoSuchElementException("$index")\n}\n\n@JsName("byteArrayIterator")\ninternal fun byteArrayIterator(array: ByteArray) = object : ByteIterator() {\n    var index = 0\n    override fun hasNext() = index < array.size\n    override
```

```

fun nextByte() = if (index < array.size) array[index++] else throw
NoSuchElementException("$index")\n}\n\n@JsName("shortArrayIterator")\ninternal fun
shortArrayIterator(array: ShortArray) = object : ShortIterator() {\n  var index = 0\n  override fun hasNext() =
index < array.size\n  override fun nextShort() = if (index < array.size) array[index++] else throw
NoSuchElementException("$index")\n}\n\n@JsName("charArrayIterator")\ninternal fun charArrayIterator(array:
CharArray) = object : CharIterator() {\n  var index = 0\n  override fun hasNext() = index < array.size\n  override
fun nextChar() = if (index < array.size) array[index++] else throw
NoSuchElementException("$index")\n}\n\n@JsName("intArrayIterator")\ninternal fun intArrayIterator(array:
IntArray) = object : IntIterator() {\n  var index = 0\n  override fun hasNext() = index < array.size\n  override fun
nextInt() = if (index < array.size) array[index++] else throw
NoSuchElementException("$index")\n}\n\n@JsName("floatArrayIterator")\ninternal fun
floatArrayIterator(array: FloatArray) = object : FloatIterator() {\n  var index = 0\n  override fun hasNext() = index
< array.size\n  override fun nextFloat() = if (index < array.size) array[index++] else throw
NoSuchElementException("$index")\n}\n\n@JsName("doubleArrayIterator")\ninternal fun
doubleArrayIterator(array: DoubleArray) = object : DoubleIterator() {\n  var index = 0\n  override fun hasNext()
= index < array.size\n  override fun nextDouble() = if (index < array.size) array[index++] else throw
NoSuchElementException("$index")\n}\n\n@JsName("longArrayIterator")\ninternal fun longArrayIterator(array:
LongArray) = object : LongIterator() {\n  var index = 0\n  override fun hasNext() = index < array.size\n
override fun nextLong() = if (index < array.size) array[index++] else throw
NoSuchElementException("$index")\n}\n\n@JsName("PropertyMetadata")\ninternal class
PropertyMetadata(@JsName("callableName") val name:
String)\n\n@JsName("noWhenBranchMatched")\ninternal fun noWhenBranchMatched(): Nothing = throw
NoWhenBranchMatchedException()\n\n@JsName("subSequence")\ninternal fun subSequence(c: CharSequence,
startIndex: Int, endIndex: Int): CharSequence {\n  if (c is String) {\n    return c.substring(startIndex, endIndex)\n
} else {\n    return c.asDynamic().`subSequence_vux9f0$`(startIndex, endIndex)\n
}\n}\n\n@JsName("captureStack")\ninternal fun captureStack(@Suppress("UNUSED_PARAMETER")
baseClass: JsClass<in Throwable>, instance: Throwable) {\n  if (js("Error").captureStackTrace) {\n    // Using
uncropped stack traces due to KT-37563.\n    // Precise stack traces are implemented in JS IR compiler and
stdlib\n    js("Error").captureStackTrace(instance);\n  } else {\n    instance.asDynamic().stack = js("new
Error()").stack;\n  }\n}\n\n@JsName("newThrowable")\ninternal fun newThrowable(message: String?, cause:
Throwable?): Throwable {\n  val throwable = js("new Error()")\n  throwable.message = if (jsTypeOf(message)
== "undefined") {\n    if (cause != null) cause.toString() else null\n  } else {\n    message\n  }\n
throwable.cause = cause\n  throwable.name = "Throwable"\n  return
throwable\n}\n\n@JsName("BoxedChar")\ninternal class BoxedChar(val c: Int) : Comparable<Int> {\n  override
fun equals(other: Any?): Boolean {\n    return other is BoxedChar && c == other.c\n  }\n\n  override fun
hashCode(): Int {\n    return c\n  }\n\n  override fun toString(): String {\n    return
js("this.c").unsafeCast<Char>().toString()\n  }\n\n  override fun compareTo(other: Int): Int {\n    return
js("this.c - other").unsafeCast<Int>()\n  }\n\n  @JsName("valueOf")\n  public fun valueOf(): Int {\n
return c\n  }\n}\n\n@kotlin.internal.InlineOnly\ninternal inline fun <T> concat(args: Array<T>): T {\n  val typed
= js("Array")(args.size)\n  for (i in args.indices) {\n    val arr = args[i]\n    if (arr !is Array<*>) {\n
typed[i] = js("[ ]").slice.call(arr)\n    } else {\n    typed[i] = arr\n    }\n  }\n  return
js("[ ]").concat.apply(js("[ ]"), typed);\n}\n\n/** Concat regular Array's and TypedArray's into an Array.\n
*\n@PublishedApi\n@JsName("arrayConcat")\n@Suppress("UNUSED_PARAMETER")\ninternal fun <T>
arrayConcat(a: T, b: T): T {\n  return concat(js("arguments"))\n}\n\n/** Concat primitive arrays. Main use:
prepare vararg arguments.\n * For compatibility with 1.1.0 the arguments may be a mixture of Array's and
TypedArray's.\n * If the first argument is TypedArray (Byte-, Short-, Char-, Int-, Float-, and DoubleArray)
returns a TypedArray, otherwise an Array.\n * If the first argument has the $type$ property (Boolean-, Char-, and
LongArray) copy its value to result.$type$.\n * If the first argument is a regular Array without the $type$ property

```

default to arrayConcat.\n

```
*\n@PublishedApi\n@jsName("primitiveArrayConcat")\n@Suppress("UNUSED_PARAMETER")\ninternal fun <T> primitiveArrayConcat(a: T, b: T): T {\n    val args: Array<T> = js("arguments")\n    if (a is Array<*> && a.asDynamic().`$type$` === undefined) {\n        return concat(args)\n    } else {\n        var size = 0\n        for (i in args.indices) {\n            size += args[i].asDynamic().length as Int\n        }\n        val result = js("new a.constructor(size)")\n        kotlin.copyArrayType(a, result)\n        size = 0\n        for (i in args.indices) {\n            val arr = args[i].asDynamic()\n            for (j in 0 until arr.length) {\n                result[size++] = arr[j]\n            }\n        }\n        return result\n    }\n}\n\n@jsName("booleanArrayOf")\ninternal fun booleanArrayOf() = withType("BooleanArray", js("[].slice.call(arguments)"))\n\n@jsName("charArrayOf") // The arguments have to be slice'd here because of Rhino (see KT-16974)\ninternal fun charArrayOf() = withType("CharArray", js("new Uint16Array([].slice.call(arguments))"))\n\n@jsName("longArrayOf")\ninternal fun longArrayOf() = withType("LongArray", js("[].slice.call(arguments)"))\n\n@jsName("withType")\n@kotlin.internal.InlineOnly\ninternal inline fun withType(type: String, array: dynamic): dynamic {\n    array.`$type$` = type\n    return array\n}\n/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.js\n\n/**\n * Function corresponding to JavaScript's `typeof` operator\n */\n\n@kotlin.internal.InlineOnly\n@Suppress("UNUSED_PARAMETER")\npublic inline fun jsTypeOf(a: Any?): String = js("typeof a")\n/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n@file:Suppress("UNUSED_PARAMETER", "NOTHING_TO_INLINE")\npackage kotlin\n\n/**\n * Returns an empty array of the specified type [T].\n */\n\npublic inline fun <T> emptyArray(): Array<T> = js("[]")\n\n@library\npublic fun <T> arrayOf(vararg elements: T): Array<T> = definedExternally\n\n@library\npublic fun doubleArrayOf(vararg elements: Double): DoubleArray = definedExternally\n\n@library\npublic fun floatArrayOf(vararg elements: Float): FloatArray = definedExternally\n\n@library\npublic fun longArrayOf(vararg elements: Long): LongArray = definedExternally\n\n@library\npublic fun intArrayOf(vararg elements: Int): IntArray = definedExternally\n\n@library\npublic fun charArrayOf(vararg elements: Char): CharArray = definedExternally\n\n@library\npublic fun shortArrayOf(vararg elements: Short): ShortArray = definedExternally\n\n@library\npublic fun byteArrayOf(vararg elements: Byte): ByteArray = definedExternally\n\n@library\npublic fun booleanArrayOf(vararg elements: Boolean): BooleanArray = definedExternally\n\n/**\n * Creates a new instance of the [Lazy] that uses the specified initialization function [initializer].\n */\n\npublic actual fun <T> lazy(initializer: () -> T): Lazy<T> = UnsafeLazyImpl(initializer)\n\n/**\n * Creates a new instance of the [Lazy] that uses the specified initialization function [initializer].\n * The [mode] parameter is ignored.\n */\n\npublic actual fun <T> lazy(mode: LazyThreadSafetyMode, initializer: () -> T): Lazy<T> = UnsafeLazyImpl(initializer)\n\n/**\n * Creates a new instance of the [Lazy] that uses the specified initialization function [initializer].\n * The [lock] parameter is ignored.\n */\n\npublic actual fun <T> lazy(lock: Any?, initializer: () -> T): Lazy<T> = UnsafeLazyImpl(initializer)\n\ninternal fun fillFrom(src: dynamic, dst: dynamic): dynamic {\n    val srcLen: Int = src.length\n    val dstLen: Int = dst.length\n    var index: Int = 0\n    while (index < srcLen && index < dstLen) dst[index] = src[index++]\n    return dst\n}\n\ninternal fun arrayCopyResize(source: dynamic, newSize: Int, defaultValue: Any?): dynamic {\n    val result = source.slice(0, newSize)\n    copyArrayType(source, result)\n    var index: Int = source.length\n    if (newSize > index) {\n        result.length = newSize\n        while (index < newSize) result[index++] = defaultValue\n    }\n    return result\n}\n\ninternal fun <T> arrayPlusCollection(array: dynamic, collection: Collection<T>): dynamic {\n    val result = array.slice()\n    result.length += collection.size\n    copyArrayType(array, result)\n    var index: Int = array.length\n    for (element in collection) result[index++] = element\n    return result\n}\n\ninternal fun <T> fillFromCollection(dst: dynamic, startIndex: Int, collection: Collection<T>): dynamic {\n    var index = startIndex\n    for (element in collection) dst[index++] = element\n    return dst\n}\n\ninternal inline fun copyArrayType(from: dynamic, to: dynamic) {\n    if
```

```

(from.`$type$` != undefined) {\n    to.`$type$` = from.`$type$`\n    }\n}\n\ninternal inline fun jsIsType(obj:
dynamic, jsClass: dynamic) = js(\`"Kotlin"\`).isType(obj, jsClass)", /*\n * Copyright 2010-2021 JetBrains s.r.o. and
Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin\n\n/**\n * Creates a Char with the specified
[code].\n */\n * @sample samples.text.Chars.charFromCode\n
*\n\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
actual inline fun Char(code: UShort): Char {\n    return code.toInt().toChar()\n}\n", /*\n * Copyright 2010-2018
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.coroutines\n\nimport
kotlin.coroutines.intrinsics.COROUTINE_SUSPENDED\n\n@SinceKotlin("1.3")\n@JsName("CoroutineImpl")\n\ninternal abstract class CoroutineImpl(private val resultContinuation: Continuation<Any?>) : Continuation<Any?>
{\n    protected var state = 0\n    protected var exceptionState = 0\n    protected var result: Any? = null\n    protected
var exception: Throwable? = null\n    protected var finallyPath: Array<Int>? = null\n\n    public override val context:
CoroutineContext = resultContinuation.context\n\n    private var intercepted_: Continuation<Any?>? = null\n\n    public fun intercepted(): Continuation<Any?> =\n        intercepted_\n        ?:\n    (context[ContinuationInterceptor]?.interceptContinuation(this) ?: this)\n        .also { intercepted_ = it }\n\n    override fun resumeWith(result: Result<Any?>) {\n        var current = this\n        var currentResult: Any? =
result.getOrNull()\n        var currentException: Throwable? = result.exceptionOrNull()\n        // This loop unrolls
recursion in current.resumeWith(param) to make saner and shorter stack traces on resume\n        while (true) {\n
            with(current) {\n                val completion = resultContinuation\n                // Set result and exception fields in
the current continuation\n                if (currentException == null) {\n                    this.result = currentResult\n
                } else {\n                    state = exceptionState\n                    exception = currentException\n                }\n
            }\n            try {\n                val outcome = doResume()\n                if (outcome === COROUTINE_SUSPENDED)\n            }\n            return\n                currentResult = outcome\n                currentException = null\n            } catch (exception:
dynamic) { // Catch all exceptions\n                currentResult = null\n                currentException =
exception.unsafeCast<Throwable>()\n            }\n            releaseIntercepted() // this state machine instance is
terminating\n            if (completion is CoroutineImpl) {\n                // unrolling recursion via loop\n                current = completion\n            } else {\n                // top-level completion reached -- invoke and return\n                currentException?.let {\n                    completion.resumeWithException(it)\n                } ?:\n                completion.resume(currentResult)\n            }\n            return\n        }\n        }\n        }\n        }\n        }\n        }\n        private fun
releaseIntercepted() {\n            val intercepted = intercepted_\n            if (intercepted != null && intercepted !== this) {\n                context[ContinuationInterceptor]!!.releaseInterceptedContinuation(intercepted)\n            }\n            this.intercepted_
= CompletedContinuation // just in case\n        }\n        protected abstract fun doResume(): Any?\n}\n\ninternal object
CompletedContinuation : Continuation<Any?> {\n    override val context: CoroutineContext\n        get() =
error(\`"This continuation is already complete"\`)\n\n    override fun resumeWith(result: Result<Any?>) {\n        error(\`"This continuation is already complete"\`)\n        }\n\n    override fun toString(): String = \`"This continuation is
already complete"\`)\n}\n", /*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\n@file:Suppress("UNCHECKED_CAST",
"RedundantVisibilityModifier")\n\npackage kotlin\n\nimport kotlin.contracts.*\nimport
kotlin.internal.InlineOnly\nimport kotlin.jvm.JvmField\nimport kotlin.jvm.JvmInline\nimport
kotlin.jvm.JvmName\n\n/**\n * A discriminated union that encapsulates a successful outcome with a value of type
[T]\n * or a failure with an arbitrary [Throwable] exception.\n */\n\n@SinceKotlin("1.3")\n@JvmInline\npublic
value class Result<out T> @PublishedApi internal constructor(\n    @PublishedApi\n    internal val value: Any?\n) :
Serializable {\n    // discovery\n\n    /**\n     * Returns `true` if this instance represents a successful outcome.\n     *
In this case [isFailure] returns `false`.\n     */\n    public val isSuccess: Boolean\n        get() = value !is Failure\n\n    /**\n     * Returns `true` if this instance represents a failed outcome.\n     *
In this case [isSuccess] returns `false`.\n     */\n    public val isFailure: Boolean\n        get() = value is Failure\n\n    // value & exception retrieval\n\n    /**\n     * Returns the

```

```

encapsulated value if this instance represents [success][Result.isSuccess] or `null` if it is
[failure][Result.isFailure].
 * This function is a shorthand for `getOrElse { null }` (see [getOrElse]) or
 * `fold(onSuccess = { it }, onFailure = { null })` (see [fold]).
 @InlineOnly
 public inline fun
 getOrNull(): T? =
     when {
         isFailure -> null
         else -> value as T
     }
 /**
 * Returns the encapsulated [Throwable] exception if this instance represents [failure][isFailure] or `null` if it is
 [success][isSuccess].
 * This function is a shorthand for `fold(onSuccess = { null }, onFailure = { it })`
 (see [fold]).
 @ public fun exceptionOrNull(): Throwable? =
     when (value) {
         is Failure ->
 value.exception
         else -> null
     }
 /**
 * Returns a string `Success(v)` if this instance represents
 [success][Result.isSuccess] where `v` is a string representation of the value or a string `Failure(x)` if it
 is [failure][isFailure] where `x` is a string representation of the exception.
 @ public override fun toString():
 String =
     when (value) {
         is Failure -> value.toString() // "Failure($exception)"
         else ->
 "Success($value)"
     }
 // companion with constructors
 /**
 * Companion object for [Result]
 class that contains its constructor functions
 * [success] and [failure].
 @ public companion object {
 /**
 * Returns an instance that encapsulates the given [value] as successful value.
 @
 @Suppress("INAPPLICABLE_JVM_NAME")
 @InlineOnly
 @JvmName("success")
 public
 inline fun <T> success(value: T): Result<T> =
     Result(value)
 /**
 * Returns an instance that
 encapsulates the given [Throwable] [exception] as failure.
 @
 @Suppress("INAPPLICABLE_JVM_NAME")
 @InlineOnly
 @JvmName("failure")
 public
 inline fun <T> failure(exception: Throwable): Result<T> =
     Result(createFailure(exception))
 }
 internal class Failure(
 @JvmField
 val exception: Throwable
 ): Serializable {
 override fun
 equals(other: Any?): Boolean = other is Failure && exception == other.exception
 override fun hashCode():
 Int = exception.hashCode()
 override fun toString(): String = "Failure($exception)"
 }
 /**
 * Creates an instance of internal marker [Result.Failure] class to
 * make sure that this class is not exposed in ABI.
 @PublishedApi
 @SinceKotlin("1.3")
 internal fun createFailure(exception: Throwable): Any =
 Result.Failure(exception)
 /**
 * Throws exception if the result is failure. This internal function minimizes
 * inlined bytecode for [getOrThrow] and makes sure that in the future we can
 * add some exception-augmenting
 logic here (if needed).
 @PublishedApi
 @SinceKotlin("1.3")
 internal fun Result<*>.throwOnFailure() {
 if (value is Result.Failure) throw value.exception
 }
 /**
 * Calls the specified function [block] and returns its
 encapsulated result if invocation was successful,
 * catching any [Throwable] exception that was thrown from the
 [block] function execution and encapsulating it as a failure.
 @InlineOnly
 @SinceKotlin("1.3")
 public
 inline fun <R> runCatching(block: () -> R): Result<R> {
 return try {
 Result.success(block())
 } catch
 (e: Throwable) {
 Result.failure(e)
 }
 }
 /**
 * Calls the specified function [block] with `this` value as
 its receiver and returns its encapsulated result if invocation was successful,
 * catching any [Throwable] exception
 that was thrown from the [block] function execution and encapsulating it as a failure.
 @InlineOnly
 @SinceKotlin("1.3")
 public inline fun <T, R> T.runCatching(block: T.() -> R): Result<R> {
 return try {
 Result.success(block())
 } catch (e: Throwable) {
 Result.failure(e)
 }
 }
 /**
 * Returns the encapsulated value if this instance represents [success][Result.isSuccess] or
 throws the encapsulated [Throwable] exception
 * if it is [failure][Result.isFailure].
 * This function is a
 shorthand for `getOrElse { throw it }` (see [getOrElse]).
 @InlineOnly
 @SinceKotlin("1.3")
 public inline
 fun <T> Result<T>.getOrThrow(): T {
 throwOnFailure()
 return value as T
 }
 /**
 * Returns the
 encapsulated value if this instance represents [success][Result.isSuccess] or the
 * result of [onFailure] function for
 the encapsulated [Throwable] exception if it is [failure][Result.isFailure].
 * Note, that this function rethrows
 any [Throwable] exception thrown by [onFailure] function.
 * This function is a shorthand for `fold(onSuccess
 = { it }, onFailure = onFailure)` (see [fold]).
 @InlineOnly
 @SinceKotlin("1.3")
 public inline fun <R, T :
 R> Result<T>.getOrElse(onFailure: (exception: Throwable) -> R): R {
 contract {
 callsInPlace(onFailure,
 InvocationKind.AT_MOST_ONCE)
 }
 return when (val exception = exceptionOrNull()) {
 null ->
 value as T
 else -> onFailure(exception)
 }
 }
 /**
 * Returns the encapsulated value if this instance
 represents [success][Result.isSuccess] or the
 * [defaultValue] if it is [failure][Result.isFailure].
 * This

```

function is a shorthand for `getOrElse { defaultValue }` (see `getOrElse`).

```

*InlineOnly@SinceKotlin("1.3")public inline fun <R, T : R> Result<T>.getOrElse(defaultValue: R):
R {
    if (isFailure) return defaultValue
    return value as T
}

```

* Returns the result of `onSuccess` for the encapsulated value if this instance represents `success` or the result of `onFailure` function for the encapsulated `Throwable` exception if it is `failure`. Note, that this function rethrows any `Throwable` exception thrown by `onSuccess` or by `onFailure` function.

```

*InlineOnly@SinceKotlin("1.3")public inline fun <R, T> Result<T>.fold(
    onSuccess: (value: T) -> R,
    onFailure: (exception: Throwable) -> R): R {
    contract {
        callsInPlace(onSuccess,
            InvocationKind.AT_MOST_ONCE)
        callsInPlace(onFailure, InvocationKind.AT_MOST_ONCE)
    }
    return when (val exception = exceptionOrNull()) {
        null -> onSuccess(value as T)
        else ->
            onFailure(exception)
    }
}

```

* Returns the encapsulated result of the given `transform` function applied to the encapsulated value if this instance represents `success` or the original encapsulated `Throwable` exception if it is `failure`. Note, that this function rethrows any `Throwable` exception thrown by `transform` function. See `mapCatching` for an alternative that encapsulates exceptions.

```

*InlineOnly@SinceKotlin("1.3")public inline fun <R, T>
Result<T>.map(transform: (value: T) -> R): Result<R> {
    contract {
        callsInPlace(transform,
            InvocationKind.AT_MOST_ONCE)
    }
    return when {
        isSuccess -> Result.success(transform(value as T))
        else -> Result(value)
    }
}

```

* Returns the encapsulated result of the given `transform` function applied to the encapsulated value if this instance represents `success` or the original encapsulated `Throwable` exception if it is `failure`. This function catches any `Throwable` exception thrown by `transform` function and encapsulates it as a failure. See `map` for an alternative that rethrows exceptions from `transform` function.

```

*InlineOnly@SinceKotlin("1.3")public inline fun <R,
T> Result<T>.mapCatching(transform: (value: T) -> R): Result<R> {
    return when {
        isSuccess ->
            runCatching { transform(value as T) }
        else -> Result(value)
    }
}

```

* Returns the encapsulated result of the given `transform` function applied to the encapsulated `Throwable` exception if this instance represents `failure` or the original encapsulated value if it is `success`. Note, that this function rethrows any `Throwable` exception thrown by `transform` function. See `recoverCatching` for an alternative that encapsulates exceptions.

```

*InlineOnly@SinceKotlin("1.3")public inline fun <R, T : R> Result<T>.recover(
    transform: (exception:
        Throwable) -> R): Result<R> {
    contract {
        callsInPlace(transform, InvocationKind.AT_MOST_ONCE)
    }
    return when (val exception = exceptionOrNull()) {
        null -> this
        else ->
            Result.success(transform(exception))
    }
}

```

* Returns the encapsulated result of the given `transform` function applied to the encapsulated `Throwable` exception if this instance represents `failure` or the original encapsulated value if it is `success`. This function catches any `Throwable` exception thrown by `transform` function and encapsulates it as a failure. See `recover` for an alternative that rethrows exceptions.

```

*InlineOnly@SinceKotlin("1.3")public inline fun <R, T : R>
Result<T>.recoverCatching(transform: (exception: Throwable) -> R): Result<R> {
    return when (val exception =
        exceptionOrNull()) {
        null -> this
        else -> runCatching {
            transform(exception)
        }
    }
}

```

* Performs the given `action` on the encapsulated `Throwable` exception if this instance represents `failure`. Returns the original `Result` unchanged.

```

*InlineOnly@SinceKotlin("1.3")public inline fun <T> Result<T>.onFailure(
    action: (exception:
        Throwable) -> Unit): Result<T> {
    contract {
        callsInPlace(action, InvocationKind.AT_MOST_ONCE)
    }
    exceptionOrNull()?.let { action(it) }
    return this
}

```

* Performs the given `action` on the encapsulated value if this instance represents `success`. Returns the original `Result` unchanged.

```

*InlineOnly@SinceKotlin("1.3")public inline fun <T> Result<T>.onSuccess(
    action: (value:
        T) -> Unit): Result<T> {
    contract {
        callsInPlace(action, InvocationKind.AT_MOST_ONCE)
    }
    if (isSuccess) action(value as T)
    return this
}

```

 * Copyright 2010-2020 JetBrains
 s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0

license that can be found in the license/LICENSE.txt file.

```

package kotlin.coroutines
import kotlin.contracts.*
import kotlin.coroutines.intrinsics.*
import kotlin.internal.InlineOnly
/**
 * Interface representing a continuation after a suspension point that returns a value of type `T`.
 */
@SinceKotlin("1.3")
public interface Continuation<in T> {
    /**
     * The context of the coroutine that corresponds to this continuation.
     */
    public val context: CoroutineContext
    /**
     * Resumes the execution of the corresponding coroutine passing a successful or failed [result] as the
     * return value of the last suspension point.
     */
    public fun resumeWith(result: Result<T>): Unit
}
/**
 * Classes and interfaces marked with this annotation are restricted when used as receivers for extension
 * `suspend` functions. These `suspend` extensions can only invoke other member or extension
 * `suspend` functions on this particular receiver and are restricted from calling arbitrary
 * suspension functions.
 */
@SinceKotlin("1.3")
@Target(AnnotationTarget.CLASS)
@Retention(AnnotationRetention.BINARY)
public annotation class RestrictsSuspension
/**
 * Resumes the execution of the corresponding coroutine passing [value] as the return value
 * of the last suspension point.
 */
@SinceKotlin("1.3")
@InlineOnly
public inline fun <T> Continuation<T>.resume(value: T): Unit =
    resumeWith(Result.success(value))
/**
 * Resumes the execution of the corresponding coroutine so that the [exception] is re-thrown
 * right after the last suspension point.
 */
@SinceKotlin("1.3")
@InlineOnly
public inline fun <T> Continuation<T>.resumeWithException(exception: Throwable): Unit =
    resumeWith(Result.failure(exception))
/**
 * Creates a [Continuation] instance with the given [context] and implementation of [resumeWith]
 * method.
 */
@SinceKotlin("1.3")
@InlineOnly
public inline fun <T> Continuation(
    context: CoroutineContext,
    crossinline resumeWith: (Result<T>) -> Unit): Continuation<T> =
    object : Continuation<T> {
        override val context: CoroutineContext
            get() = context
        override fun resumeWith(result: Result<T>) =
            resumeWith(result)
    }
/**
 * Creates a coroutine without a receiver and with result type [T].
 * This function creates a new, fresh instance of suspendable computation every time it is
 * invoked.
 * To start executing the created coroutine, invoke `resume(Unit)` on the returned
 * [Continuation] instance.
 * The [completion] continuation is invoked when the coroutine completes with a result or an
 * exception.
 * Subsequent invocation of any resume function on the resulting continuation will produce an
 * [IllegalStateException].
 */
@SinceKotlin("1.3")
@Suppress("UNCHECKED_CAST")
public fun <T> (suspend () -> T).createCoroutine(
    completion: Continuation<T>): Continuation<Unit> =
    SafeContinuation(createCoroutineUnintercepted(completion).intercepted(),
        COROUTINE_SUSPENDED)
/**
 * Creates a coroutine with receiver type [R] and result type [T].
 * This function creates a new, fresh instance of suspendable computation every time it is
 * invoked.
 * To start executing the created coroutine, invoke `resume(Unit)` on the returned
 * [Continuation] instance.
 * The [completion] continuation is invoked when the coroutine completes with a result or an
 * exception.
 * Subsequent invocation of any resume function on the resulting continuation will produce an
 * [IllegalStateException].
 */
@SinceKotlin("1.3")
@Suppress("UNCHECKED_CAST")
public fun <R, T> (suspend R.() -> T).createCoroutine(
    receiver: R,
    completion: Continuation<T>): Continuation<Unit> =
    SafeContinuation(createCoroutineUnintercepted(receiver, completion).intercepted(),
        COROUTINE_SUSPENDED)
/**
 * Starts a coroutine without a receiver and with result type [T].
 * This function creates and starts a new, fresh instance of suspendable computation every
 * time it is invoked.
 * The [completion] continuation is invoked when the coroutine completes with a result or an
 * exception.
 */
@SinceKotlin("1.3")
@Suppress("UNCHECKED_CAST")
public fun <T> (suspend () -> T).startCoroutine(
    completion: Continuation<T>) {
    createCoroutineUnintercepted(completion).intercepted().resume(Unit)
}
/**
 * Starts a coroutine with receiver type [R] and result type [T].
 * This function creates and starts a new, fresh instance of suspendable computation every
 * time it is invoked.
 * The [completion] continuation is invoked when the coroutine completes with a result or an
 * exception.
 */
@SinceKotlin("1.3")
@Suppress("UNCHECKED_CAST")
public fun <R, T> (suspend R.() -> T).startCoroutine(
    receiver: R,
    completion: Continuation<T>) {
    createCoroutineUnintercepted(receiver, completion).intercepted().resume(Unit)
}
/**
 * Obtains the current
 */

```

continuation instance inside suspend functions and suspends\n * the currently running coroutine.\n *\n * In this function both [Continuation.resume] and [Continuation.resumeWithException] can be used either synchronously in\n * the same stack-frame where the suspension function is run or asynchronously later in the same thread or\n * from a different thread of execution. Subsequent invocation of any resume function will produce an [IllegalStateException].\n *\n @SinceKotlin("1.3")\n @InlineOnly\n public suspend inline fun <T> suspendCoroutine(crossinline block: (Continuation<T>) -> Unit): T {\n contract { callsInPlace(block, InvocationKind.EXACTLY_ONCE) }\n return suspendCoroutineUninterceptedOrReturn { c: Continuation<T> -> {\n val safe = SafeContinuation(c.intercepted())\n block(safe)\n safe.getOrThrow()\n } }\n }\n /**\n Returns the context of the current coroutine.\n *\n @SinceKotlin("1.3")\n @Suppress("WRONG_MODIFIER_TARGET")\n @InlineOnly\n public suspend inline fun CoroutineContext.get() {\n throw NotImplementedError("Implemented as intrinsic")\n }\n /**\n Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n *\n Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n *\n @package kotlin.coroutines.intrinsics\n @nimport kotlin.coroutines.*\n @nimport kotlin.internal.InlineOnly\n /**\n Starts an unintercepted coroutine without a receiver and with result type [T] and executes it until its first suspension.\n *\n Returns the result of the coroutine or throws its exception if it does not suspend or [COROUTINE_SUSPENDED] if it suspends.\n *\n In the latter case, the [completion] continuation is invoked when the coroutine completes with a result or an exception.\n *\n The coroutine is started directly in the invoker's thread without going through the [ContinuationInterceptor] that might\n * be present in the completion's [CoroutineContext]. It is the invoker's responsibility to ensure that a proper invocation\n * context is established.\n *\n This function is designed to be used from inside of [suspendCoroutineUninterceptedOrReturn] to resume the execution of the suspended\n * coroutine using a reference to the suspending function.\n *\n @SinceKotlin("1.3")\n @InlineOnly\n public actual inline fun <T> (suspend () -> T).startCoroutineUninterceptedOrReturn(completion: Continuation<T>): Any? = this.asDynamic()(completion, false)\n /**\n Starts an unintercepted coroutine with receiver type [R] and result type [T] and executes it until its first suspension.\n *\n Returns the result of the coroutine or throws its exception if it does not suspend or [COROUTINE_SUSPENDED] if it suspends.\n *\n In the latter case, the [completion] continuation is invoked when the coroutine completes with a result or an exception.\n *\n The coroutine is started directly in the invoker's thread without going through the [ContinuationInterceptor] that might\n * be present in the completion's [CoroutineContext]. It is the invoker's responsibility to ensure that a proper invocation\n * context is established.\n *\n This function is designed to be used from inside of [suspendCoroutineUninterceptedOrReturn] to resume the execution of the suspended\n * coroutine using a reference to the suspending function.\n *\n @SinceKotlin("1.3")\n @InlineOnly\n public actual inline fun <R, T> (suspend R.() -> T).startCoroutineUninterceptedOrReturn(receiver: R, completion: Continuation<T>): Any? = this.asDynamic()(receiver, completion, false)\n @InlineOnly\n internal actual inline fun <R, P, T> (suspend R.(P) -> T).startCoroutineUninterceptedOrReturn(receiver: R, param: P, completion: Continuation<T>): Any? = this.asDynamic()(receiver, param, completion, false)\n /**\n Creates unintercepted coroutine without receiver and with result type [T].\n *\n This function creates a new, fresh instance of suspendable computation every time it is invoked.\n *\n To start executing the created coroutine, invoke `resume(Unit)` on the returned [Continuation] instance.\n *\n The [completion] continuation is invoked when coroutine completes with result or exception.\n *\n This function returns unintercepted continuation.\n *\n Invocation of `resume(Unit)` starts coroutine immediately in the invoker's call stack without going through the\n * [ContinuationInterceptor] that might be present in the completion's [CoroutineContext].\n *\n It is the invoker's responsibility to ensure that a proper invocation context is established.\n *\n Note that [completion] of this function may get invoked in an arbitrary context.\n *\n [Continuation.intercepted] can be used to acquire the intercepted continuation.\n *\n Invocation of `resume(Unit)` on intercepted continuation guarantees that execution of\n * both the coroutine and [completion] happens in the invocation context established by\n * [ContinuationInterceptor].\n *\n Repeated invocation of any resume function on the resulting continuation corrupts the\n * state machine of the coroutine and may result in arbitrary behaviour or

```

exception.\n *\n@SinceKotlin("1.3")\npublic actual fun <T> (suspend () -> T).createCoroutineUnintercepted(\n
completion: Continuation<T>)\n): Continuation<Unit> =\n // Kotlin/JS suspend lambdas have an extra parameter
`suspended`\n if (this.asDynamic().length == 2) {\n // When `suspended` is true the continuation is created,
but not executed\n this.asDynamic()(completion, true)\n } else {\n
createCoroutineFromSuspendFunction(completion) {\n this.asDynamic()(completion)\n }\n }\n\n/**\n
* Creates unintercepted coroutine with receiver type [R] and result type [T].\n * This function creates a new, fresh
instance of suspendable computation every time it is invoked.\n *\n * To start executing the created coroutine,
invoke `resume(Unit)` on the returned [Continuation] instance.\n *\n * The [completion] continuation is invoked when
coroutine completes with result or exception.\n *\n * This function returns unintercepted continuation.\n *
Invocation of `resume(Unit)` starts coroutine immediately in the invoker's call stack without going through the\n *
[ContinuationInterceptor] that might be present in the completion's [CoroutineContext].\n * It is the invoker's
responsibility to ensure that a proper invocation context is established.\n * Note that [completion] of this function
may get invoked in an arbitrary context.\n *\n * [Continuation.intercepted] can be used to acquire the intercepted
continuation.\n * Invocation of `resume(Unit)` on intercepted continuation guarantees that execution of\n * both the
coroutine and [completion] happens in the invocation context established by\n * [ContinuationInterceptor].\n *\n *
Repeated invocation of any resume function on the resulting continuation corrupts the\n * state machine of the
coroutine and may result in arbitrary behaviour or exception.\n *\n@SinceKotlin("1.3")\npublic actual fun <R, T>
(suspend R.() -> T).createCoroutineUnintercepted(\n receiver: R,\n completion: Continuation<T>)\n):
Continuation<Unit> =\n // Kotlin/JS suspend lambdas have an extra parameter `suspended`\n if
(this.asDynamic().length == 3) {\n // When `suspended` is true the continuation is created, but not executed\n
this.asDynamic()(receiver, completion, true)\n } else {\n createCoroutineFromSuspendFunction(completion)
{\n this.asDynamic()(receiver, completion)\n }\n }\n\n/**\n
* Intercepts this continuation with [ContinuationInterceptor].\n *\n * This function shall be used on the immediate result of
[createCoroutineUnintercepted] or [suspendCoroutineUninterceptedOrReturn],\n * in which case it checks for
[ContinuationInterceptor] in the continuation's [context][Continuation.context],\n * invokes
[ContinuationInterceptor.interceptContinuation], caches and returns the result.\n *\n * If this function is invoked on
other [Continuation] instances it returns `this` continuation unchanged.\n *\n@SinceKotlin("1.3")\npublic actual
fun <T> Continuation<T>.intercepted(): Continuation<T> =\n (this as? CoroutineImpl)?.intercepted() ?:
this\n\nprivate inline fun <T> createCoroutineFromSuspendFunction(\n completion: Continuation<T>,\n
crossinline block: () -> Any?)\n): Continuation<Unit> {\n @Suppress("UNCHECKED_CAST")\n return object
: CoroutineImpl(completion as Continuation<Any?>) {\n override fun doResume(): Any? {\n
exception?.let { throw it }\n return block()\n }\n }\n}\n\n"/*\n * Copyright 2010-2018 JetBrains s.r.o.
and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license
that can be found in the license/LICENSE.txt file.\n *\n@package kotlin.js\n// Mirrors signature from JS IR
BE\n// Used for
js.translator/testData/box/number/mulInt32.kt\n@library\n@JsName("imulEmulated")\n@Suppress("UNUSED_P
ARAMETER")\ninternal fun imul(x: Int, y: Int): Int =
definedExternally\n\n@Suppress("NOTHING_TO_INLINE")\ninternal inline fun isArrayish(o: dynamic) =
js("Kotlin").isArrayish(o)\n"/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n *\n@package kotlin\n// NOTE: Do not author your exceptions as they are written in
this file, instead use this template:\n\npublic open class MyException : Exception {\n constructor() : super()\n
constructor(message: String?) : super(message)\n constructor(message: String?, cause: Throwable?) :
super(message, cause)\n constructor(cause: Throwable?) : super(cause)\n}\n\n// TODO: remove primary
constructors, make all secondary KT-22055\n\n@Suppress("USELESS_ELVIS_RIGHT_IS_NULL")\npublic
actual open class Error actual constructor(message: String?, cause: Throwable?) : Throwable(message, cause ?: null)
{\n actual constructor() : this(null, null)\n actual constructor(message: String?) : this(message, null)\n
actual constructor(cause: Throwable?) : this(undefined,

```

```

cause)\n\n@Suppress("USELESS_ELVIS_RIGHT_IS_NULL")\npublic actual open class Exception actual
constructor(message: String?, cause: Throwable?): Throwable(message, cause ?: null) {\n  actual constructor() :
this(null, null)\n  actual constructor(message: String?): this(message, null)\n  actual constructor(cause:
Throwable?): this(undefiend, cause)\n}\n\npublic actual open class RuntimeException actual constructor(message:
String?, cause: Throwable?): Exception(message, cause) {\n  actual constructor() : this(null, null)\n  actual
constructor(message: String?): this(message, null)\n  actual constructor(cause: Throwable?): this(undefiend,
cause)\n}\n\npublic actual open class IllegalArgumentException actual constructor(message: String?, cause:
Throwable?): RuntimeException(message, cause) {\n  actual constructor() : this(null, null)\n  actual
constructor(message: String?): this(message, null)\n  actual constructor(cause: Throwable?): this(undefiend,
cause)\n}\n\npublic actual open class IllegalStateException actual constructor(message: String?, cause: Throwable?):
RuntimeException(message, cause) {\n  actual constructor() : this(null, null)\n  actual constructor(message:
String?): this(message, null)\n  actual constructor(cause: Throwable?): this(undefiend, cause)\n}\n\npublic actual
open class IndexOutOfBoundsException actual constructor(message: String?): RuntimeException(message) {\n
actual constructor() : this(null)\n}\n\npublic actual open class ConcurrentModificationException actual
constructor(message: String?, cause: Throwable?): RuntimeException(message, cause) {\n  actual constructor() :
this(null, null)\n  actual constructor(message: String?): this(message, null)\n  actual constructor(cause:
Throwable?): this(undefiend, cause)\n}\n\npublic actual open class UnsupportedOperationException actual
constructor(message: String?, cause: Throwable?): RuntimeException(message, cause) {\n  actual constructor() :
this(null, null)\n  actual constructor(message: String?): this(message, null)\n  actual constructor(cause:
Throwable?): this(undefiend, cause)\n}\n\npublic actual open class NumberFormatException actual
constructor(message: String?): IllegalArgumentException(message) {\n  actual constructor() :
this(null)\n}\n\npublic actual open class NullPointerException actual constructor(message: String?):
RuntimeException(message) {\n  actual constructor() : this(null)\n}\n\npublic actual open class
ClassCastException actual constructor(message: String?): RuntimeException(message) {\n  actual constructor() :
this(null)\n}\n\npublic actual open class AssertionError\n@SinceKotlin("1.4")\nconstructor(message: String?,
cause: Throwable?): Error(message, cause) {\n  actual constructor() : this(null)\n  constructor(message: String?):
this(message, null)\n  actual constructor(message: Any?): this(message.toString(), message as?
Throwable)\n}\n\npublic actual open class NoSuchElementException actual constructor(message: String?):
RuntimeException(message) {\n  actual constructor() : this(null)\n}\n\n@SinceKotlin("1.3")\npublic actual open
class ArithmeticException actual constructor(message: String?): RuntimeException(message) {\n  actual
constructor() : this(null)\n}\n\npublic actual open class NoWhenBranchMatchedException actual
constructor(message: String?, cause: Throwable?): RuntimeException(message, cause) {\n  actual constructor() :
this(null, null)\n  actual constructor(message: String?): this(message, null)\n  actual constructor(cause:
Throwable?): this(undefiend, cause)\n}\n\npublic actual open class UninitializedPropertyAccessException actual
constructor(message: String?, cause: Throwable?): RuntimeException(message, cause) {\n  actual constructor() :
this(null, null)\n  actual constructor(message: String?): this(message, null)\n  actual constructor(cause:
Throwable?): this(undefiend, cause)\n}\n"}\n\n/*\n * Copyright 2010-2019 JetBrains s.r.o. Use of this source code is
governed by the Apache 2.0 license\n * that can be found in the license/LICENSE.txt file.\n
*/\n\n@file:Suppress("UNUSED_PARAMETER")\n\npackage kotlin.js\n\n@kotlin.internal.InlineOnly\n\ninternal
inline fun jsDeleteProperty(obj: Any, property: Any) {\n  js("delete
obj[property]")\n}\n\n@kotlin.internal.InlineOnly\n\ninternal inline fun jsBitwiseOr(lhs: Any?, rhs: Any?): Int =\n
js("lhs | rhs").unsafeCast<Int>()\n"}\n\n/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\npackage kotlin.math\n\n/**\n * Returns this value with the sign bit same as of the
[sign] value.\n * If [sign] is `NaN` the sign of the result is undefined.\n */\n\n@SinceKotlin("1.2")\n\npublic actual
fun Double.withSign(sign: Double): Double {\n  val thisSignBit =
js("Kotlin").doubleSignBit(this).unsafeCast<Int>()\n  val newSignBit =
js("Kotlin").doubleSignBit(sign).unsafeCast<Int>()\n  return if (thisSignBit == newSignBit) this else -

```

```

this\n}"/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of
this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\npackage kotlin\n\n/**\n * Returns a bit representation of the specified floating-point value as [Long]\n * according to the IEEE 754 floating-point "double format" bit layout.\n
*/\n@SinceKotlin("1.2")\n@library("doubleToBits")\npublic actual fun Double.toBits(): Long =
definedExternally\n\n/**\n * Returns a bit representation of the specified floating-point value as [Long]\n * according to the IEEE 754 floating-point "double format" bit layout,\n * preserving `NaN` values exact layout.\n
*/\n@SinceKotlin("1.2")\n@library("doubleToRawBits")\npublic actual fun Double.toRawBits(): Long =
definedExternally\n\n/**\n * Returns the [Double] value corresponding to a given bit representation.\n
*/\n@SinceKotlin("1.2")\n@kotlin.internal.InlineOnly\npublic actual inline fun Double.Companion.fromBits(bits:
Long): Double = js("Kotlin").doubleFromBits(bits).unsafeCast<Double>()\n\n/**\n * Returns a bit representation
of the specified floating-point value as [Int]\n * according to the IEEE 754 floating-point "single format" bit
layout.\n * Note that in Kotlin/JS [Float] range is wider than "single format" bit layout can represent,\n * so
some [Float] values may overflow, underflow or lose their accuracy after conversion to bits and back.\n
*/\n@SinceKotlin("1.2")\n@library("floatToBits")\npublic actual fun Float.toBits(): Int =
definedExternally\n\n/**\n * Returns a bit representation of the specified floating-point value as [Int]\n * according
to the IEEE 754 floating-point "single format" bit layout,\n * preserving `NaN` values exact layout.\n * Note
that in Kotlin/JS [Float] range is wider than "single format" bit layout can represent,\n * so some [Float] values
may overflow, underflow or lose their accuracy after conversion to bits and back.\n
*/\n@SinceKotlin("1.2")\n@library("floatToRawBits")\npublic actual fun Float.toRawBits(): Int =
definedExternally\n\n/**\n * Returns the [Float] value corresponding to a given bit representation.\n
*/\n@SinceKotlin("1.2")\n@kotlin.internal.InlineOnly\npublic actual inline fun Float.Companion.fromBits(bits:
Int): Float =
js("Kotlin").floatFromBits(bits).unsafeCast<Float>()\n\n@Suppress("NOTHING_TO_INLINE")\ninternal
inline fun Long(low: Int, high: Int) = js("Kotlin").Long.fromBits(low, high).unsafeCast<Long>()\ninternal inline
val Long.low: Int get() = this.asDynamic().getLowBits().unsafeCast<Int>()\ninternal inline val Long.high: Int get()
= this.asDynamic().getHighBits().unsafeCast<Int>()\n"/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n */\n\nimport kotlin.reflect.KClass\n\n@PublishedApi\ninternal fun <T :
Annotation> KClass<*>.findAssociatedObject(@Suppress("UNUSED_PARAMETER") annotationClass:
KClass<T>): Any? {\n // This API is not supported in js-v1. Return `null` to be source-compatible with js-ir.\n
return null\n}\n"/*\n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.\n *
Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\npackage kotlin.text\n\n/**\n * Returns a string representation of this [Long] value in the specified [radix].\n
*/\n * @throws IllegalArgumentException when [radix] is not a valid radix for number to string conversion.\n
*/\n@SinceKotlin("1.2")\npublic actual fun Long.toString(radix: Int): String =
asDynamic().toString(checkRadix(radix))"/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\npackage kotlin.text\n\n/\n// NOTE: THIS FILE IS AUTO-GENERATED by the
GenerateUnicodeData.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n/\n/\n// 1343 ranges
totally\nprivate object Category {\n val decodedRangeStart: IntArray\n val decodedRangeCategory: IntArray\n
\n init {\n val toBase64 =
"\n val fromBase64 =
IntArray(128)\n for (i in toBase64.indices) {\n fromBase64[toBase64[i].code] = i\n }\n \n //
rangeStartDiff.length = 1482\n val rangeStartDiff =
"\ngBCFEDCKCDCaDDaDBhBCEEDDDDDDEDXBHYBH5BRwBGDCHDCIDFHDCHFDCDEIRTEE7BGHDDJI
CBbSEMOFGERwDEDDDDDECEFCRBjHbFDCYFFCCzBvBjBBFC3BohDBmBDGpBDDCtBBJlBEELGDFC
LDCgBBKVKEDiDDHCFECECKCEODBebC5CLBOKhBJDDDDWEBHFCFCPBZDEL1BVBSLPBgBB2BDB

```



```

2 -> code shr 10\n          1 -> (code shr 5) and 0x1f\n          else -> code and 0x1f\n          }\n }\n}\n\n**\n
* Returns the Unicode general category of this character as an Int.\n *^\ninternal fun Char.getCategoryValue(): Int
{\n  val ch = this.code\n  val index = binarySearchRange(Category.decodedRangeStart, ch)\n  val start =
Category.decodedRangeStart[index]\n  val code = Category.decodedRangeCategory[index]\n  val value =
categoryValueFrom(code, ch - start)\n  return if (value == 17) CharCategory.UNASSIGNED.value else
value\n}\n\ninternal fun decodeVarLenBase64(base64: String, fromBase64: IntArray, resultLength: Int): IntArray
{\n  val result = IntArray(resultLength)\n  var index = 0\n  var int = 0\n  var shift = 0\n  for (char in base64)
{\n    val sixBit = fromBase64[char.code]\n    int = int or ((sixBit and 0x1f) shl shift)\n    if (sixBit < 0x20)
{\n      result[index++] = int\n      int = 0\n      shift = 0\n    } else {\n      shift += 5\n    }\n  }\n}\n
return result\n}\n", "/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n *
Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*^\n\npackage kotlin.collections\n\n/\n// NOTE: THIS FILE IS AUTO-GENERATED by the
GenerateStandardLib.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\nimport
kotlin.js.*\nimport kotlin.ranges.contains\nimport kotlin.ranges.reversed\n\n**\n * Reverses elements in the list in-
place.\n *^\npublic actual fun <T> MutableList<T>.reverse(): Unit {\n  val midPoint = (size / 2) - 1\n  if
(midPoint < 0) return\n  var reverseIndex = lastIndex\n  for (index in 0..midPoint) {\n    val tmp = this[index]\n
this[index] = this[reverseIndex]\n    this[reverseIndex] = tmp\n    reverseIndex--\n  }\n}\n\n", "/*\n *
Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is
governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n *^\n\npackage
kotlin.text\n\n/\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateUnicodeData.kt\n// See:
https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\n/\n// 37 ranges totally\nprivate object Digit {\n
internal val rangeStart = intArrayOf(\n  0x0030, 0x0660, 0x06f0, 0x07c0, 0x0966, 0x09e6, 0x0a66, 0x0ae6,
0x0b66, 0x0be6, 0x0c66, 0x0ce6, 0x0d66, 0x0de6, 0x0e50, 0x0ed0, 0x0f20, 0x1040, 0x1090, 0x17e0, \n
0x1810, 0x1946, 0x19d0, 0x1a80, 0x1a90, 0x1b50, 0x1bb0, 0x1c40, 0x1c50, 0xa620, 0xa8d0, 0xa900, 0xa9d0,
0xa9f0, 0xaa50, 0xabf0, 0xff10, \n  )\n}\n\n**\n * Returns the index of the largest element in [array] smaller or
equal to the specified [needle],\n * or -1 if [needle] is smaller than the smallest element in [array].\n *^\ninternal fun
binarySearchRange(array: IntArray, needle: Int): Int {\n  var bottom = 0\n  var top = array.size - 1\n  var middle
= -1\n  var value = 0\n  while (bottom <= top) {\n    middle = (bottom + top) / 2\n    value = array[middle]\n
if (needle > value)\n      bottom = middle + 1\n    else if (needle == value)\n      return middle\n
else\n      top = middle - 1\n  }\n  return middle - (if (needle < value) 1 else 0)\n}\n\n**\n * Returns an integer
from 0..9 indicating the digit this character represents,\n * or -1 if this character is not a digit.\n *^\ninternal fun
Char.digitToIntImpl(): Int {\n  val ch = this.code\n  val index = binarySearchRange(Digit.rangeStart, ch)\n  val
diff = ch - Digit.rangeStart[index]\n  return if (diff < 10) diff else -1\n}\n\n**\n * Returns `true` if this character is
a digit.\n *^\ninternal fun Char.isDigitImpl(): Boolean {\n  return digitToIntImpl() >= 0\n}\n\n", "/*\n * Copyright
2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed
by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n *^\n\npackage kotlin.text\n\n/\n//
NOTE: THIS FILE IS AUTO-GENERATED by the GenerateUnicodeData.kt\n// See:
https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\n/\n// 222 ranges totally\nprivate object Letter {\n
val decodedRangeStart: IntArray\n  val decodedRangeLength: IntArray\n  val decodedRangeCategory: IntArray\n
\n  init {\n    val toBase64 =
\"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/\n    val fromBase64 =
IntArray(128)\n    for (i in toBase64.indices) {\n      fromBase64[toBase64[i].code] = i\n    }\n    \n    //
rangeStartDiff.length = 356\n    val rangeStartDiff =
\"hCgBpCQGYHZH5BRpBPPPPPRMP5BPPICPP6BkEPPPPcXPzBvBrB3BOiDoBHwD+E3DauCnFmBmB2D
6E1BIBTiBmBIBP5BhBiBrBvBjBqBnBPRtBiCmCtBIB0BmB5BiB7BmBgEmChBZgCoEoGVpBsFrhBPqKQ2B
wBYoFgB4CJuTiEvBuCuDrF5DgEgFIJ1DgFmBQtBsBRGsB+BPiBID1EiJDPRPPPPPPPPPGQSQS/DxENVNU+
B9zCwBwBPPcKDPNnBPqDYI1R8B7FkFgTgwGgwUwmBgKwBuBScmEP/BPPPPPrBP8B7F1B/ErBqC6B7B
iBmBfQsBUwCw/KwqIwLwETPcPjQgJxFgBIBsD\n    val diff = decodeVarLenBase64(rangeStartDiff,

```

```

fromBase64, 222)\n    val start = IntArray(diff.size)\n    for (i in diff.indices) {\n        if (i == 0) start[i] =
diff[i]\n        else start[i] = start[i - 1] + diff[i]\n    }\n    decodedRangeStart = start\n    \n    //
rangeLength.length = 328\n    val rangeLength =
\"aaMBXHYH5BRpBPPPPPRMP5BPPICPPzBDOOPPcXPzBvBjB3BOhDmBBpB7DoDYxB+EiBP1DoExBkB
QhBekBPmBgBhBctBiBMWOOXhCsBpBkBUV3Ba4BkB0DiCgBXgBtD4FSdBfPhBPpKP0BvBXjEQ2CGsT8Dh
BtCqDpFvD1D3E0lrD2EkBJrBDOBsB+BPiBIB1EiJDPPPPPPPPPPGPPMNLsBNPNPKCvBvBPPcKDPBmBPh
DXXgD4B6FzEgDguG9vUtkB9JcuBSckEP/BPPPPPPBPf4FrBjEhBpC3B5BKaWPrBOWck/KsCuLqDHPbPxPsFt
EaaqDL\""\n    decodedRangeLength = decodeVarLenBase64(rangeLength, fromBase64, 222)\n    \n    //
rangeCategory.length = 959\n    val rangeCategory =
\"GFjgggUHGFFZZZmzpz5qB6s6020B60ptltB6smt2sB60mz22B1+vv+8BZZ5s2850BW5q1ymtB506smzBF3q1
q1qB1q1q1+Bgii4wDTm74g3KigxqM60q1q1Bq1o1q1BF1qlqrBZ2q5wprBGFZWWZGHFsjiioLowgmOowjkw
CkgoiIk7ligGogiioBkwkiYkzj2oNoi+sbkwj04DghhKQ8wgiYkgoioDsgnkwC4gikQ//v+85BkwvoIsgoyI4ygu0whiw
Eowri4CoghsJowgqYowgm4DkwgsY/nwnzPowhmYkg6wI8yggZswikwHgxgmIoxgqYkwwg4DkxgmIkgoioBsgsso
BgzyI8g9gL8g9kI0wgwJoxgkoC0wgioFkw/wI0w53iF4gioYowjmgBHGq1qkgwBF1q1q8qBHwghuIwghyKk0go
QkwgoQk3goQHGFHkyg0pBgxj6IoinkxDswno7Ikwhz9Bo0gioB8z48Rwli0xN0mpjoX8w78pDwltoqKHFGGwwg
sIHFH3q1q16BFHWFZ1q10q1B2qlwq1B1q10q1B2q1yq1B6q1gq1Biq1qhxBir1qp1Bqt1q1qB1g1q1+B//3q16B//q
1qBH/qlq9Bholqq9B1i00a1q10qD1op1HkwmigEigy6Cptogq1Bixo1kDq7/j00B2qgoBWGFm1lz50B6s5q1+BG
WhggzhwBFFhgk4//Bo2jigE8wguI8wguI8wgugUog1qoB4qjmIwwi2KgkYHHH4lBgiFWkgIWoghssMmz5smrBZ
3q1y50B5sm7gzBtz1smzB5smz50BqzqtzmzB5sgzqzBF2/9//5BowgoIwmnkzPkwgk4C8ys65BkgoqI0wgy6FghquZ0
2giY0ghiIsgH24B4ghsQ8QF/v1q1OFs0O8iCHHF1qggz/B8wg6Izvn+//B08QgohsjK0QGFk7hsQ4gB\""\n
decodedRangeCategory = decodeVarLenBase64(rangeCategory, fromBase64, 222)\n    }\n}\n\n/**\n * Returns
`true` if this character is a letter.\n */\ninternal fun Char.isLetterImpl(): Boolean {\n    return getLetterType() !=
0\n}\n\n/**\n * Returns `true` if this character is a lower case letter, or it has contributory property
`Other_Lowercase`.\n */\ninternal fun Char.isLowercaseImpl(): Boolean {\n    return getLetterType() == 1 ||
code.isOtherLowercase()\n}\n\n/**\n * Returns `true` if this character is an upper case letter, or it has contributory
property `Other_Uppercase`.\n */\ninternal fun Char.isUppercaseImpl(): Boolean {\n    return getLetterType() == 2
|| code.isOtherUppercase()\n}\n\n/**\n * Returns\n * - `1` if the character is a lower case letter,\n * - `2` if the
character is an upper case letter,\n * - `3` if the character is a letter but not a lower or upper case letter,\n * - `0`
otherwise.\n */\nprivate fun Char.getLetterType(): Int {\n    val ch = this.code\n    val index =
binarySearchRange(Letter.decodedRangeStart, ch)\n    val rangeStart = Letter.decodedRangeStart[index]\n    val
rangeEnd = rangeStart + Letter.decodedRangeLength[index] - 1\n    val code =
Letter.decodedRangeCategory[index]\n    if (ch > rangeEnd) {\n        return 0\n    }\n    val lastTwoBits = code
and 0x3\n    if (lastTwoBits == 0) { // gap pattern\n        val shift = 2\n        val threshold = rangeStart\n        for (i
in 0..1) {\n            threshold += (code shr shift) and 0x7f\n            if (threshold > ch) {\n                return 3\n
            }\n            shift += 7\n            threshold += (code shr shift) and 0x7f\n            if (threshold > ch) {\n
                return 0\n            }\n            shift += 7\n        }\n        return 3\n    }\n    if (code <= 0x7) {\n
        return lastTwoBits\n    }\n    val distance = (ch - rangeStart)\n    val shift = if (code <= 0x1F) distance % 2
else distance\n    return (code shr (2 * shift)) and 0x3\n}\n\n", "/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\npackage kotlin.text\n\n/\n// NOTE: THIS FILE IS AUTO-GENERATED by the
GenerateUnicodeData.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n\n/\n\nprivate object
OtherLowercase {\n    internal val otherLowerStart = intArrayOf(\n        0x00aa, 0x00ba, 0x02b0, 0x02c0, 0x02e0,
0x0345, 0x037a, 0x1d2c, 0x1d78, 0x1d9b, 0x2071, 0x207f, 0x2090, 0x2170, 0x24d0, 0x2c7c, 0xa69c, 0xa770,
0xa7f8, 0xab5c, \n    )\n    internal val otherLowerLength = intArrayOf(\n        1, 1, 9, 2, 5, 1, 1, 63, 1, 37, 1, 1, 13,
16, 26, 2, 2, 1, 2, 4, \n    )\n}\n\ninternal fun Int.isOtherLowercase(): Boolean {\n    val index =
binarySearchRange(OtherLowercase.otherLowerStart, this)\n    return index >= 0 && this <
OtherLowercase.otherLowerStart[index] + OtherLowercase.otherLowerLength[index]\n}\n\n", "/*\n * Copyright
2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed

```

by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.text\n\n/\n//
NOTE: THIS FILE IS AUTO-GENERATED by the GenerateUnicodeData.kt\n// See:
[\n\ninternal fun Int.isOtherUppercase\(\): Boolean](https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib)
{\n return this in 0x2160..0x216f\n || this in 0x24b6..0x24cf\n}\n","/*\n * Copyright 2010-2021 JetBrains
s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0
license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.text\n\n/\n// NOTE: THIS FILE IS
AUTO-GENERATED by the GenerateStandardLib.kt\n// See:
[\n\nimport kotlin.js.*\n\n/**\n * Returns a
character at the given \[index\] or throws an \[IndexOutOfBoundsException\] if the \[index\] is out of bounds of this char
sequence.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n */\n\npublic actual fun
CharSequence.elementAt\(index: Int\): Char {\n return elementAtOrElse\(index\) { throw
IndexOutOfBoundsException\("index: \\$index, length: \\$length}"\) }\n}\n\n","/*\n * Copyright 2010-2021 JetBrains
s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0
license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.text\n\n/\n// NOTE: THIS FILE IS
AUTO-GENERATED by the GenerateUnicodeData.kt\n// See:
\[\n\n// 4 ranges totally\n\ninternal fun\]\(https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\)
Char.titlecaseCharImpl\(\): Char {\n val code = this.code\n // Letters repeating <Lu, Lt, Ll> sequence and code of
the Lt is a multiple of 3, e.g. <\u01c4, \u01c5, \u01c6>\n if \(code in 0x01c4..0x01cc || code in 0x01f1..0x01f3\) {\n
return \(3 * \(\(code + 1\) / 3\)\).toChar\(\)\n }\n // Lower case letters whose title case mapping equivalent is equal
to the original letter\n if \(code in 0x10d0..0x10fa || code in 0x10fd..0x10ff\) {\n return this\n }\n return
uppercaseChar\(\)\n}","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\npackage kotlin.collections\n\n/\n// NOTE: THIS FILE IS AUTO-GENERATED
by the GenerateStandardLib.kt\n// See: \[\n\nimport\]\(https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\)
kotlin.js.*\nimport kotlin.ranges.contains\nimport kotlin.ranges.reversed\n\n/**\n * Returns an element at the given
\[index\] or throws an \[IndexOutOfBoundsException\] if the \[index\] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAt\n */\n\n@SinceKotlin\("1.3"\)\n@ExperimentalUnsignedTypes\n\npublic actual fun UIntArray.elementAt\(index: Int\):
UInt {\n return elementAtOrElse\(index\) { throw IndexOutOfBoundsException\("index: \\$index, size: \\$size"\) }\n
}\n}\n\n/**\n * Returns an element at the given \[index\] or throws an \[IndexOutOfBoundsException\] if the \[index\] is
out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n */\n\n@SinceKotlin\("1.3"\)\n@ExperimentalUnsignedTypes\n\npublic actual fun ULongArray.elementAt\(index: Int\):
ULong {\n return elementAtOrElse\(index\) { throw IndexOutOfBoundsException\("index: \\$index, size: \\$size"\) }\n
}\n}\n\n/**\n * Returns an element at the given \[index\] or throws an \[IndexOutOfBoundsException\] if the \[index\] is
out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n */\n\n@SinceKotlin\("1.3"\)\n@ExperimentalUnsignedTypes\n\npublic actual fun UByteArray.elementAt\(index: Int\):
UByte {\n return elementAtOrElse\(index\) { throw IndexOutOfBoundsException\("index: \\$index, size: \\$size"\) }\n
}\n}\n\n/**\n * Returns an element at the given \[index\] or throws an \[IndexOutOfBoundsException\] if the \[index\] is
out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n */\n\n@SinceKotlin\("1.3"\)\n@ExperimentalUnsignedTypes\n\npublic actual fun UShortArray.elementAt\(index: Int\):
UShort {\n return elementAtOrElse\(index\) { throw IndexOutOfBoundsException\("index: \\$index, size: \\$size"\) }\n
}\n}\n\n/**\n * Returns a \[List\] that wraps the original array.\n */\n\n@SinceKotlin\("1.3"\)\n@ExperimentalUnsignedTypes\n\npublic actual fun UIntArray.asList\(\): List<UInt> {\n
return object : AbstractList<UInt>\(\), RandomAccess {\n override val size: Int get\(\) = this@asList.size\n
override fun isEmpty\(\): Boolean = this@asList.isEmpty\(\)\n override fun contains\(element: UInt\): Boolean =
this@asList.contains\(element\)\n override fun get\(index: Int\): UInt {\n
AbstractList.checkElementIndex\(index, size\)\n return this@asList\[index\]\n }\n override fun
indexOf\(element: UInt\): Int {\n @Suppress\("USELESS_CAST"\)\n if \(\(element as Any?\) !is UInt\)](https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib)

applied to either files or top-level declarations.\n * It is currently prohibited to export the following kinds of declarations:\n * `expect` declarations\n * inline functions with reified type parameters\n * suspend functions\n * secondary constructors without `@JsName`\n * extension properties\n * enum classes\n * annotation classes\n * Signatures of exported declarations must only contain \"exportable\" types:\n * `dynamic`, `Any`, `String`, `Boolean`, `Byte`, `Short`, `Int`, `Float`, `Double`\n * `BooleanArray`, `ByteArray`, `ShortArray`, `IntArray`, `FloatArray`, `DoubleArray`\n * `Array<exportable-type>`\n * Function types with exportable parameters and return types\n * `external` or `@JsExport` classes and interfaces\n * Nullable counterparts of types above\n * Unit return type. Must not be nullable\n * This annotation is experimental, meaning that restrictions mentioned above are subject to change.\n

```
*\n@ExperimentalJsExport\n@Retention(AnnotationRetention.BINARY)\n@Target(CLASS, PROPERTY, FUNCTION, FILE)\n@SinceKotlin(\"1.3\")\npublic actual annotation class JsExport\n/**\n * Forces a top-level property to be initialized eagerly, opposed to lazily on the first access to file and/or property.\n
```

```
*\n@ExperimentalStdlibApi\n@Retention(AnnotationRetention.BINARY)\n@Target(AnnotationTarget.PROPERTY)\n@SinceKotlin(\"1.6\")\n@Deprecated(\"This annotation is a temporal migration assistance and may be removed in the future releases, please consider filing an issue about the case where it is needed\")\npublic annotation class EagerInitialization\n", /*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\npackage kotlin.jvm\n\n// these are used in common generated code in stdlib\n\n// TODO: find how to deprecate these
```

```
ones\n\n@Target(AnnotationTarget.FIELD)\n@Retention(AnnotationRetention.SOURCE)\npublic actual annotation class Volatile\n\n@Target(AnnotationTarget.FUNCTION, AnnotationTarget.PROPERTY_GETTER, AnnotationTarget.PROPERTY_SETTER)\n@Retention(AnnotationRetention.SOURCE)\npublic actual annotation class Synchronized\n", /*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\npackage kotlin.collections\n\n/**\n * Provides a skeletal implementation of the [MutableCollection] interface.\n * @param E the type of elements contained in the collection. The collection is invariant in its element type.\n */\npublic actual abstract class AbstractMutableCollection<E> protected actual constructor() : AbstractCollection<E>(), MutableCollection<E> {\n\n    actual abstract override fun add(element: E): Boolean\n\n    actual override fun remove(element: E): Boolean {\n        checkIsMutable()\n        val iterator = iterator()\n        while (iterator.hasNext()) {\n            if (iterator.next() == element) {\n                iterator.remove()\n                return true\n            }\n        }\n        return false\n    }\n\n    actual override fun addAll(elements: Collection<E>): Boolean {\n        checkIsMutable()\n        var modified = false\n        for (element in elements) {\n            if (add(element)) modified = true\n        }\n        return modified\n    }\n\n    actual override fun removeAll(elements: Collection<E>): Boolean {\n        checkIsMutable()\n        return (this as MutableIterable<E>).removeAll { it in elements }\n    }\n\n    actual override fun retainAll(elements: Collection<E>): Boolean {\n        checkIsMutable()\n        return (this as MutableIterable<E>).removeAll { it !in elements }\n    }\n\n    actual override fun clear(): Unit {\n        checkIsMutable()\n        val iterator = this.iterator()\n        while (iterator.hasNext()) {\n            iterator.next()\n            iterator.remove()\n        }\n    }\n}
```

```
@Deprecated(\"Provided so that subclasses inherit this function\", level = DeprecationLevel.HIDDEN)\n\n@JsName(\"toJSON\")\nprotected fun toJSON(): Any = this.toArray()\n\n/**\n * This method is called every time when a mutating method is called on this mutable collection.\n * Mutable collections that are built (frozen) must throw `UnsupportedOperationException`.\n */\ninternal open fun checkIsMutable(): Unit {\n}\n\n", /*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
```

```
*\n\n/**\n * Based on GWT AbstractList\n * Copyright 2007 Google Inc.\n */\n\npackage kotlin.collections\n\n/**\n * Provides a skeletal implementation of the [MutableList] interface.\n * @param E the type of elements contained in the list. The list is invariant in its element type.\n */\npublic actual abstract class AbstractMutableList<E> protected actual constructor() : AbstractMutableCollection<E>(), MutableList<E> {\n
```



```

modCount++\n    return true\n    }\n\n    actual override fun removeAt(index: Int): E {\n        checkIsMutable()\n        rangeCheck(index)\n        modCount++\n        return if (index == lastIndex)\n            array.asDynamic().pop()\n        else\n            array.asDynamic().splice(index, 1)[0]\n    }\n\n    actual override fun remove(element: E): Boolean {\n        checkIsMutable()\n        for (index in array.indices) {\n            if (array[index] == element) {\n                array.asDynamic().splice(index, 1)\n                modCount++\n                return true\n            }\n        }\n        return false\n    }\n\n    override fun removeRange(fromIndex: Int, toIndex: Int) {\n        checkIsMutable()\n        modCount++\n        array.asDynamic().splice(fromIndex, toIndex - fromIndex)\n    }\n\n    actual override fun clear() {\n        checkIsMutable()\n        array = emptyArray()\n        modCount++\n    }\n\n\n    actual override fun indexOf(element: E): Int = array.indexOf(element)\n\n    actual override fun lastIndexOf(element: E): Int = array.lastIndexOf(element)\n\n    override fun toString() = arrayToString(array)\n\n    @Suppress("UNCHECKED_CAST")\n    override fun <T> toArray(array: Array<T>): Array<T> {\n        if (array.size < size) {\n            return toArray() as Array<T>\n        }\n        (this.array as Array<T>).copyInto(array)\n        if (array.size > size) {\n            array[size] = null as T // null-terminate\n        }\n        return array\n    }\n\n    override fun toArray(): Array<Any?> {\n        return js("[]").slice.call(array)\n    }\n\n\n    internal override fun checkIsMutable() {\n        if (isReadOnly) throw UnsupportedOperationException()\n    }\n\n    private fun rangeCheck(index: Int) = index.apply {\n        AbstractList.checkElementIndex(index, size)\n    }\n\n    private fun insertionRangeCheck(index: Int) = index.apply {\n        AbstractList.checkPositionIndex(index, size)\n    }\n}"/**\n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.collections\n\n\ninternal fun <T> sortArrayWith(array: Array<out T>, comparison: (T, T) -> Int) {\n    if (getStableSortingIsSupported()) {\n        array.asDynamic().sort(comparison)\n    } else {\n        mergeSort(array.unsafeCast<Array<T>>(), 0, array.lastIndex, Comparator(comparison))\n    }\n}\n\n\ninternal fun <T> sortArrayWith(array: Array<out T>, comparator: Comparator<in T>) {\n    if (getStableSortingIsSupported()) {\n        val comparison = { a: T, b: T -> comparator.compare(a, b) }\n        array.asDynamic().sort(comparison)\n    } else {\n        mergeSort(array.unsafeCast<Array<T>>(), 0, array.lastIndex, comparator)\n    }\n}\n\n\ninternal fun <T> sortArrayWith(array: Array<out T>, fromIndex: Int, toIndex: Int, comparator: Comparator<in T>) {\n    if (fromIndex < toIndex - 1) {\n        mergeSort(array.unsafeCast<Array<T>>(), fromIndex, toIndex - 1, comparator)\n    }\n}\n\n\ninternal fun <T : Comparable<T>> sortArray(array: Array<out T>) {\n    if (getStableSortingIsSupported()) {\n        val comparison = { a: T, b: T -> a.compareTo(b) }\n        array.asDynamic().sort(comparison)\n    } else {\n        mergeSort(array.unsafeCast<Array<T>>(), 0, array.lastIndex, naturalOrder())\n    }\n}\n\n\nprivate var _stableSortingIsSupported: Boolean? = null\n\nprivate fun getStableSortingIsSupported(): Boolean {\n    _stableSortingIsSupported?.let { return it }\n    _stableSortingIsSupported = false\n\n    val array = js("[]").unsafeCast<Array<Int>>()\n    // known implementations may use stable sort for arrays of up to 512 elements\n    // so we create slightly more elements to test stability\n    for (index in 0 until 600) array.asDynamic().push(index)\n    val comparison = { a: Int, b: Int -> (a and 3) - (b and 3) }\n    array.asDynamic().sort(comparison)\n    for (index in 1 until array.size) {\n        val a = array[index - 1]\n        val b = array[index]\n        if ((a and 3) == (b and 3) && a >= b) return false\n    }\n    _stableSortingIsSupported = true\n    return true\n}\n\n\nprivate fun <T> mergeSort(array: Array<T>, start: Int, endInclusive: Int, comparator: Comparator<in T>) {\n    val buffer = arrayOfNulls<Any?>(array.size).unsafeCast<Array<T>>()\n    val result = mergeSort(array, buffer, start, endInclusive, comparator)\n    if (result !== array) {\n        for (i in start..endInclusive) array[i] = result[i]\n    }\n}\n\n\n// Both start and end are inclusive indices.\nprivate fun <T> mergeSort(array: Array<T>, buffer: Array<T>, start: Int, end: Int, comparator: Comparator<in T>): Array<T> {\n    if (start == end) {\n        return array\n    }\n\n    val median = (start + end) / 2\n    val left = mergeSort(array, buffer, start, median, comparator)\n    val right = mergeSort(array, buffer, median + 1, end, comparator)\n    val target = if (left === buffer) array else buffer\n\n    // Merge.\n    var leftIndex = start\n    var rightIndex = median + 1\n    for (i in start..end) {\n        when {\n            leftIndex <= median && rightIndex <= end -> {\n                val leftValue = left[leftIndex]\n                val rightValue = right[rightIndex]\n                if (comparator.compare(leftValue, rightValue) <= 0) {\n                    target[i] =

```

```

leftValue\n          leftIndex++\n          } else {\n          target[i] = rightValue\n
rightIndex++\n          }\n          }\n          leftIndex <= median -> {\n          target[i] = left[leftIndex]\n
leftIndex++\n          }\n          else /* rightIndex <= end */ -> {\n          target[i] = right[rightIndex]\n
rightIndex++\n          Unit // TODO: Fix KT-31506\n          }\n          }\n          }\n          return target\n}", "/*\n *
Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is
governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage
kotlin.collections\n\n@OptIn(ExperimentalUnsignedTypes::class)\n@SinceKotlin("1.3")\n@kotlin.js.JsName("\n
contentDeepHashCodeImpl")\n\ninternal fun <T> Array<out T>?.contentDeepHashCodeImpl(): Int {\n if (this ==
null) return 0\n var result = 1\n for (element in this) {\n val elementHash = when {\n element == null
-> 0\n isArrayish(element) -> (element.unsafeCast<Array<*>>()).contentDeepHashCodeImpl()\n
element is UByteArray -> element.contentHashCode()\n element is UShortArray ->
element.contentHashCode()\n element is UIntArray -> element.contentHashCode()\n element is
ULongArray -> element.contentHashCode()\n else -> element.hashCode()\n }\n\n
result = 31 * result + elementHash\n }\n return result\n}", "/*\n * Copyright 2010-2018 JetBrains s.r.o. and
Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.collections\n\ninternal interface
EqualityComparator {\n /*\n * Subclasses must override to return a value indicating\n * whether or not two
keys or values are equal.\n */\n abstract fun equals(value1: Any?, value2: Any?): Boolean\n\n /*\n *
Subclasses must override to return the hash code of a given key.\n */\n abstract fun getHashCode(value: Any?):
Int\n\n\n object GetHashCode : EqualityComparator {\n override fun equals(value1: Any?, value2: Any?):
Boolean = value1 == value2\n\n override fun getHashCode(value: Any?): Int = value?.hashCode() ?: 0\n
}\n}", "/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this
source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n/*\n *
Based on GWT AbstractHashMap\n * Copyright 2008 Google Inc.\n */\n\npackage kotlin.collections\n\nimport
kotlin.collections.MutableMap.MutableEntry\n\n/*\n * Hash table based implementation of the [MutableMap]
interface.\n *\n * This implementation makes no guarantees regarding the order of enumeration of [keys], [values]
and [entries] collections.\n */\n\n// Classes that extend HashMap and implement `build()` (freezing) operation\n// have
to make sure mutating methods check `checkIsMutable`.\n\npublic actual open class HashMap<K, V> :
AbstractMutableMap<K, V>, MutableMap<K, V> {\n\n private inner class EntrySet :
AbstractEntrySet<MutableEntry<K, V>, K, V>() {\n\n override fun add(element: MutableEntry<K, V>):
Boolean = throw UnsupportedOperationException("Add is not supported on entries")\n\n override fun clear()
{\n this@HashMap.clear()\n }\n\n override fun containsEntry(element: Map.Entry<K, V>): Boolean
= this@HashMap.containsEntry(element)\n\n override operator fun iterator():
MutableIterator<MutableEntry<K, V>> = internalMap.iterator()\n\n override fun removeEntry(element:
Map.Entry<K, V>): Boolean {\n if (contains(element)) {\n this@HashMap.remove(element.key)\n
return true\n }\n return false\n }\n\n override val size: Int get() =
this@HashMap.size\n }\n\n\n /*\n * Internal implementation of the map: either string-based or hashcode-
based.\n */\n private val internalMap: InternalMap<K, V>\n private val equality: EqualityComparator\n\n
internal constructor(internalMap: InternalMap<K, V>) : super() {\n this.internalMap = internalMap\n
this.equality = internalMap.equality\n }\n\n\n /*\n * Constructs an empty [HashMap] instance.\n */\n\n
actual constructor() : this(InternalHashCodeMap(EqualityComparator.HashCode))\n\n /*\n * Constructs an
empty [HashMap] instance.\n *\n * @param initialCapacity the initial capacity (ignored)\n * @param
loadFactor the load factor (ignored)\n *\n * @throws IllegalArgumentException if the initial capacity or
load factor are negative\n */\n\n actual constructor(initialCapacity: Int, loadFactor: Float) : this() {\n // This
implementation of HashMap has no need of load factors or capacities.\n require(initialCapacity >= 0) {\n
\n "Negative initial capacity: $initialCapacity"\n }\n require(loadFactor >= 0) {\n\n "Non-positive load factor:
$loadFactor"\n }\n }\n\n\n actual constructor(initialCapacity: Int) : this(initialCapacity, 0.0f)\n\n\n /*\n *
Constructs an instance of [HashMap] filled with the contents of the specified [original] map.\n */\n\n actual

```

```

constructor(original: Map<out K, V>) : this() {\n    this.putAll(original)\n    }\n\n    actual override fun clear() {\n        internalMap.clear()\n    }\n\n    structureChanged(this)\n    }\n\n    actual override fun containsKey(key: K): Boolean = internalMap.containsKey()\n\n    actual override fun containsValue(value: V): Boolean = internalMap.any { equality.equals(it.value, value) }\n\n    private var _entries: MutableSet<MutableMap.MutableEntry<K, V>>? = null\n\n    actual override val entries: MutableSet<MutableMap.MutableEntry<K, V>>\n        get() {\n            if (_entries == null) {\n                _entries = createEntrySet()\n            }\n            return _entries!!\n        }\n\n    internal open fun createEntrySet(): MutableSet<MutableMap.MutableEntry<K, V>> = EntrySet()\n\n    actual override operator fun get(key: K): V? = internalMap.get(key)\n\n    actual override fun put(key: K, value: V): V? = internalMap.put(key, value)\n\n    actual override fun remove(key: K): V? = internalMap.remove(key)\n\n    actual override val size: Int get() = internalMap.size\n    }\n\n    /**\n     * Constructs the specialized implementation of [HashMap] with [String] keys, which stores the keys as properties of\n     * JS object without hashing them.\n     */\n\n    public fun <V> stringMapOf(vararg pairs: Pair<String, V>): HashMap<String, V> {\n        return HashMap<String, V>(InternalStringMap(EqualityComparator.HashCode)).apply { putAll(pairs) }\n    }\n\n    "\n    /*\n     * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n     * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n     */\n\n    /*\n     * Based on GWT HashSet\n     * Copyright 2008 Google Inc.\n     */\n\n    package kotlin.collections\n\n    /**\n     * The implementation of the [MutableSet] interface, backed by a [HashMap] instance.\n     */\n\n    // Classes that extend HashSet and implement `build()` (freezing) operation\n    // have to make sure mutating methods check `checkIsMutable`\n\n    public actual open class HashSet<E> : AbstractMutableSet<E>, MutableSet<E> {\n        internal val map: HashMap<E, Any>\n\n        /**\n         * Constructs a new empty [HashSet].\n         */\n        actual constructor() {\n            map = HashMap<E, Any>()\n        }\n\n        /**\n         * Constructs a new [HashSet] filled with the elements of the specified collection.\n         */\n        actual constructor(elements: Collection<E>) {\n            map = HashMap<E, Any>(elements.size)\n            addAll(elements)\n        }\n\n        /**\n         * Constructs a new empty [HashSet].\n         */\n        /**\n         * @param initialCapacity the initial capacity (ignored)\n         * @param loadFactor the load factor (ignored)\n         */\n        /**\n         * @throws IllegalArgumentException if the initial capacity or load factor are negative\n         */\n        actual constructor(initialCapacity: Int, loadFactor: Float) {\n            map = HashMap<E, Any>(initialCapacity, loadFactor)\n        }\n\n        actual constructor(initialCapacity: Int) : this(initialCapacity, 0.0f)\n\n        /**\n         * Protected constructor to specify the underlying map. This is used by\n         * LinkedHashSet.\n         */\n        /**\n         * @param map underlying map to use.\n         */\n        internal constructor(map: HashMap<E, Any>) {\n            this.map = map\n        }\n\n        actual override fun add(element: E): Boolean {\n            val old = map.put(element, this)\n            return old == null\n        }\n\n        actual override fun clear() {\n            map.clear()\n        }\n\n        public override fun clone(): Any {\n            return HashSet<E>(this)\n        }\n\n        actual override operator fun contains(element: E): Boolean = map.containsKey(element)\n\n        actual override fun isEmpty(): Boolean = map.isEmpty()\n\n        actual override fun iterator(): MutableIterator<E> = map.keys.iterator()\n\n        actual override fun remove(element: E): Boolean = map.remove(element) != null\n\n        actual override val size: Int get() = map.size\n    }\n\n    /**\n     * Creates a new instance of the specialized implementation of [HashSet] with the specified [String] elements,\n     * which elements the keys as properties of JS object without hashing them.\n     */\n\n    public fun stringSetOf(vararg elements: String): HashSet<String> {\n        return HashSet(stringMapOf<Any>()).apply { addAll(elements) }\n    }\n\n    "\n    /*\n     * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n     * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n     */\n\n    /*\n     * Based on GWT InternalHashCodeMap\n     * Copyright 2008 Google Inc.\n     */\n\n    package kotlin.collections\n\n    import kotlin.collections.MutableMap.MutableEntry\n    import kotlin.collections.AbstractMutableMap.SimpleEntry\n\n    /**\n     * A simple wrapper around JavaScriptObject to provide [java.util.Map]-like semantics for any\n     * key type.\n     */\n\n    /**\n     * Implementation notes:\n     */\n\n    /**\n     * A key's hashCode is the index in backingMap which should contain that key. Since several keys may\n     * have the same hash, each value in hashCodeMap is actually an array containing all entries whose\n     * keys share the same hash.\n     */\n\n    internal class InternalHashCodeMap<K, V>(override val equality: EqualityComparator) : InternalMap<K, V> {\n        private var backingMap: dynamic = createJsMap()\n        override var size: Int = 0\n        private set\n\n        override fun put(key: K, value: V): V? {\n            val hashCode = equality.getHashCode(key)\n            val chainOrEntry

```



```

entry we use includes next/prev pointers for a doubly-linked circular
 * list with a head node. This reduces the
special cases we have to deal with
 * in the list operations.
 * Note that we duplicate the key from the
underlying hash map so we can find
 * the eldest entry. The alternative would have been to modify HashMap so
more
 * of the code was directly usable here, but this would have added some
 * overhead to HashMap, or to
reimplement most of the HashMap code here with
 * small modifications. Paying a small storage cost only if
you use
 * LinkedHashMap and minimizing code size seemed like a better tradeoff
 *
private inner class
ChainEntry<K, V>(key: K, value: V) : AbstractMutableMap.SimpleEntry<K, V>(key, value) {
    internal var
next: ChainEntry<K, V>? = null
    internal var prev: ChainEntry<K, V>? = null
    override fun
setValue(newValue: V): V {
        this@LinkedHashMap.checkIsMutable()
        return
super.setValue(newValue)
    }
}
private inner class EntrySet : AbstractEntrySet<MutableEntry<K,
V>, K, V>() {
    private inner class EntryIterator : MutableIterator<MutableEntry<K, V>> {
        // The
last entry that was returned from this iterator.
        private var last: ChainEntry<K, V>? = null
        // The
next entry to return from this iterator.
        private var next: ChainEntry<K, V>? = null
        init {
            next = head
        }
        recordLastKnownStructure(map, this)
        override fun hasNext():
Boolean {
            return next != null
        }
        override fun next(): MutableEntry<K, V> {
            //
checkStructuralChange(map, this)
            if (!hasNext()) throw NoSuchElementException()
            val
current = next!!
            last = current
            next = current.next.takeIf { it != head }
            return
current
        }
        override fun remove() {
            check(last != null)
            this@EntrySet.checkIsMutable()
            checkStructuralChange(map, this)
            last!!.remove()
            map.remove(last!!.key)
            recordLastKnownStructure(map, this)
            last = null
        }
    }
    override fun add(element: MutableEntry<K, V>): Boolean = throw
UnsupportedOperationException("Add is not supported on entries")
    override fun clear() {
        this@LinkedHashMap.clear()
    }
    override fun containsEntry(element: Map.Entry<K, V>): Boolean =
this@LinkedHashMap.containsEntry(element)
    override operator fun iterator():
MutableIterator<MutableEntry<K, V>> = EntryIterator()
    override fun removeEntry(element: Map.Entry<K,
V>): Boolean {
        checkIsMutable()
        if (contains(element)) {
            this@LinkedHashMap.remove(element.key)
            return true
        }
        return false
    }
    override val size: Int get() = this@LinkedHashMap.size
    override fun checkIsMutable(): Unit =
this@LinkedHashMap.checkIsMutable()
}
/*
 * The head of the insert order chain, which is a doubly-
linked circular
 * list.
 * The most recently inserted node is at the end of the chain, ie.
 * chain.prev.
 */
private var head: ChainEntry<K, V>? = null
/**
 * Add this node to the end of the chain.
 */
private fun ChainEntry<K, V>.addToEnd() {
    // This entry is not in the list.
    check(next == null && prev
== null)
    val _head = head
    if (_head == null) {
        head = this
        next = this
        prev =
this
    } else {
        // Chain is valid.
        val _tail = checkNotNull(_head.prev)
        // Update me.
        prev = _tail
        next = _head
        // Update my new siblings: current head and old tail
        _head.prev = this
        _tail.next = this
    }
}
/**
 * Remove this node from the chain it is a part
of.
 */
private fun ChainEntry<K, V>.remove() {
    if (this.next === this) {
        // if this is single
element, remove head
        head = null
    } else {
        if (head === this) {
            // if this is first
element, move head to next
            head = next
        }
        next!!.prev = prev
        prev!!.next =
next
    }
    next = null
    prev = null
}
/*
 * The hashmap that keeps track of our entries and
the chain. Note that we
 * duplicate the key here to eliminate changes to HashMap and minimize the
 * code
here, at the expense of additional space.
 */
private val map: HashMap<K, ChainEntry<K, V>>
private
var isReadOnly: Boolean = false
/**
 * Constructs an empty [LinkedHashMap] instance.
 */
actual
constructor() : super() {
    map = HashMap<K, ChainEntry<K, V>>()
}
internal
constructor(backingMap: HashMap<K, Any>) : super() {
    @Suppress("UNCHECKED_CAST") // expected
to work due to erasure
    map = backingMap as HashMap<K, ChainEntry<K, V>>
}
/**
 *
Constructs an empty [LinkedHashMap] instance.
 *
 * @param initialCapacity the initial capacity
(ignored)
 * @param loadFactor the load factor (ignored)
 *
 * @throws IllegalArgumentException if

```



```

inline fun <R> synchronized(lock: Any, block: () -> R): R {
    contract {
        callsInPlace(block,
        InvocationKind.EXACTLY_ONCE)
    }
    return block()
}

/*
 * Copyright 2010-2018 JetBrains s.r.o. and
 * Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that
 * can be found in the license/LICENSE.txt file.
 */
package kotlin.io
internal abstract class BaseOutput {
    open fun println() {
        print("\n")
    }
    open fun println(message: Any?) {
        print(message)
        println()
    }
    abstract fun print(message: Any?)
    open fun flush() {}
}

/** JsName used to make the
 * declaration available outside of module to test it */
@JsName("NodeJsOutput")
internal class NodeJsOutput(val
    outputStream: dynamic) : BaseOutput() {
    override fun print(message: Any?) {
        // TODO: Using local
        variable because of bug in block decomposition lowering in IR backend
        val messageString =
        String(message)
        outputStream.write(messageString)
    }
}

/** JsName used to make the declaration
 * available outside of module to test it */
@JsName("OutputToConsoleLog")
internal class OutputToConsoleLog
: BaseOutput() {
    override fun print(message: Any?) {
        console.log(message)
    }
    override fun
    println(message: Any?) {
        console.log(message)
    }
    override fun println() {
        console.log("")
    }
}

/** JsName used to make the declaration available outside of module to test it and use at try.kotl.in
 */
@JsName("BufferedOutput")
internal open class BufferedOutput : BaseOutput() {
    var buffer = ""
    override fun print(message: Any?) {
        buffer += String(message)
    }
    override fun flush() {
        buffer = ""
    }
}

/** JsName used to make the declaration available outside of module to test it
 */
@JsName("BufferedOutputToConsoleLog")
internal class BufferedOutputToConsoleLog : BufferedOutput() {
    override fun print(message: Any?) {
        var s = String(message)
        val i = s.nativeLastIndexOf("\n",
        0)
        if (i >= 0) {
            buffer += s.substring(0, i)
            flush()
            s = s.substring(i + 1)
        }
        buffer += s
    }
    override fun flush() {
        console.log(buffer)
        buffer = ""
    }
}

/** JsName
 * used to make the declaration available outside of module to test it and use at try.kotl.in
 */
@JsName("output")
internal var output = run {
    val isNode: Boolean = js("typeof process !== 'undefined'
    && process.versions && !!process.versions.node")
    if (isNode) NodeJsOutput(js("process.stdout")) else
    BufferedOutputToConsoleLog()
}

@kotlin.internal.InlineOnly
private inline fun String(value: Any?): String =
    js("String")(value)

/** Prints the line separator to the standard output stream. */
public actual fun println() {
    output.println()
}

/** Prints the given [message] and the line separator to the standard output stream. */
public actual fun println(message: Any?) {
    output.println(message)
}

/** Prints the given [message] to the standard
 * output stream. */
public actual fun print(message: Any?) {
    output.print(message)
}

@SinceKotlin("1.6")
public actual fun readln(): String = throw
UnsupportedOperationException("readln is not supported in Kotlin/JS")

@SinceKotlin("1.6")
public actual
fun readlnOrNull(): String? = throw UnsupportedOperationException("readlnOrNull is not supported in
Kotlin/JS")

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use
 * of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */
package kotlin.coroutines
import kotlin.coroutines.intrinsics.CoroutineSingletons
import
kotlin.coroutines.intrinsics.COROUTINE_SUSPENDED
@PublishedApi
@SinceKotlin("1.3")
internal
actual class SafeContinuation<in T>
internal actual constructor(
    private val delegate: Continuation<T>,
    initialResult: Any?) : Continuation<T> {
    @PublishedApi
    internal actual constructor(delegate:
    Continuation<T>) : this(delegate, UNDECIDED)
    public actual override val context: CoroutineContext
    get() = delegate.context
    private var result: Any? = initialResult
    public actual override fun
    resumeWith(result: Result<T>) {
        val cur = this.result
        when {
            cur === UNDECIDED -> {
                this.result = result.value
            }
            cur === COROUTINE_SUSPENDED -> {
                this.result =
                RESUMED
                delegate.resumeWith(result)
            }
            else -> throw
            IllegalStateException("Already resumed")
        }
    }
    @PublishedApi
    internal actual fun
    getOrThrow(): Any? {
        if (result === UNDECIDED) {
            result = COROUTINE_SUSPENDED
            return COROUTINE_SUSPENDED
        }
        val result = this.result
        return when {
            result ===
            RESUMED -> COROUTINE_SUSPENDED // already called continuation, indicate COROUTINE_SUSPENDED
            upstream
            result is Result.Failure -> throw result.exception
            else -> result // either
        }
    }
}

```



```

Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.dom\n\nimport
org.w3c.dom.Document\nimport org.w3c.dom.Element\nimport
kotlin.internal.LowPriorityInOverloadResolution\nimport kotlin.dom.appendElement as
newAppendElement\nimport kotlin.dom.createElement as newCreateElement\n\n/**\n * Creates a new element
with the specified [name].\n *\n * The element is initialized with the specified [init] function.\n
*\n * @LowPriorityInOverloadResolution\n * @Deprecated(\n     message = `\"This API is moved to another package,
use 'kotlinx.dom.createElement' instead.`,\n     replaceWith = ReplaceWith(`\"this.createElement(name, init)`\",
`\"kotlinx.dom.createElement`\")\n)\n * @DeprecatedSinceKotlin(warningSince = `\"1.4`\", errorSince = `\"1.6`\")\npublic
inline fun Document.createElement(name: String, noinline init: Element.() -> Unit): Element =
this.newCreateElement(name, init)\n\n/**\n * Appends a newly created element with the specified [name] to this
element.\n *\n * The element is initialized with the specified [init] function.\n
*\n * @LowPriorityInOverloadResolution\n * @Deprecated(\n     message = `\"This API is moved to another package,
use 'kotlinx.dom.appendElement' instead.`,\n     replaceWith = ReplaceWith(`\"this.appendElement(name, init)`\",
`\"kotlinx.dom.appendElement`\")\n)\n * @DeprecatedSinceKotlin(warningSince = `\"1.4`\", errorSince = `\"1.6`\")\npublic
inline fun Element.appendElement(name: String, noinline init: Element.() -> Unit): Element =
this.newAppendElement(name, init)\n\n"/**\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\npackage kotlin.dom\n\nimport org.w3c.dom.Element\nimport
kotlin.internal.LowPriorityInOverloadResolution\nimport kotlin.dom.addClass as newAddClass\nimport
kotlin.dom.hasClass as newHasClass\nimport kotlin.dom.removeClass as newRemoveClass\n\n/** Returns true if
the element has the given CSS class style in its 'class' attribute
*\n * @LowPriorityInOverloadResolution\n * @Deprecated(\n     message = `\"This API is moved to another package,
use 'kotlinx.dom.hasClass' instead.`,\n     replaceWith = ReplaceWith(`\"this.hasClass(cssClass)`\",
`\"kotlinx.dom.hasClass`\")\n)\n * @DeprecatedSinceKotlin(warningSince = `\"1.4`\", errorSince = `\"1.6`\")\ninline fun
Element.hasClass(cssClass: String): Boolean = this.newHasClass(cssClass)\n\n/**\n * Adds CSS class to element.
Has no effect if all specified classes are already in class attribute of the element\n *\n * @return true if at least one
class has been added\n *\n * @LowPriorityInOverloadResolution\n * @Deprecated(\n     message = `\"This API is moved
to another package, use 'kotlinx.dom.addClass' instead.`,\n     replaceWith =
ReplaceWith(`\"this.addClass(cssClasses)`\", `\"kotlinx.dom.addClass`\")\n)\n * @DeprecatedSinceKotlin(warningSince
= `\"1.4`\", errorSince = `\"1.6`\")\ninline fun Element.addClass(vararg cssClasses: String): Boolean =
this.newAddClass(*cssClasses)\n\n/**\n * Removes all [cssClasses] from element. Has no effect if all specified
classes are missing in class attribute of the element\n *\n * @return true if at least one class has been removed\n
*\n * @LowPriorityInOverloadResolution\n * @Deprecated(\n     message = `\"This API is moved to another package,
use 'kotlinx.dom.removeClass' instead.`,\n     replaceWith = ReplaceWith(`\"this.removeClass(cssClasses)`\",
`\"kotlinx.dom.removeClass`\")\n)\n * @DeprecatedSinceKotlin(warningSince = `\"1.4`\", errorSince = `\"1.6`\")\ninline
fun Element.removeClass(vararg cssClasses: String): Boolean = this.newRemoveClass(*cssClasses)\n\n"/**\n *
Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is
governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage
kotlin.dom\n\nimport org.w3c.dom.Element\nimport org.w3c.dom.Node\nimport
kotlin.internal.LowPriorityInOverloadResolution\nimport kotlin.dom.isElement as newIsElement\nimport
kotlin.dom.isText as newIsText\n\n/**\n * Gets a value indicating whether this node is a TEXT_NODE or a
CDATA_SECTION_NODE.\n *\n * @LowPriorityInOverloadResolution\n * @Deprecated(\n     message = `\"This API
is moved to another package, use 'kotlinx.dom.isText' instead.`,\n     replaceWith = ReplaceWith(`\"this.isText`\",
`\"kotlinx.dom.isText`\")\n)\n * @DeprecatedSinceKotlin(warningSince = `\"1.4`\", errorSince = `\"1.6`\")\npublic val
Node.isText: Boolean\n     inline get() = this.newIsText\n\n/**\n * Gets a value indicating whether this node is an
[Element].\n *\n * @LowPriorityInOverloadResolution\n * @Deprecated(\n     message = `\"This API is moved to
another package, use 'kotlinx.dom.isElement' instead.`,\n     replaceWith = ReplaceWith(`\"this.isElement`\",
`\"kotlinx.dom.isElement`\")\n)\n * @DeprecatedSinceKotlin(warningSince = `\"1.4`\", errorSince = `\"1.6`\")\npublic val

```

```

Node.isElement: Boolean\n inline get() = this.newIsElement\n", "/*\n * Copyright 2010-2018 JetBrains s.r.o. and
Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file.\n */\n\npackage org.w3c.dom.events\n\npublic fun
EventListener(handler: (Event) -> Unit): EventListener = EventListenerHandler(handler)\n\nprivate class
EventListenerHandler(private val handler: (Event) -> Unit) : EventListener {\n public override fun
handleEvent(event: Event) {\n handler(event)\n }\n\n public override fun toString(): String =
`EventListenerHandler($handler)`\n}\n", "/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\npackage org.w3c.dom\n\npublic external interface ItemArrayLike<out T> {\n
val length: Int\n fun item(index: Int): T?\n}\n\n/**\n * Returns the view of this `ItemArrayLike<T>` collection as
`List<T>`\n */\n\npublic fun <T> ItemArrayLike<T>.asList(): List<T> = object : AbstractList<T>() {\n override val
size: Int get() = this@asList.length\n\n override fun get(index: Int): T = when (index) {\n in 0..lastIndex ->
this@asList.item(index).unsafeCast<T>()\n else -> throw IndexOutOfBoundsException("index $index is not in
range [0..$lastIndex]")\n }\n}\n", "/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\npackage kotlin.dom\n\nimport org.w3c.dom.Element\nimport
org.w3c.dom.Node\nimport kotlin.internal.LowPriorityInOverloadResolution\nimport kotlin.dom.appendText as
newAppendText\nimport kotlin.dom.clear as newClear\n\n/** Removes all the children from this node.
*\n */\n\n@LowPriorityInOverloadResolution\n@Deprecated(\n message = `This API is moved to another package,
use 'kotlinx.dom.clear' instead.`,\n replaceWith = ReplaceWith("this.clear()"),
`kotlinx.dom.clear`)\n)\n\n@DeprecatedSinceKotlin(warningSince = `1.4`, errorSince = `1.6`)\n\npublic inline fun
Node.clear() = this.newClear()\n\n/**\n * Creates text node and append it to the element.\n */\n\n@return this
element\n */\n\n@LowPriorityInOverloadResolution\n@Deprecated(\n message = `This API is moved to another
package, use 'kotlinx.dom.appendText' instead.`,\n replaceWith = ReplaceWith("this.appendText(text)"),
`kotlinx.dom.appendText`)\n)\n\n@DeprecatedSinceKotlin(warningSince = `1.4`, errorSince = `1.6`)\n\ninline fun
Element.appendText(text: String): Element = this.newAppendText(text)\n", "/*\n * Copyright 2010-2018 JetBrains
s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0
license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.js\n\n/**\n * Reinterprets this value
as a value of the [dynamic type](/docs/reference/dynamic-type.html).\n */\n\n@kotlin.internal.InlineOnly\n\npublic
inline fun Any?.asDynamic(): dynamic = this\n\n/**\n * Reinterprets this value as a value of the specified type [T]
without any actual type checking.\n */\n\n@kotlin.internal.InlineOnly\n\npublic inline fun <T> Any?.unsafeCast():
@kotlin.internal.NoInfer T = this.asDynamic()\n\n/**\n * Reinterprets this `dynamic` value as a value of the
specified type [T] without any actual type checking.\n */\n\n@kotlin.internal.DynamicExtension\n\n@JsName("unsafeCastDynamic")\n\n@kotlin.internal.InlineOnly\n\npublic
inline fun <T> dynamic.unsafeCast(): @kotlin.internal.NoInfer T = this\n\n/**\n * Allows to iterate this `dynamic`
object in the following cases:\n * - when it has an `iterator` function,\n * - when it is an array\n * - when it is an
instance of [kotlin.collections.Iterable]\n */\n\n@kotlin.internal.DynamicExtension\n\npublic operator fun
dynamic.iterator(): Iterator<dynamic> {\n val r: Any? = this\n\n return when {\n this["iterator"] != null ->
this["iterator"]()\n isArrayish(r) -> r.unsafeCast<Array<*>>().iterator()\n\n else ->
(r as Iterable<*>).iterator()\n }\n}\n", "/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\n// a package is omitted to get declarations directly under the
module\n\n@JsName("throwNPE")\n\ninternal fun throwNPE(message: String) {\n throw
NullPointerException(message)\n}\n\n@JsName("throwCCE")\n\ninternal fun throwCCE() {\n throw
ClassCastException("Illegal cast")\n}\n\n@JsName("throwISE")\n\ninternal fun throwISE(message: String) {\n
throw IllegalStateException(message)\n}\n\n@JsName("throwUPAE")\n\ninternal fun throwUPAE(propertyName:
String) {\n throw UninitializedPropertyAccessException("lateinit property ${propertyName} has not been
initialized")\n}\n", "/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n

```


arguments, and stores the results in a new map.

- An initial value of accumulator is provided by [initialValueSelector] function.
- It's invoked with parameters:
 - key: the key of the group;
 - element: the first element being encountered in that group.
- operation a function that is invoked on each element with the following parameters:
 - key: the key of the group this element belongs to;
 - accumulator: the current value of the accumulator of the group;
 - element: the element from the source being accumulated.

@return a [Map] associating the key of each group with the result of accumulating the group elements.

@sample samples.collections.Grouping.foldByEvenLengthWithComputedInitialValue

```

*\/n@SinceKotlin("1.1")\npublic inline fun <T, K, R> Grouping<T, K>.fold(\n    initialValueSelector: (key: K,\n    element: T) -> R,\n    operation: (key: K, accumulator: R, element: T) -> R\n): Map<K, R> =\n    @Suppress("UNCHECKED_CAST")\n    aggregate { key, acc, e, first -> operation(key, if (first)\n    initialValueSelector(key, e) else acc as R, e) }\n/n/n/**\n * Groups elements from the [Grouping] source by key and\n * applies [operation] to the elements of each group sequentially,\n * passing the previously accumulated value and the\n * current element as arguments,\n * and stores the results in the given [destination] map.\n * An initial value of\n * accumulator is provided by [initialValueSelector] function.\n * It's invoked with parameters:\n * - `key`: the key of the\n * group;\n * - `element`: the first element being encountered in that group.\n * If the [destination] map already has\n * a value corresponding to some key, that value is used as an initial value of\n * the accumulator for that group and the\n * [initialValueSelector] function is not called for that group.\n * operation a function that is invoked on\n * each element with the following parameters:\n * - `key`: the key of the group this element belongs to;\n * -\n * `accumulator`: the current value of the accumulator of the group;\n * - `element`: the element from the source being\n * accumulated.\n * @return the [destination] map associating the key of each group with the result of\n * accumulating the group elements.\n * @sample\n * samples.collections.Grouping.foldByEvenLengthWithComputedInitialValueTo\n */\n@SinceKotlin("1.1")\npublic\n * inline fun <T, K, R, M : MutableMap<in K, R>> Grouping<T, K>.foldTo(\n * destination: M,\n * initialValueSelector: (key: K, element: T) -> R,\n * operation: (key: K, accumulator: R, element: T) -> R\n * ): M =\n * @Suppress("UNCHECKED_CAST")\n * aggregateTo(destination) { key, acc, e, first -> operation(key, if (first)\n * initialValueSelector(key, e) else acc as R, e) }\n/n/n/**\n * Groups elements from the [Grouping] source by key and\n * applies [operation] to the elements of each group sequentially,\n * passing the previously accumulated value and the\n * current element as arguments, and stores the results in a new map.\n * An initial value of accumulator is the same\n * [initialValue] for each group.\n * operation a function that is invoked on each element with the\n * following parameters:\n * - `accumulator`: the current value of the accumulator of the group;\n * - `element`: the\n * element from the source being accumulated.\n * @return a [Map] associating the key of each group with the\n * result of accumulating the group elements.\n * @sample\n * samples.collections.Grouping.foldByEvenLengthWithConstantInitialValue\n */\n@SinceKotlin("1.1")\npublic\n * inline fun <T, K, R> Grouping<T, K>.fold(\n * initialValue: R,\n * operation: (accumulator: R, element: T) -> R\n * ): Map<K, R> =\n * @Suppress("UNCHECKED_CAST")\n * aggregate { _, acc, e, first -> operation(if (first)\n * initialValue else acc as R, e) }\n/n/n/**\n * Groups elements from the [Grouping] source by key and applies\n * [operation] to the elements of each group sequentially,\n * passing the previously accumulated value and the current\n * element as arguments,\n * and stores the results in the given [destination] map.\n * An initial value of accumulator is\n * the same [initialValue] for each group.\n * If the [destination] map already has a value corresponding to the key\n * of some group,\n * that value is used as an initial value of the accumulator for that group.\n * operation\n * a function that is invoked on each element with the following parameters:\n * - `accumulator`: the current value of\n * the accumulator of the group;\n * - `element`: the element from the source being accumulated.\n * @return the\n * [destination] map associating the key of each group with the result of accumulating the group elements.\n * @sample\n * samples.collections.Grouping.foldByEvenLengthWithConstantInitialValueTo\n */\n@SinceKotlin("1.1")\npublic inline fun <T, K, R, M : MutableMap<in K, R>> Grouping<T, K>.foldTo(\n * destination: M,\n * initialValue: R,\n * operation: (accumulator: R, element: T) -> R\n * ): M =\n
```

```

@Suppress("UNCHECKED_CAST")\n    aggregateTo(destination) { _, acc, e, first -> operation(if (first)
initialValue else acc as R, e) }\n\n/**\n * Groups elements from the [Grouping] source by key and applies the
reducing [operation] to the elements of each group\n * sequentially starting from the second element of the group,\n * passing the previously accumulated value and the current element as arguments,\n * and stores the results in a new
map.\n * An initial value of accumulator is the first element of the group.\n * @param operation a function that
is invoked on each subsequent element of the group with the following parameters:\n * - `key`: the key of the group
this element belongs to;\n * - `accumulator`: the current value of the accumulator of the group;\n * - `element`: the
element from the source being accumulated.\n * @return a [Map] associating the key of each group with the
result of accumulating the group elements.\n * @sample samples.collections.Grouping.reduceByMaxVowels\n
*\n@SinceKotlin("1.1")\npublic inline fun <S, T : S, K> Grouping<T, K>.reduce(\n    operation: (key: K,
accumulator: S, element: T) -> S)\n): Map<K, S> =\n    aggregate { key, acc, e, first ->\n
@Suppress("UNCHECKED_CAST")\n        if (first) e else operation(key, acc as S, e)\n    }\n\n/**\n * Groups
elements from the [Grouping] source by key and applies the reducing [operation] to the elements of each group\n *
sequentially starting from the second element of the group,\n * passing the previously accumulated value and the
current element as arguments,\n * and stores the results in the given [destination] map.\n * An initial value of
accumulator is the first element of the group.\n * If the [destination] map already has a value corresponding to
the key of some group,\n * that value is used as an initial value of the accumulator for that group and the first
element of that group is also\n * subjected to the [operation].\n * @param operation a function that is invoked on
each subsequent element of the group with the following parameters:\n * - `accumulator`: the current value of the
accumulator of the group;\n * - `element`: the element from the source being folded;\n * @return the
[destination] map associating the key of each group with the result of accumulating the group elements.\n *
@sample samples.collections.Grouping.reduceByMaxVowelsTo\n
*\n@SinceKotlin("1.1")\npublic inline fun <S,
T : S, K, M : MutableMap<in K, S>> Grouping<T, K>.reduceTo(\n    destination: M,\n    operation: (key: K,
accumulator: S, element: T) -> S)\n): M =\n    aggregateTo(destination) { key, acc, e, first ->\n
@Suppress("UNCHECKED_CAST")\n        if (first) e else operation(key, acc as S, e)\n    }\n\n/**\n * Groups
elements from the [Grouping] source by key and counts elements in each group to the given [destination] map.\n *
If the [destination] map already has a value corresponding to the key of some group,\n * that value is used as an
initial value of the counter for that group.\n * @return the [destination] map associating the key of each group
with the count of elements in the group.\n * @sample samples.collections.Grouping.groupingByEachCount\n
*\n@SinceKotlin("1.1")\npublic fun <T, K, M : MutableMap<in K, Int>> Grouping<T,
K>.eachCountTo(destination: M): M =\n    foldTo(destination, 0) { acc, _ -> acc + 1 }\n\n/**\n * Groups
elements from the [Grouping] source by key and sums values provided by the [valueSelector] function for elements
in each group\n * to the given [destination] map.\n * If the [destination] map already has a value
corresponding to the key of some group,\n * that value is used as an initial value of the sum for that group.\n *
@return the [destination] map associating the key of each group with the sum of elements in the group.\n
*\n@SinceKotlin("1.1")\npublic inline fun <T, K, M : MutableMap<in K, Int>> Grouping<T,
K>.eachSumOfTo(destination: M, valueSelector: (T) -> Int): M =\n    foldTo(destination, 0) { acc, e -> acc +
valueSelector(e) }\n\n/**\n * TODO: sum by long and by double overloads\n\npublic inline fun <T, K, M :
MutableMap<in K, Long>> Grouping<T, K>.sumEachByLongTo(destination: M, valueSelector: (T) -> Long): M
=\n    foldTo(destination, 0L) { acc, e -> acc + valueSelector(e) }\n\npublic inline fun <T, K> Grouping<T,
K>.sumEachByLong(valueSelector: (T) -> Long): Map<K, Long> =\n    fold(0L) { acc, e -> acc +
valueSelector(e) }\n\npublic inline fun <T, K, M : MutableMap<in K, Double>> Grouping<T,
K>.sumEachByDoubleTo(destination: M, valueSelector: (T) -> Double): M =\n    foldTo(destination, 0.0) { acc, e
-> acc + valueSelector(e) }\n\npublic inline fun <T, K> Grouping<T, K>.sumEachByDouble(valueSelector: (T) ->
Double): Map<K, Double> =\n    fold(0.0) { acc, e -> acc + valueSelector(e) }\n\n"/*\n * Copyright 2010-2021
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage
kotlin.js\n\n@Retention(AnnotationRetention.BINARY)\n@Target(AnnotationTarget.FUNCTION,

```


`nativeMath.atan(x)` Returns the angle θ of the polar coordinates (r, θ) that correspond to the rectangular coordinates (x, y) by computing the arc tangent of the value y / x ; the returned value is an angle in the range from $-\pi$ to π radians. Special cases: $\text{atan}(0.0, 0.0)$ is 0.0 ; $\text{atan}(0.0, x)$ is 0.0 for $x > 0$ and π for $x < 0$; $\text{atan}(-0.0, x)$ is -0.0 for $x > 0$ and $-\pi$ for $x < 0$; $\text{atan}(y, +\text{Inf})$ is 0.0 for $0 < y < +\text{Inf}$ and -0.0 for $-\text{Inf} < y < 0$; $\text{atan}(y, -\text{Inf})$ is π for $0 < y < +\text{Inf}$ and $-\pi$ for $-\text{Inf} < y < 0$; $\text{atan}(y, 0.0)$ is $\pi/2$ for $y > 0$ and $-\pi/2$ for $y < 0$; $\text{atan}(+\text{Inf}, x)$ is $\pi/2$ for finite x ; $\text{atan}(-\text{Inf}, x)$ is $-\pi/2$ for finite x ; $\text{atan}(\text{NaN}, x)$ and $\text{atan}(y, \text{NaN})$ is NaN .


```

    @SinceKotlin("1.2")
    @InlineOnly
    public actual inline fun atan2(y: Double, x: Double): Double =
        nativeMath.atan2(y, x)
    
```

`nativeMath.atanh(x)` Computes the inverse hyperbolic tangent of the value x . Special cases: $\text{atanh}(\text{NaN})$ is NaN ; $\text{atanh}(+\text{Inf})$ is $+\text{Inf}$; $\text{atanh}(-\text{Inf})$ is $-\text{Inf}$.


```

    @SinceKotlin("1.2")
    @InlineOnly
    public actual inline fun atanh(x: Double): Double =
        nativeMath.atanh(x)
    
```

`nativeMath.cosh(x)` Computes the hyperbolic cosine of the value x . Special cases: $\text{cosh}(\text{NaN})$ is NaN ; $\text{cosh}(+\text{Inf})$ is $+\text{Inf}$; $\text{cosh}(-\text{Inf})$ is $+\text{Inf}$.


```

    @SinceKotlin("1.2")
    @InlineOnly
    public actual inline fun cosh(x: Double): Double = nativeMath.cosh(x)
    
```

`nativeMath.cosh(x)` Computes the hyperbolic cosine of the value x . Special cases: $\text{cosh}(\text{NaN})$ is NaN ; $\text{cosh}(+\text{Inf})$ is $+\text{Inf}$; $\text{cosh}(-\text{Inf})$ is $+\text{Inf}$.


```

    @SinceKotlin("1.2")
    @InlineOnly
    public actual inline fun cosh(x: Double): Double = nativeMath.cosh(x)
    
```

`nativeMath.tanh(x)` Computes the hyperbolic tangent of the value x . Special cases: $\text{tanh}(\text{NaN})$ is NaN ; $\text{tanh}(+\text{Inf})$ is 1.0 ; $\text{tanh}(-\text{Inf})$ is -1.0 .


```

    @SinceKotlin("1.2")
    @InlineOnly
    public actual inline fun tanh(x: Double): Double =
        nativeMath.tanh(x)
    
```

`nativeMath.asinh(x)` Computes the inverse hyperbolic sine of the value x . The returned value is y such that $\sinh(y) = x$. Special cases: $\text{asinh}(\text{NaN})$ is NaN ; $\text{asinh}(+\text{Inf})$ is $+\text{Inf}$; $\text{asinh}(-\text{Inf})$ is $-\text{Inf}$.


```

    @SinceKotlin("1.2")
    @InlineOnly
    public actual inline fun asinh(x: Double): Double =
        nativeMath.asinh(x)
    
```

`nativeMath.acosh(x)` Computes the inverse hyperbolic cosine of the value x . The returned value is positive y such that $\cosh(y) = x$. Special cases: $\text{acosh}(\text{NaN})$ is NaN ; $\text{acosh}(x)$ is NaN when $x < 1$; $\text{acosh}(+\text{Inf})$ is $+\text{Inf}$.


```

    @SinceKotlin("1.2")
    @InlineOnly
    public actual inline fun acosh(x: Double): Double = nativeMath.acosh(x)
    
```

`nativeMath.atanh(x)` Computes the inverse hyperbolic tangent of the value x . The returned value is y such that $\tanh(y) = x$. Special cases: $\text{atanh}(\text{NaN})$ is NaN ; $\text{atanh}(x)$ is NaN when $x > 1$ or $x < -1$; $\text{atanh}(1.0)$ is $+\text{Inf}$; $\text{atanh}(-1.0)$ is $-\text{Inf}$.


```

    @SinceKotlin("1.2")
    @InlineOnly
    public actual inline fun atanh(x: Double): Double =
        nativeMath.atanh(x)
    
```

`nativeMath.hypot(x, y)` Computes $\sqrt{x^2 + y^2}$ without intermediate overflow or underflow. Special cases: returns $+\text{Inf}$ if any of arguments is infinite; returns NaN if any of arguments is NaN and the other is not infinite.


```

    @SinceKotlin("1.2")
    @InlineOnly
    public actual inline fun hypot(x: Double, y: Double): Double = nativeMath.hypot(x, y)
    
```

`nativeMath.sqrt(x)` Computes the positive square root of the value x . Special cases: \sqrt{x} is NaN when $x < 0$ or x is NaN .


```

    @SinceKotlin("1.2")
    @InlineOnly
    public actual inline fun sqrt(x: Double): Double =
        nativeMath.sqrt(x)
    
```

`nativeMath.exp(x)` Computes Euler's number e raised to the power of the value x . Special cases: $\text{exp}(\text{NaN})$ is NaN ; $\text{exp}(+\text{Inf})$ is $+\text{Inf}$; $\text{exp}(-\text{Inf})$ is 0.0 .


```

    @SinceKotlin("1.2")
    @InlineOnly
    public actual inline fun exp(x: Double): Double =
        nativeMath.exp(x)
    
```

`nativeMath.expm1(x)` Computes $\exp(x) - 1$. This function can be implemented to produce more precise result for x near zero. Special cases: $\text{expm1}(\text{NaN})$ is NaN ; $\text{expm1}(+\text{Inf})$ is $+\text{Inf}$; $\text{expm1}(-\text{Inf})$ is -1.0 . @see [exp] function.


```

    @SinceKotlin("1.2")
    @InlineOnly
    public actual inline fun expm1(x: Double): Double = nativeMath.expm1(x)
    
```

`nativeMath.log(x, base)` Computes the logarithm of the value x to the given [base]. Special cases: $\log(x, b)$ is NaN if either x or b are NaN ; $\log(x, b)$ is NaN when $x < 0$ or $b \leq 0$ or $b = 1.0$; $\log(+\text{Inf}, +\text{Inf})$ is NaN ; $\log(+\text{Inf}, b)$ is $+\text{Inf}$ for $b > 1$ and $-\text{Inf}$ for $b < 1$; $\log(0.0, b)$ is $-\text{Inf}$ for $b > 1$ and $+\text{Inf}$ for $b > 1$. See also logarithm functions for common fixed bases: [ln], [log10] and [log2].


```

    @SinceKotlin("1.2")
    public actual fun log(x: Double, base: Double): Double {
        if (base <= 0.0 || base == 1.0) return Double.NaN
        return nativeMath.log(x) / nativeMath.log(base)
    }
    
```

`nativeMath.log(x)` Computes the natural logarithm (base E) of the value x . Special cases: $\ln(\text{NaN})$ is NaN ; $\ln(x)$ is NaN when $x < 0.0$; $\ln(+\text{Inf})$ is $+\text{Inf}$; $\ln(0.0)$ is $-\text{Inf}$.


```

    @SinceKotlin("1.2")
    @InlineOnly
    public actual inline fun ln(x: Double): Double =
        nativeMath.log(x)
    
```

`nativeMath.log(x)` Computes the common logarithm (base 10) of the value x . @see [ln] function


```

    @SinceKotlin("1.2")
    @InlineOnly
    public actual inline fun log(x: Double): Double =
        nativeMath.log(x)
    
```

for special cases.

```

@SinceKotlin("1.2")@InlineOnly@public actual inline fun log10(x: Double): Double =
    nativeMath.log10(x)

```

Computes the binary logarithm (base 2) of the value [x].

```

@SinceKotlin("1.2")@InlineOnly@public actual inline fun log2(x: Double): Double =
    nativeMath.log2(x)

```

Computes $\ln(x + 1)$.

This function can be implemented to produce more precise result for [x] near zero.

Special cases:

- $-\ln1p(\text{NaN})$ is NaN
- $-\ln1p(x)$ is NaN where $x < -1.0$
- $-\ln1p(-1.0)$ is $-\text{Inf}$
- $-\ln1p(+\text{Inf})$ is $+\text{Inf}$

See [Ln] function

```

@SinceKotlin("1.2")@InlineOnly@public actual inline fun ln1p(x: Double): Double =
    nativeMath.log1p(x)

```

Rounds the given value [x] to an integer towards positive infinity.

Return the smallest double value that is greater than or equal to the given value [x] and is a mathematical integer.

Special cases:

- $\text{ceil}(x)$ is x where x is NaN or $+\text{Inf}$ or $-\text{Inf}$ or already a mathematical integer.

```

@SinceKotlin("1.2")@InlineOnly@public actual inline fun ceil(x: Double): Double =
    nativeMath.ceil(x)

```

Rounds the given value [x] to an integer towards negative infinity.

Return the largest double value that is smaller than or equal to the given value [x] and is a mathematical integer.

Special cases:

- $\text{floor}(x)$ is x where x is NaN or $+\text{Inf}$ or $-\text{Inf}$ or already a mathematical integer.

```

@SinceKotlin("1.2")@InlineOnly@public actual inline fun floor(x: Double): Double =
    nativeMath.floor(x)

```

Rounds the given value [x] to an integer towards zero.

Return the value [x] having its fractional part truncated.

Special cases:

- $\text{truncate}(x)$ is x where x is NaN or $+\text{Inf}$ or $-\text{Inf}$ or already a mathematical integer.

```

@SinceKotlin("1.2")@InlineOnly@public actual inline fun truncate(x: Double): Double =
    nativeMath.trunc(x)

```

Rounds the given value [x] towards the closest integer with ties rounded towards even integer.

Special cases:

- $\text{round}(x)$ is x where x is NaN or $+\text{Inf}$ or $-\text{Inf}$ or already a mathematical integer.

```

@SinceKotlin("1.2")@public actual fun round(x: Double): Double
{
    if (x % 0.5 != 0.0)
        return nativeMath.round(x)
    val floor = floor(x)
    return if (floor % 2 == 0.0) floor else ceil(x)
}

```

Returns the absolute value of the given value [x].

Special cases:

- $\text{abs}(\text{NaN})$ is NaN

See absoluteValue extension property for [Double]

```

@SinceKotlin("1.2")@InlineOnly@public actual inline fun abs(x: Double): Double =
    nativeMath.abs(x)

```

Returns the sign of the given value [x]:

- -1.0 if the value is negative,
- zero if the value is zero,
- 1.0 if the value is positive

Special case:

- $\text{sign}(\text{NaN})$ is NaN

```

@SinceKotlin("1.2")@InlineOnly@public actual inline fun sign(x: Double): Double =
    nativeMath.sign(x)

```

Returns the smaller of two values.

If either value is NaN , then the result is NaN .

```

@SinceKotlin("1.2")@InlineOnly@public actual inline fun min(a: Double, b: Double): Double =
    nativeMath.min(a, b)

```

Returns the greater of two values.

If either value is NaN , then the result is NaN .

```

@SinceKotlin("1.2")@InlineOnly@public actual inline fun max(a: Double, b: Double): Double =
    nativeMath.max(a, b)

```

Extensions

Raises this value to the power [x].

Special cases:

- $b.\text{pow}(0.0)$ is 1.0
- $b.\text{pow}(1.0) == b$
- $b.\text{pow}(\text{NaN})$ is NaN
- $\text{NaN}.\text{pow}(x)$ is NaN for $x != 0.0$
- $b.\text{pow}(\text{Inf})$ is NaN for $\text{abs}(b) == 1.0$
- $b.\text{pow}(x)$ is NaN for $b < 0$ and x is finite and not an integer

```

@SinceKotlin("1.2")@InlineOnly@public actual inline fun Double.pow(x: Double): Double =
    nativeMath.pow(this, x)

```

Raises this value to the integer power [n].

See the other overload of [pow] for details.

```

@SinceKotlin("1.2")@InlineOnly@public actual inline fun Double.pow(n: Int): Double =
    nativeMath.pow(this, n.toDouble())

```

Returns the absolute value of this value.

Special cases:

- $\text{NaN}.\text{absoluteValue}$ is NaN

See abs function

```

@SinceKotlin("1.2")@InlineOnly@public actual inline val Double.absoluteValue: Double get() =
    nativeMath.abs(this)

```

Returns the sign of this value:

- -1.0 if the value is negative,
- zero if the value is zero,
- 1.0 if the value is positive

Special case:

- $\text{NaN}.\text{sign}$ is NaN

```

@SinceKotlin("1.2")@InlineOnly@public actual inline val Double.sign: Double get() =
    nativeMath.sign(this)

```

Returns this value with the sign bit same as of the [sign] value.

```

@SinceKotlin("1.2")@InlineOnly@public actual inline fun Double.withSign(sign: Int):
    Double = this.withSign(sign.toDouble())

```

Returns the ulp (unit in the last place) of this value.

An ulp is a positive distance between this value and the next nearest [Double] value larger in magnitude.

Special Cases:

- $\text{NaN}.\text{ulp}$ is NaN
- $x.\text{ulp}$ is $+\text{Inf}$ when x is $+\text{Inf}$ or $-\text{Inf}$
- $0.0.\text{ulp}$ is

```

`Double.MIN_VALUE`\n *\n@SinceKotlin("1.2")\npublic actual val Double.ulp: Double get() = when {\n this
< 0 -> (-this).ulp\n this.isNaN() || this == Double.POSITIVE_INFINITY -> this\n this ==
Double.MAX_VALUE -> this - this.nextDown()\n else -> this.nextUp() - this\n}\n\n/**\n * Returns the [Double]
value nearest to this value in direction of positive infinity.\n *\n@SinceKotlin("1.2")\npublic actual fun
Double.nextUp(): Double = when {\n this.isNaN() || this == Double.POSITIVE_INFINITY -> this\n this == 0.0
-> Double.MIN_VALUE\n else -> Double.fromBits(this.toRawBits() + if (this > 0) 1 else -1)\n}\n\n/**\n *
Returns the [Double] value nearest to this value in direction of negative infinity.\n
*\n@SinceKotlin("1.2")\npublic actual fun Double.nextDown(): Double = when {\n this.isNaN() || this ==
Double.NEGATIVE_INFINITY -> this\n this == 0.0 -> -Double.MIN_VALUE\n else ->
Double.fromBits(this.toRawBits() + if (this > 0) -1 else 1)\n}\n\n/**\n * Returns the [Double] value nearest to this
value in direction from this value towards the value [to].\n *\n * Special cases:\n * - `x.nextTowards(y)` is `NaN` if
either `x` or `y` are `NaN`\n * - `x.nextTowards(x) == x`\n *\n@SinceKotlin("1.2")\npublic actual fun
Double.nextTowards(to: Double): Double = when {\n this.isNaN() || to.isNaN() -> Double.NaN\n to == this ->
to\n to > this -> this.nextUp()\n else /* to < this */ -> this.nextDown()\n}\n\n/**\n * Rounds this [Double]
value to the nearest integer and converts the result to [Int].\n *\n * Ties are rounded towards positive infinity.\n *\n *
Special cases:\n * - `x.roundToInt() == Int.MAX_VALUE` when `x > Int.MAX_VALUE`\n * - `x.roundToInt()
== Int.MIN_VALUE` when `x < Int.MIN_VALUE`\n *\n * @throws IllegalArgumentException when this value is
`NaN`\n *\n@SinceKotlin("1.2")\npublic actual fun Double.roundToInt(): Int = when {\n isNaN() -> throw
IllegalArgumentException("Cannot round NaN value.")\n this > Int.MAX_VALUE -> Int.MAX_VALUE\n this < Int.MIN_VALUE -> Int.MIN_VALUE\n else -> nativeMath.round(this).toInt()\n}\n\n/**\n * Rounds this
[Double] value to the nearest integer and converts the result to [Long].\n *\n * Ties are rounded towards positive
infinity.\n *\n * Special cases:\n * - `x.roundToLong() == Long.MAX_VALUE` when `x >
Long.MAX_VALUE`\n * - `x.roundToLong() == Long.MIN_VALUE` when `x < Long.MIN_VALUE`\n *\n * @throws
IllegalArgumentException when this value is `NaN`\n *\n@SinceKotlin("1.2")\npublic actual fun
Double.roundToLong(): Long = when {\n isNaN() -> throw IllegalArgumentException("Cannot round NaN
value.")\n this > Long.MAX_VALUE -> Long.MAX_VALUE\n this < Long.MIN_VALUE ->
Long.MIN_VALUE\n else -> nativeMath.round(this).toLong()\n}\n\n// endregion\n\n// region
==== Float Math =====\n\n/**\n * Computes the
sine of the angle [x] given in radians.\n *\n * Special cases:\n * - `sin(NaN|+Inf|-Inf)` is `NaN`\n
*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun sin(x: Float): Float =
nativeMath.sin(x.toDouble()).toFloat()\n\n/**\n * Computes the cosine of the angle [x] given in radians.\n *\n *
Special cases:\n * - `cos(NaN|+Inf|-Inf)` is `NaN`\n *\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun
cos(x: Float): Float = nativeMath.cos(x.toDouble()).toFloat()\n\n/**\n * Computes the tangent of the angle [x] given in
radians.\n *\n * Special cases:\n * - `tan(NaN|+Inf|-Inf)` is `NaN`\n
*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun tan(x: Float): Float =
nativeMath.tan(x.toDouble()).toFloat()\n\n/**\n * Computes the arc sine of the value [x];\n *\n * the returned value is
an angle in the range from `-PI/2` to `PI/2` radians.\n *\n * Special cases:\n * - `asin(x)` is `NaN`, when `abs(x) >
1` or x is `NaN`\n *\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun asin(x: Float): Float =
nativeMath.asin(x.toDouble()).toFloat()\n\n/**\n * Computes the arc cosine of the value [x];\n *\n * the returned value
is an angle in the range from `0.0` to `PI` radians.\n *\n * Special cases:\n * - `acos(x)` is `NaN`, when `abs(x) >
1` or x is `NaN`\n *\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun acos(x: Float): Float =
nativeMath.acos(x.toDouble()).toFloat()\n\n/**\n * Computes the arc tangent of the value [x];\n *\n * the returned value
is an angle in the range from `-PI/2` to `PI/2` radians.\n *\n * Special cases:\n * - `atan(NaN)` is `NaN`\n
*\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun atan(x: Float): Float =
nativeMath.atan(x.toDouble()).toFloat()\n\n/**\n * Returns the angle `theta` of the polar coordinates `(r, theta)` that
correspond\n *\n * to the rectangular coordinates `(x, y)` by computing the arc tangent of the value [y] / [x];\n *\n * the
returned value is an angle in the range from `-PI` to `PI` radians.\n *\n * Special cases:\n * - `atan2(0.0, 0.0)` is
`0.0`\n * - `atan2(0.0, x)` is `0.0` for `x > 0` and `PI` for `x < 0`\n * - `atan2(-0.0, x)` is `-0.0` for `x > 0` and
`-PI`

```

for $x < 0$ `atan2(y, +Inf)` is `0.0` for $0 < y < +Inf$ and `-0.0` for $-Inf < y < 0$ `atan2(y, -Inf)` is `PI` for $0 < y < +Inf$ and `-PI` for $-Inf < y < 0$ `atan2(y, 0.0)` is `PI/2` for $y > 0$ and `-PI/2` for $y < 0$ `atan2(+Inf, x)` is `PI/2` for finite x `atan2(-Inf, x)` is `-PI/2` for finite x `atan2(NaN, x)` and `atan2(y, NaN)` is `NaN`

```

@SinceKotlin("1.2")
@InlineOnly
public actual inline fun atan2(y: Float, x: Float): Float = nativeMath.atan2(y.toDouble(), x.toDouble()).toFloat()

```

`atan2` Computes the hyperbolic sine of the value `[x]`.

Special cases:

- `sinh(NaN)` is `NaN`
- `sinh(+Inf)` is `+Inf`
- `sinh(-Inf)` is `-Inf`

```

@SinceKotlin("1.2")
@InlineOnly
public actual inline fun sinh(x: Float): Float = nativeMath.sinh(x.toDouble()).toFloat()

```

`sinh` Computes the hyperbolic cosine of the value `[x]`.

Special cases:

- `cosh(NaN)` is `NaN`
- `cosh(+Inf|-Inf)` is `+Inf`

```

@SinceKotlin("1.2")
@InlineOnly
public actual inline fun cosh(x: Float): Float = nativeMath.cosh(x.toDouble()).toFloat()

```

`cosh` Computes the hyperbolic tangent of the value `[x]`.

Special cases:

- `tanh(NaN)` is `NaN`
- `tanh(+Inf)` is `1.0`
- `tanh(-Inf)` is `-1.0`

```

@SinceKotlin("1.2")
@InlineOnly
public actual inline fun tanh(x: Float): Float = nativeMath.tanh(x.toDouble()).toFloat()

```

`tanh` Computes the inverse hyperbolic sine of the value `[x]`.

The returned value is `y` such that `sinh(y) == x`.

Special cases:

- `asinh(NaN)` is `NaN`
- `asinh(+Inf)` is `+Inf`
- `asinh(-Inf)` is `-Inf`

```

@SinceKotlin("1.2")
@InlineOnly
public actual inline fun asinh(x: Float): Float = nativeMath.asinh(x.toDouble()).toFloat()

```

`asinh` Computes the inverse hyperbolic cosine of the value `[x]`.

The returned value is positive `y` such that `cosh(y) == x`.

Special cases:

- `acosh(NaN)` is `NaN`
- `acosh(x)` is `NaN` when $x < 1$
- `acosh(+Inf)` is `+Inf`

```

@SinceKotlin("1.2")
@InlineOnly
public actual inline fun acosh(x: Float): Float = nativeMath.acosh(x.toDouble()).toFloat()

```

`acosh` Computes the inverse hyperbolic tangent of the value `[x]`.

The returned value is `y` such that `tanh(y) == x`.

Special cases:

- `atanh(NaN)` is `NaN`
- `atanh(x)` is `NaN` when $x > 1$ or $x < -1$
- `atanh(1.0)` is `+Inf`
- `atanh(-1.0)` is `-Inf`

```

@SinceKotlin("1.2")
@InlineOnly
public actual inline fun atanh(x: Float): Float = nativeMath.atanh(x.toDouble()).toFloat()

```

`atanh` Computes `sqrt(x^2 + y^2)` without intermediate overflow or underflow.

Special cases:

- returns `+Inf` if any of arguments is infinite
- returns `NaN` if any of arguments is `NaN` and the other is not infinite

```

@SinceKotlin("1.2")
@InlineOnly
public actual inline fun hypot(x: Float, y: Float): Float = nativeMath.hypot(x.toDouble(), y.toDouble()).toFloat()

```

`hypot` Computes the positive square root of the value `[x]`.

Special cases:

- `sqrt(x)` is `NaN` when $x < 0$ or x is `NaN`

```

@SinceKotlin("1.2")
@InlineOnly
public actual inline fun sqrt(x: Float): Float = nativeMath.sqrt(x.toDouble()).toFloat()

```

`sqrt` Computes Euler's number `e` raised to the power of the value `[x]`.

Special cases:

- `exp(NaN)` is `NaN`
- `exp(+Inf)` is `+Inf`
- `exp(-Inf)` is `0.0`

```

@SinceKotlin("1.2")
@InlineOnly
public actual inline fun exp(x: Float): Float = nativeMath.exp(x.toDouble()).toFloat()

```

`exp` Computes `exp(x) - 1`.

This function can be implemented to produce more precise result for `[x]` near zero.

Special cases:

- `expm1(NaN)` is `NaN`
- `expm1(+Inf)` is `+Inf`
- `expm1(-Inf)` is `-1.0`

@see `[exp]` function.

```

@SinceKotlin("1.2")
@InlineOnly
public actual inline fun expm1(x: Float): Float = nativeMath.expm1(x.toDouble()).toFloat()

```

`expm1` Computes the logarithm of the value `[x]` to the given `[base]`.

Special cases:

- `log(x, b)` is `NaN` if either `x` or `b` are `NaN`
- `log(x, b)` is `NaN` when $x < 0$ or $b <= 0$ or $b == 1.0$
- `log(+Inf, +Inf)` is `NaN`
- `log(+Inf, b)` is `+Inf` for $b > 1$ and `-Inf` for $b < 1$
- `log(0.0, b)` is `-Inf` for $b > 1$ and `+Inf` for $b > 1$

See also logarithm functions for common fixed bases: `[ln]`, `[log10]` and `[log2]`.

```

@SinceKotlin("1.2")
@InlineOnly
public actual inline fun log(x: Float, base: Float): Float = log(x.toDouble(), base.toDouble()).toFloat()

```

`log` Computes the natural logarithm (base `E`) of the value `[x]`.

Special cases:

- `ln(NaN)` is `NaN`
- `ln(x)` is `NaN` when $x < 0$
- `ln(+Inf)` is `+Inf`
- `ln(0.0)` is `-Inf`

```

@SinceKotlin("1.2")
@InlineOnly
public actual inline fun ln(x: Float): Float = nativeMath.log(x.toDouble()).toFloat()

```

`ln` Computes the common logarithm (base 10) of the value `[x]`.

@see `[ln]` function for special cases.

```

@SinceKotlin("1.2")
@InlineOnly
public actual inline fun log10(x: Float): Float = nativeMath.log10(x.toDouble()).toFloat()

```

`log10` Computes the binary

logarithm (base 2) of the value [x].
`@see [ln] function for special cases.`

```

*^@SinceKotlin("1.2")@InlineOnly\npublic actual inline fun log2(x: Float): Float =
nativeMath.log2(x.toDouble()).toFloat()

```

Computes $\ln(a + 1)$.
This function can be implemented to produce more precise result for [x] near zero.
Special cases:
 $\ln(\text{NaN})$ is NaN
 $\ln(x)$ is NaN where $x < -1.0$
 $\ln(-1.0)$ is $-\text{Inf}$
 $\ln(+\text{Inf})$ is $+\text{Inf}$
@see [ln] function
@see [expm1] function

```

*^@SinceKotlin("1.2")@InlineOnly\npublic actual inline fun ln1p(x: Float): Float =
nativeMath.log1p(x.toDouble()).toFloat()

```

Rounds the given value [x] to an integer towards positive infinity.
@return the smallest Float value that is greater than or equal to the given value [x] and is a mathematical integer.
Special cases:
 $\text{ceil}(x)$ is x where x is NaN or $+\text{Inf}$ or $-\text{Inf}$ or already a mathematical integer.

```

*^@SinceKotlin("1.2")@InlineOnly\npublic actual inline fun ceil(x: Float): Float =
nativeMath.ceil(x.toDouble()).toFloat()

```

Rounds the given value [x] to an integer towards negative infinity.
@return the largest Float value that is smaller than or equal to the given value [x] and is a mathematical integer.
Special cases:
 $\text{floor}(x)$ is x where x is NaN or $+\text{Inf}$ or $-\text{Inf}$ or already a mathematical integer.

```

*^@SinceKotlin("1.2")@InlineOnly\npublic actual inline fun floor(x: Float): Float =
nativeMath.floor(x.toDouble()).toFloat()

```

Rounds the given value [x] to an integer towards zero.
@return the value [x] having its fractional part truncated.
Special cases:
 $\text{truncate}(x)$ is x where x is NaN or $+\text{Inf}$ or $-\text{Inf}$ or already a mathematical integer.

```

*^@SinceKotlin("1.2")@InlineOnly\npublic actual inline fun truncate(x: Float): Float =
truncate(x.toDouble()).toFloat()

```

Rounds the given value [x] towards the closest integer with ties rounded towards even integer.
Special cases:
 $\text{round}(x)$ is x where x is NaN or $+\text{Inf}$ or $-\text{Inf}$ or already a mathematical integer.

```

*^@SinceKotlin("1.2")@InlineOnly\npublic actual inline fun round(x: Float): Float =
round(x.toDouble()).toFloat()

```

Returns the absolute value of the given value [x].
Special cases:
 $\text{abs}(\text{NaN})$ is NaN
@see absoluteValue extension property for [Float]

```

*^@SinceKotlin("1.2")@InlineOnly\npublic actual inline fun abs(x: Float): Float =
nativeMath.abs(x.toDouble()).toFloat()

```

Returns the sign of the given value [x]:
 -1.0 if the value is negative,
zero if the value is zero,
 1.0 if the value is positive
Special case:
 $\text{sign}(\text{NaN})$ is NaN

```

*^@SinceKotlin("1.2")@InlineOnly\npublic actual inline fun sign(x: Float): Float =
nativeMath.sign(x.toDouble()).toFloat()

```

Returns the smaller of two values.
If either value is NaN , then the result is NaN .

```

*^@SinceKotlin("1.2")@InlineOnly\npublic actual inline fun min(a: Float, b: Float): Float =
nativeMath.min(a, b)

```

Returns the greater of two values.
If either value is NaN , then the result is NaN .

```

*^@SinceKotlin("1.2")@InlineOnly\npublic actual inline fun max(a: Float, b: Float): Float =
nativeMath.max(a, b)

```

extensions

```

*^@SinceKotlin("1.2")@InlineOnly\npublic actual inline fun Float.pow(x: Float): Float =
nativeMath.pow(this.toDouble(), x.toDouble()).toFloat()

```

Raises this value to the power [x].
Special cases:
 $b.\text{pow}(0.0)$ is 1.0
 $b.\text{pow}(1.0) == b$
 $b.\text{pow}(\text{NaN})$ is NaN
 $\text{NaN}.\text{pow}(x)$ is NaN for $x \neq 0.0$
 $b.\text{pow}(\text{Inf})$ is NaN for $\text{abs}(b) == 1.0$
 $b.\text{pow}(x)$ is NaN for $b < 0$ and x is finite and not an integer

```

*^@SinceKotlin("1.2")@InlineOnly\npublic actual inline fun Float.pow(n: Int): Float =
nativeMath.pow(this.toDouble(), n.toDouble()).toFloat()

```

Raises this value to the integer power [n].
See the other overload of [pow] for details.

```

*^@SinceKotlin("1.2")@InlineOnly\npublic actual inline fun Float.absoluteValue: Float get() =
nativeMath.abs(this.toDouble()).toFloat()

```

Returns the sign of this value:
 -1.0 if the value is negative,
zero if the value is zero,
 1.0 if the value is positive
Special case:
 $\text{NaN}.\text{sign}$ is NaN

```

*^@SinceKotlin("1.2")@InlineOnly\npublic actual inline fun Float.sign: Float get() =
nativeMath.sign(this.toDouble()).toFloat()

```

Returns this value with the sign bit same as of the [sign] value.
If [sign] is NaN the sign of the result is undefined.

```

*^@SinceKotlin("1.2")@InlineOnly\npublic actual inline fun Float.withSign(sign: Float): Float =
this.toDouble().withSign(sign.toDouble()).toFloat()

```

Returns this value with the sign bit same as of the [sign] value.

```

*^@SinceKotlin("1.2")@InlineOnly\npublic actual inline fun Float.withSign(sign: Int): Float =

```

```

this.toDouble().withSign(sign.toDouble()).toFloat()\n\n/**\n * Rounds this [Float] value to the nearest integer and
converts the result to [Int].\n * Ties are rounded towards positive infinity.\n * Special cases:\n * -
`x.roundToInt() == Int.MAX_VALUE` when `x > Int.MAX_VALUE`\n * - `x.roundToInt() == Int.MIN_VALUE`
when `x < Int.MIN_VALUE`\n * @throws IllegalArgumentException when this value is `NaN`\n
*/\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun Float.roundToInt(): Int =
toDouble().roundToInt()\n\n/**\n * Rounds this [Float] value to the nearest integer and converts the result to
[Long].\n * Ties are rounded towards positive infinity.\n * Special cases:\n * - `x.roundToLong() ==
Long.MAX_VALUE` when `x > Long.MAX_VALUE`\n * - `x.roundToLong() == Long.MIN_VALUE` when `x
< Long.MIN_VALUE`\n * @throws IllegalArgumentException when this value is `NaN`\n
*/\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun Float.roundToLong(): Long =
toDouble().roundToLong()\n\n// endregion\n\n// region ===== Integer Math
=====
\n\n/**\n * Returns the absolute value of the given value
[n].\n * Special cases:\n * - `abs(Int.MIN_VALUE)` is `Int.MIN_VALUE` due to an overflow\n * @see
absoluteValue extension property for [Int]\n * // TODO: remove manual 'or' when KT-19290 is
fixed\n@SinceKotlin("1.2")\npublic actual fun abs(n: Int): Int = if (n < 0) (-n or 0) else n\n\n/**\n * Returns the
smaller of two values.\n * // @SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun min(a: Int, b: Int): Int =
nativeMath.min(a, b)\n\n/**\n * Returns the greater of two values.\n
*/\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline fun max(a: Int, b: Int): Int = nativeMath.max(a,
b)\n\n/**\n * Returns the absolute value of this value.\n * Special cases:\n * -
`Int.MIN_VALUE.absoluteValue` is `Int.MIN_VALUE` due to an overflow\n * @see abs function\n
*/\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline val Int.absoluteValue: Int get() = abs(this)\n\n/**\n *
Returns the sign of this value:\n * - -1` if the value is negative,\n * - `0` if the value is zero,\n * - `1` if the value
is positive\n * // @SinceKotlin("1.2")\npublic actual val Int.sign: Int get() = when {\n    this < 0 -> -1\n    this > 0 -
> 1\n    else -> 0\n}\n\n\n/**\n * Returns the absolute value of the given value [n].\n * Special cases:\n * -
`abs(Long.MIN_VALUE)` is `Long.MIN_VALUE` due to an overflow\n * @see absoluteValue extension
property for [Long]\n * // @SinceKotlin("1.2")\npublic actual fun abs(n: Long): Long = if (n < 0) -n else
n\n\n/**\n * Returns the smaller of two values.\n
*/\n@SinceKotlin("1.2")\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline fun min(a: Long, b:
Long): Long = if (a <= b) a else b\n\n/**\n * Returns the greater of two values.\n
*/\n@SinceKotlin("1.2")\n@Suppress("NOTHING_TO_INLINE")\npublic actual inline fun max(a: Long, b:
Long): Long = if (a >= b) a else b\n\n/**\n * Returns the absolute value of this value.\n * Special cases:\n * -
`Long.MIN_VALUE.absoluteValue` is `Long.MIN_VALUE` due to an overflow\n * @see abs function\n
*/\n@SinceKotlin("1.2")\n@InlineOnly\npublic actual inline val Long.absoluteValue: Long get() =
abs(this)\n\n/**\n * Returns the sign of this value:\n * - -1` if the value is negative,\n * - `0` if the value is zero,\n
* - `1` if the value is positive\n * // @SinceKotlin("1.2")\npublic actual val Long.sign: Int get() = when {\n    this
< 0 -> -1\n    this > 0 -> 1\n    else -> 0\n}\n\n\n// endregion\n", "/*\n * Copyright 2010-2021 JetBrains s.r.o. and
Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin\n\n/**\n * Returns `true` if the specified
number is a\n * Not-a-Number (NaN) value, `false` otherwise.\n * // @public actual fun Double.isNaN(): Boolean =
this != this\n\n/**\n * Returns `true` if the specified number is a\n * Not-a-Number (NaN) value, `false` otherwise.\n
*/\n@public actual fun Float.isNaN(): Boolean = this != this\n\n/**\n * Returns `true` if this value is infinitely large in
magnitude.\n * // @public actual fun Double.isInfinite(): Boolean = this == Double.POSITIVE_INFINITY || this ==
Double.NEGATIVE_INFINITY\n\n/**\n * Returns `true` if this value is infinitely large in magnitude.\n * // @public
actual fun Float.isInfinite(): Boolean = this == Float.POSITIVE_INFINITY || this ==
Float.NEGATIVE_INFINITY\n\n/**\n * Returns `true` if the argument is a finite floating-point value; returns
`false` otherwise (for `NaN` and infinity arguments).\n * // @public actual fun Double.isFinite(): Boolean =
!isInfinite() && !isNaN()\n\n/**\n * Returns `true` if the argument is a finite floating-point value; returns `false`
otherwise (for `NaN` and infinity arguments).\n * // @public actual fun Float.isFinite(): Boolean = !isInfinite() &&

```

```

!isNaN()\n\n\n/**\n * Counts the number of set bits in the binary representation of this [Int] number.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun
Int.countOneBits(): Int {\n // Hacker's Delight 5-1 algorithm\n var v = this\n v = (v and 0x55555555) +
(v.ushr(1) and 0x55555555)\n v = (v and 0x33333333) + (v.ushr(2) and 0x33333333)\n v = (v and 0x0F0F0F0F)
+ (v.ushr(4) and 0x0F0F0F0F)\n v = (v and 0x00FF00FF) + (v.ushr(8) and 0x00FF00FF)\n v = (v and
0x0000FFFF) + (v.ushr(16))\n return v}\n\n\n/**\n * Counts the number of consecutive most significant bits that
are zero in the binary representation of this [Int] number.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npubli
c actual inline fun Int.countLeadingZeroBits(): Int = JsMath.clz32(this)\n\n\n/**\n * Counts the number of
consecutive least significant bits that are zero in the binary representation of this [Int] number.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun
Int.countTrailingZeroBits(): Int =\n // Hacker's Delight 5-4 algorithm for expressing countTrailingZeroBits with
countLeadingZeroBits\n Int.SIZE_BITS - (this or -this).inv().countLeadingZeroBits()\n\n\n/**\n * Returns a
number having a single bit set in the position of the most significant set bit of this [Int] number,\n * or zero, if this
number is zero.\n *\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun
Int.takeHighestOneBit(): Int =\n if (this == 0) 0 else 1.shl(Int.SIZE_BITS - 1 - countLeadingZeroBits())\n\n\n/**\n *
Returns a number having a single bit set in the position of the least significant set bit of this [Int] number,\n * or
zero, if this number is zero.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun
Int.takeLowestOneBit(): Int =\n // Hacker's Delight 2-1 algorithm for isolating rightmost 1-bit\n this and -
this\n\n\n/**\n * Rotates the binary representation of this [Int] number left by the specified [bitCount] number of
bits.\n * The most significant bits pushed out from the left side reenter the number as the least significant bits on the
right side.\n *\n * Rotating the number left by a negative bit count is the same as rotating it right by the negated bit
count:\n * `number.rotateLeft(-n) == number.rotateRight(n)`\n *\n * Rotating by a multiple of [Int.SIZE_BITS] (32)
returns the same number, or more generally\n * `number.rotateLeft(n) == number.rotateLeft(n % 32)`\n
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun
Int.rotateLeft(bitCount: Int): Int =\n shl(bitCount) or ushr(Int.SIZE_BITS - bitCount)\n\n\n/**\n * Rotates the
binary representation of this [Int] number right by the specified [bitCount] number of bits.\n * The least significant
bits pushed out from the right side reenter the number as the most significant bits on the left side.\n *\n * Rotating
the number right by a negative bit count is the same as rotating it left by the negated bit count:\n *
`number.rotateRight(-n) == number.rotateLeft(n)`\n *\n * Rotating by a multiple of [Int.SIZE_BITS] (32) returns
the same number, or more generally\n * `number.rotateRight(n) == number.rotateRight(n % 32)`\n
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun
Int.rotateRight(bitCount: Int): Int =\n shl(Int.SIZE_BITS - bitCount) or ushr(bitCount)\n\n\n\n/**\n * Counts the
number of set bits in the binary representation of this [Long] number.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun
Long.countOneBits(): Int =\n high.countOneBits() + low.countOneBits()\n\n\n/**\n * Counts the number of
consecutive most significant bits that are zero in the binary representation of this [Long] number.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun
Long.countLeadingZeroBits(): Int =\n when (val high = this.high) {\n 0 -> Int.SIZE_BITS +
low.countLeadingZeroBits()\n else -> high.countLeadingZeroBits()\n }\n\n\n/**\n * Counts the number of
consecutive least significant bits that are zero in the binary representation of this [Long] number.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun
Long.countTrailingZeroBits(): Int =\n when (val low = this.low) {\n 0 -> Int.SIZE_BITS +
high.countTrailingZeroBits()\n else -> low.countTrailingZeroBits()\n }\n\n\n/**\n * Returns a number having a
single bit set in the position of the most significant set bit of this [Long] number,\n * or zero, if this number is
zero.\n *\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun
Long.takeHighestOneBit(): Long =\n when (val high = this.high) {\n 0 -> Long(low.takeHighestOneBit(),

```

```

0)\n    else -> Long(0, high.takeHighestOneBit())\n } \n\n/**\n * Returns a number having a single bit set in the
position of the least significant set bit of this [Long] number,\n * or zero, if this number is zero.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun
Long.takeLowestOneBit(): Long =\n    when (val low = this.low) {\n        0 -> Long(0, high.takeLowestOneBit())\n        else -> Long(low.takeLowestOneBit(), 0)\n    } \n\n/**\n * Rotates the binary representation of this [Long]
number left by the specified [bitCount] number of bits.\n * The most significant bits pushed out from the left side
reenter the number as the least significant bits on the right side.\n *\n * Rotating the number left by a negative bit
count is the same as rotating it right by the negated bit count:\n * `number.rotateLeft(-n) ==
number.rotateRight(n)`\n *\n * Rotating by a multiple of [Long.SIZE_BITS] (64) returns the same number, or more
generally\n * `number.rotateLeft(n) == number.rotateLeft(n % 64)`\n
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun
Long.rotateLeft(bitCount: Int): Long {\n    if ((bitCount and 31) != 0) {\n        val low = this.low\n        val high =
this.high\n        val newLow = low.shl(bitCount) or high.ushr(-bitCount)\n        val newHigh = high.shl(bitCount) or
low.ushr(-bitCount)\n        return if ((bitCount and 32) == 0) Long(newLow, newHigh) else Long(newHigh,
newLow)\n    } else {\n        return if ((bitCount and 32) == 0) this else Long(high, low)\n    } \n\n\n\n/**\n *
Rotates the binary representation of this [Long] number right by the specified [bitCount] number of bits.\n * The
least significant bits pushed out from the right side reenter the number as the most significant bits on the left side.\n
*\n * Rotating the number right by a negative bit count is the same as rotating it left by the negated bit count:\n *
`number.rotateRight(-n) == number.rotateLeft(n)`\n *\n * Rotating by a multiple of [Long.SIZE_BITS] (64) returns
the same number, or more generally\n * `number.rotateRight(n) == number.rotateRight(n % 64)`\n
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
c actual inline fun Long.rotateRight(bitCount: Int): Long = rotateLeft(-bitCount)\n", "/*\n * Copyright 2010-2018
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.js\n\nimport
kotlin.internal.LowPriorityInOverloadResolution\n\n/**\n * Exposes the JavaScript [Promise
object](https://developer.mozilla.org/en/docs/Web/JavaScript/Reference/Global_Objects/Promise) to Kotlin.\n
*\n@Suppress("NOT_DOCUMENTED")\npublic open external class Promise<out T>(executor: (resolve: (T) ->
Unit, reject: (Throwable) -> Unit) -> Unit) {\n    @LowPriorityInOverloadResolution\n    public open fun <S>
then(onFulfilled: ((T) -> S)?): Promise<S>\n\n    @LowPriorityInOverloadResolution\n    public open fun <S>
then(onFulfilled: ((T) -> S)?, onRejected: ((Throwable) -> S)?): Promise<S>\n\n    public open fun <S>
catch(onRejected: (Throwable) -> S): Promise<S>\n\n    public open fun finally(onFinally: () -> Unit):
Promise<T>\n\n    companion object {\n        public fun <S> all(promise: Array<out Promise<S>>):
Promise<Array<out S>>\n\n        public fun <S> race(promise: Array<out Promise<S>>): Promise<S>\n\n        public
fun reject(e: Throwable): Promise<Nothing>\n\n        public fun <S> resolve(e: S): Promise<S>\n        public
fun <S> resolve(e: Promise<S>): Promise<S>\n    } \n\n\n\n// It's workaround for KT-19672 since we can fix it
properly until KT-11265 isn't fixed.\n\ninline fun <T, S> Promise<Promise<T>>.then(\n    noinline onFulfilled: ((T) -
> S)?\n): Promise<S> {\n    return this.unsafeCast<Promise<T>>().then(onFulfilled)\n} \n\n\ninline fun <T, S>
Promise<Promise<T>>.then(\n    noinline onFulfilled: ((T) -> S)?,\n    noinline onRejected: ((Throwable) -> S)?\n):
Promise<S> {\n    return this.unsafeCast<Promise<T>>().then(onFulfilled, onRejected)\n} \n\n", "/*\n * Copyright
2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed
by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage
kotlin.random\n\nimport kotlin.math.pow\n\ninternal actual fun defaultPlatformRandom(): Random =\n    Random(js("Math.random() * Math.pow(2, 32)) | 0").unsafeCast<Int>())\n\n\nprivate val INV_2_26: Double =
2.0.pow(-26)\n\nprivate val INV_2_53: Double = 2.0.pow(-53)\n\ninternal actual fun doubleFromParts(hi26: Int, low27:
Int): Double =\n    hi26 * INV_2_26 + low27 * INV_2_53", "/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n */\n\npackage kotlin.reflect\n\nimport findAssociatedObject\n\n/**\n * The
experimental marker for associated objects API.\n *\n * Any usage of a declaration annotated with

```

```

`@ExperimentalAssociatedObjects` must be accepted either by\n * annotating that usage with the [OptIn]
annotation, e.g. `@OptIn(ExperimentalAssociatedObjects::class)`,\n * or by using the compiler argument `-opt-
in=kotlin.reflect.ExperimentalAssociatedObjects`.\n * ^\n @RequiresOptIn(level =
RequiresOptIn.Level.ERROR)\n @Retention(value = AnnotationRetention.BINARY)\n public annotation class
ExperimentalAssociatedObjects\n\n /**\n * Makes the annotated annotation class an associated object key.\n * ^\n *
An associated object key annotation should have single [KClass] parameter.\n * When applied to a class with
reference to an object declaration as an argument, it binds\n * the object to the class, making this binding
discoverable at runtime using [findAssociatedObject].\n
*\n @ExperimentalAssociatedObjects\n @Retention(AnnotationRetention.BINARY)\n @Target(AnnotationTarget.A
NNOTATION_CLASS)\n public annotation class AssociatedObjectKey\n\n /**\n * If [T] is an
@[AssociatedObjectKey]-annotated annotation class and [this] class is annotated with @[T] (`S::class`),\n * returns
object `S`.\n * ^\n * Otherwise returns `null`.\n * ^\n @ExperimentalAssociatedObjects\n public inline fun <reified T :
Annotation> KClass<*>.findAssociatedObject(): Any? =\n this.findAssociatedObject(T::class)", /*\n * Copyright
2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed
by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n * ^\n\n package kotlin.js\n\n import
getKClass\n\n import kotlin.reflect.KClass\n\n import kotlin.reflect.js.internal.KClassImpl\n\n\n /**\n * Represents the
constructor of a class. Instances of `JsClass` can be passed to JavaScript APIs that expect a constructor reference.\n
*\n ^\n external interface JsClass<T : Any> {\n /*\n * Returns the unqualified name of the class represented by
this instance.\n * ^\n val name: String\n }\n\n /**\n * Obtains a constructor reference for the given `KClass`.\n
*\n ^\n val <T : Any> KClass<T>.js: JsClass<T>\n get() = (this as KClassImpl<T>).jClass\n\n /**\n * Obtains a
`KClass` instance for the given constructor reference.\n * ^\n val <T : Any> JsClass<T>.kotlin: KClass<T>\n get()
= getKClass(this)\n", /*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n * ^\n\n package kotlin.reflect.js.internal\n\n import kotlin.reflect.*\n\n internal abstract
class KClassImpl<T : Any>(\n internal open val jClass: JsClass<T>\n) : KClass<T> {\n\n override val
qualifiedName: String?\n get() = TODO()\n\n override fun equals(other: Any?): Boolean {\n return other
is KClassImpl<*> && jClass == other.jClass\n }\n\n // TODO: use FQN\n override fun hashCode(): Int =
simpleName?.hashCode() ?: 0\n\n override fun toString(): String {\n // TODO: use FQN\n return "\"class
$simpleName\"\n }\n}\n\n internal class SimpleKClassImpl<T : Any>(jClass: JsClass<T>) :
KClassImpl<T>(jClass) {\n override val simpleName: String? =
jClass.asDynamic().$metadata$.simpleName.unsafeCast<String?>()\n\n override fun isInstance(value: Any?):
Boolean {\n return jsIsType(value, jClass)\n }\n}\n\n internal class PrimitiveKClassImpl<T : Any>(\n
jClass: JsClass<T>,\n private val givenSimpleName: String,\n private val isInstanceFunction: (Any?) ->
Boolean\n) : KClassImpl<T>(jClass) {\n override fun equals(other: Any?): Boolean {\n if (other !is
PrimitiveKClassImpl<*>) return false\n return super.equals(other) && givenSimpleName ==
other.givenSimpleName\n }\n\n override val simpleName: String? get() = givenSimpleName\n\n override fun
isInstance(value: Any?): Boolean {\n return isInstanceFunction(value)\n }\n}\n\n internal object
NothingKClassImpl : KClassImpl<Nothing>(js("Object")) {\n override val simpleName: String =
\"Nothing\"\n\n override fun isInstance(value: Any?): Boolean = false\n\n override val jClass:
JsClass<Nothing>\n get() = throw UnsupportedOperationException("There's no native JS class for Nothing
type")\n\n override fun equals(other: Any?): Boolean = other === this\n\n override fun hashCode(): Int =
0\n}\n\n internal class ErrorKClass : KClass<Nothing> {\n override val simpleName: String? get() =
error("Unknown simpleName for ErrorKClass")\n\n override val qualifiedName: String? get() = error("Unknown
qualifiedName for ErrorKClass")\n\n override fun isInstance(value: Any?): Boolean = error("Can't check
isInstance on ErrorKClass")\n\n override fun equals(other: Any?): Boolean = other === this\n\n override fun
hashCode(): Int = 0\n}", /*\n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n * ^\n\n package kotlin.reflect\n\n internal actual inline val

```

```

KClass<*>.qualifiedOrSimpleName: String? \n    get() = simpleName", /* \n * Copyright 2010-2018 JetBrains s.r.o.
and Kotlin Programming Language contributors. \n * Use of this source code is governed by the Apache 2.0 license
that can be found in the license/LICENSE.txt file. \n */ \n // a package is omitted to get declarations directly under
the module \n \n // TODO: Remove once JsReflectionAPICallChecker supports more reflection
types \n @file: Suppress("Unsupported") \n \n import kotlin.reflect.* \n import
kotlin.reflect.js.internal.* \n @JsName("createKType") \n internal fun createKType(\n    classifier: KClassifier, \n
arguments: Array<KTypeProjection>, \n    isMarkedNullable: Boolean \n) = \n    KTypeImpl(classifier,
arguments.asList(), isMarkedNullable) \n @JsName("createDynamicKType") \n internal fun
createDynamicKType(): KType = DynamicKType \n @JsName("markKTypeNullable") \n internal fun
markKTypeNullable(kType: KType) = KTypeImpl(kType.classifier!!, kType.arguments,
true) \n @JsName("createKTypeParameter") \n internal fun createKTypeParameter(\n    name: String, \n
upperBounds: Array<KType>, \n    variance: String \n): KTypeParameter { \n    val kVariance = when (variance) { \n
    "in" -> KVariance.IN \n    "out" -> KVariance.OUT \n    else -> KVariance.INVARIANT \n    } \n    return
KTypeParameterImpl(name, upperBounds.asList(), kVariance,
false) \n } \n @JsName("getStarKTypeProjection") \n internal fun getStarKTypeProjection(): KTypeProjection = \n
KTypeProjection.STAR \n @JsName("createCovariantKTypeProjection") \n internal fun
createCovariantKTypeProjection(type: KType): KTypeProjection = \n
KTypeProjection.covariant(type) \n @JsName("createInvariantKTypeProjection") \n internal fun
createInvariantKTypeProjection(type: KType): KTypeProjection = \n
KTypeProjection.invariant(type) \n @JsName("createContravariantKTypeProjection") \n internal fun
createContravariantKTypeProjection(type: KType): KTypeProjection = \n
KTypeProjection.contravariant(type) \n /* \n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming
Language contributors. \n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file. \n */ \n \n package kotlin.reflect.js.internal \n \n import kotlin.reflect.* \n \n internal class
KTypeImpl(\n    override val classifier: KClassifier, \n    override val arguments: List<KTypeProjection>, \n
override val isMarkedNullable: Boolean \n) : KType { \n    override fun equals(other: Any?): Boolean = \n        other
is KTypeImpl && \n            classifier == other.classifier && arguments == other.arguments &&
isMarkedNullable == other.isMarkedNullable \n    \n    override fun hashCode(): Int = \n        (classifier.hashCode() * 31
+ arguments.hashCode()) * 31 + isMarkedNullable.hashCode() \n    \n    override fun toString(): String { \n        val
kClass = (classifier as? KClass<*>) \n            val classifierName = when { \n                kClass == null ->
classifier.toString() \n                kClass.simpleName != null -> kClass.simpleName \n                else -> "(non-denotable
type)" \n            } \n            val args = \n                if (arguments.isEmpty()) "" \n                else arguments.joinToString(", ",
"<", ">") \n            val nullable = if (isMarkedNullable) "?" else "" \n            return classifierName + args +
nullable \n        } \n    } \n    \n    internal object DynamicKType : KType { \n        override val classifier: KClassifier? = null \n
override val arguments: List<KTypeProjection> = emptyList() \n        override val isMarkedNullable: Boolean = false \n
override fun toString(): String = "dynamic" \n    } \n } \n /* \n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin
Programming Language contributors. \n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file. \n */ \n \n package kotlin.reflect.js.internal \n \n import kotlin.reflect.* \n \n internal
data class KTypeParameterImpl(\n    override val name: String, \n    override val upperBounds: List<KType>, \n
override val variance: KVariance, \n    override val isReified: Boolean \n) : KTypeParameter { \n    override fun
toString(): String = name \n } \n /* \n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
contributors. \n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file. \n */ \n \n package kotlin.reflect.js.internal \n \n import
kotlin.js.JsClass \n @JsName("PrimitiveClasses") \n internal object PrimitiveClasses { \n
@JsName("anyClass") \n    val anyClass = PrimitiveKClassImpl(js("Object").unsafeCast<JsClass<Any>>(),
"Any", { it is Any }) \n    @JsName("numberClass") \n    val numberClass =
PrimitiveKClassImpl(js("Number").unsafeCast<JsClass<Number>>(), "Number", { it is Number }) \n
@JsName("nothingClass") \n    val nothingClass = NothingKClassImpl \n    @JsName("booleanClass") \n    val

```

```

booleanClass = PrimitiveKClassImpl(js("Boolean").unsafeCast<JsClass<Boolean>>(), "Boolean", { it is Boolean
})\n\n @JsName("byteClass")\n val byteClass =
PrimitiveKClassImpl(js("Number").unsafeCast<JsClass<Byte>>(), "Byte", { it is Byte })\n\n
@JsName("shortClass")\n val shortClass = PrimitiveKClassImpl(js("Number").unsafeCast<JsClass<Short>>(),
"Short", { it is Short })\n\n @JsName("intClass")\n val intClass =
PrimitiveKClassImpl(js("Number").unsafeCast<JsClass<Int>>(), "Int", { it is Int })\n\n
@JsName("floatClass")\n val floatClass = PrimitiveKClassImpl(js("Number").unsafeCast<JsClass<Float>>(),
"Float", { it is Float })\n\n @JsName("doubleClass")\n val doubleClass =
PrimitiveKClassImpl(js("Number").unsafeCast<JsClass<Double>>(), "Double", { it is Double })\n\n
@JsName("arrayClass")\n val arrayClass =
PrimitiveKClassImpl(js("Array").unsafeCast<JsClass<Array<*>>>(), "Array", { it is Array<*> })\n\n
@JsName("stringClass")\n val stringClass = PrimitiveKClassImpl(js("String").unsafeCast<JsClass<String>>(),
"String", { it is String })\n\n @JsName("throwableClass")\n val throwableClass =
PrimitiveKClassImpl(js("Error").unsafeCast<JsClass<Throwable>>(), "Throwable", { it is Throwable })\n\n
@JsName("booleanArrayClass")\n val booleanArrayClass =
PrimitiveKClassImpl(js("Array").unsafeCast<JsClass<BooleanArray>>(), "BooleanArray", { it is BooleanArray
})\n\n @JsName("charArrayClass")\n val charArrayClass =
PrimitiveKClassImpl(js("Uint16Array").unsafeCast<JsClass<CharArray>>(), "CharArray", { it is CharArray
})\n\n @JsName("byteArrayClass")\n val byteArrayClass =
PrimitiveKClassImpl(js("Int8Array").unsafeCast<JsClass<ByteArray>>(), "ByteArray", { it is ByteArray })\n\n
@JsName("shortArrayClass")\n val shortArrayClass =
PrimitiveKClassImpl(js("Int16Array").unsafeCast<JsClass<ShortArray>>(), "ShortArray", { it is ShortArray
})\n\n @JsName("intArrayClass")\n val intArrayClass =
PrimitiveKClassImpl(js("Int32Array").unsafeCast<JsClass<IntArray>>(), "IntArray", { it is IntArray })\n\n
@JsName("longArrayClass")\n val longArrayClass =
PrimitiveKClassImpl(js("Array").unsafeCast<JsClass<LongArray>>(), "LongArray", { it is LongArray })\n\n
@JsName("floatArrayClass")\n val floatArrayClass =
PrimitiveKClassImpl(js("Float32Array").unsafeCast<JsClass<FloatArray>>(), "FloatArray", { it is FloatArray
})\n\n @JsName("doubleArrayClass")\n val doubleArrayClass =
PrimitiveKClassImpl(js("Float64Array").unsafeCast<JsClass<DoubleArray>>(), "DoubleArray", { it is
DoubleArray })\n\n @JsName("functionClass")\n fun functionClass(arity: Int): KClassImpl<Any> {\n
return functionClasses.get(arity) ?: run {\n val result =
PrimitiveKClassImpl(js("Function").unsafeCast<JsClass<Any>>(), "Function$arity",\n
{ jsTypeOf(it) === "function" && it.asDynamic().length === arity })\n functionClasses.asDynamic()[arity]
= result\n result\n }\n }\n\nprivate val functionClasses =
arrayOfNulls<KClassImpl<Any>>(0), "/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\n// a package is omitted to get declarations directly under the module\n\nimport
kotlin.reflect.*\nimport kotlin.reflect.js.internal.*\n\n@JsName("getKClass")\n\ninternal fun <T : Any>
getKClass(jClass: Any /* JsClass<T> | Array<JsClass<T>> */): KClass<T> {\n return if
(js("Array").isArray(jClass)) {\n getKClassM(jClass.unsafeCast<Array<JsClass<T>>>())\n } else {\n
getKClass1(jClass.unsafeCast<JsClass<T>>())\n }\n }\n\n@JsName("getKClassM")\n\ninternal fun <T : Any>
getKClassM(jClasses: Array<JsClass<T>>): KClass<T> = when (jClasses.size) {\n 1 ->
getKClass1(jClasses[0])\n 0 -> NothingKClassImpl.unsafeCast<KClass<T>>()\n else ->
ErrorKClass().unsafeCast<KClass<T>>()\n }\n\n@JsName("getKClassFromExpression")\n\ninternal fun <T : Any>
getKClassFromExpression(e: T): KClass<T> =\n when (jsTypeOf(e)) {\n "string" ->
PrimitiveClasses.stringClass\n "number" -> if (jsBitwiseOr(e, 0).asDynamic() === e)
PrimitiveClasses.intClass else PrimitiveClasses.doubleClass\n "boolean" -> PrimitiveClasses.booleanClass\n

```



```

Unicode specification.\n *^n TITLECASE_LETTER(3, \"Lt\"),\n\n /**\n * General category \"Lm\" in the
Unicode specification.\n *^n MODIFIER_LETTER(4, \"Lm\"),\n\n /**\n * General category \"Lo\" in the
Unicode specification.\n *^n OTHER_LETTER(5, \"Lo\"),\n\n /**\n * General category \"Mn\" in the
Unicode specification.\n *^n NON_SPACING_MARK(6, \"Mn\"),\n\n /**\n * General category \"Me\" in
the Unicode specification.\n *^n ENCLOSING_MARK(7, \"Me\"),\n\n /**\n * General category \"Mc\" in
the Unicode specification.\n *^n COMBINING_SPACING_MARK(8, \"Mc\"),\n\n /**\n * General
category \"Nd\" in the Unicode specification.\n *^n DECIMAL_DIGIT_NUMBER(9, \"Nd\"),\n\n /**\n *
General category \"Nl\" in the Unicode specification.\n *^n LETTER_NUMBER(10, \"Nl\"),\n\n /**\n *
General category \"No\" in the Unicode specification.\n *^n OTHER_NUMBER(11, \"No\"),\n\n /**\n *
General category \"Zs\" in the Unicode specification.\n *^n SPACE_SEPARATOR(12, \"Zs\"),\n\n /**\n *
General category \"Zl\" in the Unicode specification.\n *^n LINE_SEPARATOR(13, \"Zl\"),\n\n /**\n *
General category \"Zp\" in the Unicode specification.\n *^n PARAGRAPH_SEPARATOR(14, \"Zp\"),\n\n
/**\n * General category \"Cc\" in the Unicode specification.\n *^n CONTROL(15, \"Cc\"),\n\n /**\n *
General category \"Cf\" in the Unicode specification.\n *^n FORMAT(16, \"Cf\"),\n\n /**\n * General
category \"Co\" in the Unicode specification.\n *^n PRIVATE_USE(18, \"Co\"),\n\n /**\n * General
category \"Cs\" in the Unicode specification.\n *^n SURROGATE(19, \"Cs\"),\n\n /**\n * General category
\"Pd\" in the Unicode specification.\n *^n DASH_PUNCTUATION(20, \"Pd\"),\n\n /**\n * General
category \"Ps\" in the Unicode specification.\n *^n START_PUNCTUATION(21, \"Ps\"),\n\n /**\n *
General category \"Pe\" in the Unicode specification.\n *^n END_PUNCTUATION(22, \"Pe\"),\n\n /**\n *
General category \"Pc\" in the Unicode specification.\n *^n CONNECTOR_PUNCTUATION(23, \"Pc\"),\n\n
/**\n * General category \"Po\" in the Unicode specification.\n *^n OTHER_PUNCTUATION(24,
\"Po\"),\n\n /**\n * General category \"Sm\" in the Unicode specification.\n *^n MATH_SYMBOL(25,
\"Sm\"),\n\n /**\n * General category \"Sc\" in the Unicode specification.\n *^n
CURRENCY_SYMBOL(26, \"Sc\"),\n\n /**\n * General category \"Sk\" in the Unicode specification.\n *^n
MODIFIER_SYMBOL(27, \"Sk\"),\n\n /**\n * General category \"So\" in the Unicode specification.\n *^n
OTHER_SYMBOL(28, \"So\"),\n\n /**\n * General category \"Pi\" in the Unicode specification.\n *^n
INITIAL_QUOTE_PUNCTUATION(29, \"Pi\"),\n\n /**\n * General category \"Pf\" in the Unicode
specification.\n *^n FINAL_QUOTE_PUNCTUATION(30, \"Pf\");\n\n /**\n * Returns `true` if [char]
character belongs to this category.\n *^n public actual operator fun contains(char: Char): Boolean =
char.getCategoryValue() == this.value\n\n companion object {\n    internal fun valueOf(category: Int):
CharCategory =\n        when (category) {\n            in 0..16 -> values()[category]\n            in 18..30 ->
values()[category - 1]\n            else -> throw IllegalArgumentException(\"Category #\$category is not defined.\")\n        }\n    }\n\n }\"/>\n\n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.\n
* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*\n\npackage kotlin.text\n\n/**\n * The exception thrown when a character encoding or decoding error occurs.\n
*\n\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual open class
CharacterCodingException(message: String?) : Exception(message) {\n    actual constructor() : this(null)\n\n }\"/>\n\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n
* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*\n\npackage
kotlin.text\n\n/**\n * A mutable sequence of characters.\n *^n * String builder can be used to efficiently perform
multiple string manipulation operations.\n *^n\npublic actual class StringBuilder actual constructor(content: String) :
Appendable, CharSequence {\n    /**\n     * Constructs an empty string builder with the specified initial [capacity].\n
*\n     * In Kotlin/JS implementation of StringBuilder the initial capacity has no effect on the further performance
of operations.\n     *\n     actual constructor(capacity: Int) : this() {\n     }\n\n     /** Constructs a string builder that
contains the same characters as the specified [content] char sequence. *\n     actual constructor(content:
CharSequence) : this(content.toString()) {\n     }\n\n     /** Constructs an empty string builder. *\n     actual constructor() :
this(\"\")\n\n     private var string: String = if (content !== undefined) content else \"\"\n\n     actual override val
length: Int\n     get() = string.asDynamic().length\n\n     actual override fun get(index: Int): Char =\n

```

```

string.getOrElse(index) { throw IndexOutOfBoundsException("index: $index, length: $length}") }
actual
override fun subSequence(startIndex: Int, endIndex: Int): CharSequence = string.substring(startIndex, endIndex)
actual
override fun append(value: Char): String Builder {
    string += value
    return this
}
actual
override fun append(value: CharSequence?): String Builder {
    string += value.toString()
    return this
}
actual
override fun append(value: CharSequence?, startIndex: Int, endIndex: Int): String Builder =
    this.appendRange(value ?: "null", startIndex, endIndex)
/**
 * Reverses the contents of this string builder
and returns this instance.
 * Surrogate pairs included in this string builder are treated as single
characters.
 * Therefore, the order of the high-low surrogates is never reversed.
 * Note that the reverse
operation may produce new surrogate pairs that were unpaired low-surrogates and high-surrogates before the
operation.
 * For example, reversing "\uDC00\uD800" produces "\uD800\uDC00" which is a valid
surrogate pair.
 */
actual fun reverse(): String Builder {
    var reversed = ""
    var index =
string.length - 1
    while (index >= 0) {
        val low = string[index--]
        if (low.isLowSurrogate() &&
index >= 0) {
            val high = string[index--]
            if (high.isHighSurrogate()) {
                reversed =
reversed + high + low
            } else {
                reversed = reversed + low + high
            }
        } else {
            reversed += low
        }
    }
    string = reversed
    return this
}
/**
 * Appends the string representation of the specified object [value] to this string builder and returns this instance.
 * The overall effect is exactly as if the [value] were converted to a string by the `value.toString()` method,
 * and then that string was appended to this string builder.
 */
actual fun append(value: Any?): String Builder {
    string += value.toString()
    return this
}
/**
 * Appends the string representation of the
specified boolean [value] to this string builder and returns this instance.
 * The overall effect is exactly as
if the [value] were converted to a string by the `value.toString()` method,
 * and then that string was appended to
this string builder.
 */
@SinceKotlin("1.3")
actual fun append(value: Boolean): String Builder {
    string += value
    return this
}
/**
 * Appends characters in the specified character array [value] to
this string builder and returns this instance.
 * Characters are appended in order, starting at the index 0.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
actual fun append(value:
CharArray): String Builder {
    string += value.concatToString()
    return this
}
@Deprecated("Provided for binary compatibility.", level = DeprecationLevel.HIDDEN)
fun append(value:
String): String Builder = append(value)
/**
 * Appends the specified string [value] to this string builder and
returns this instance.
 * If [value] is `null`, then the four characters "null" are appended.
 */
@SinceKotlin("1.3")
actual fun append(value: String?): String Builder {
    this.string += value ?: "null"
    return this
}
/**
 * Returns the current capacity of this string builder.
 * The capacity is the
maximum length this string builder can have before an allocation occurs.
 * In Kotlin/JS implementation
of String Builder the value returned from this method may not indicate the actual size of the backing storage.
 */
@SinceKotlin("1.3")
// @ExperimentalStdlibApi
@Deprecated("Obtaining String Builder capacity is
not supported in JS and common code.", level = DeprecationLevel.ERROR)
actual fun capacity(): Int =
length
/**
 * Ensures that the capacity of this string builder is at least equal to the specified
[minimumCapacity].
 * If the current capacity is less than the [minimumCapacity], a new backing storage
is allocated with greater capacity.
 * Otherwise, this method takes no action and simply returns.
 * In
Kotlin/JS implementation of String Builder the size of the backing storage is not extended to comply the given
[minimumCapacity],
 * thus calling this method has no effect on the further performance of operations.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
actual fun
ensureCapacity(minimumCapacity: Int) {}
/**
 * Returns the index within this string builder of the
first occurrence of the specified [string].
 * Returns -1 if the specified [string] does not occur in this
string builder.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
actual
fun indexOf(string: String): Int = this.string.asDynamic().indexOf(string)
/**
 * Returns the index within
this string builder of the first occurrence of the specified [string],
 * starting at the specified [startIndex].
 * Returns -1 if the specified [string] does not occur in this string builder starting at the specified [startIndex].
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
actual fun indexOf(string:

```

```

String, startIndex: Int): Int = this.string.asDynamic().indexOf(string, startIndex)\n\n /**\n * Returns the index
within this string builder of the last occurrence of the specified [string].\n * The last occurrence of empty string
`\"\"` is considered to be at the index equal to `this.length`.\n *\n * Returns `-1` if the specified [string] does not
occur in this string builder.\n *\n @SinceKotlin("1.4")\n
@WasExperimental(ExperimentalStdlibApi::class)\n actual fun lastIndexOf(string: String): Int =
this.string.asDynamic().lastIndexOf(string)\n\n /**\n * Returns the index within this string builder of the last
occurrence of the specified [string],\n * starting from the specified [startIndex] toward the beginning.\n *\n *
Returns `-1` if the specified [string] does not occur in this string builder starting at the specified [startIndex].\n
*\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n actual fun
lastIndexOf(string: String, startIndex: Int): Int {\n    if (string.isEmpty() && startIndex < 0) return -1\n    return
this.string.asDynamic().lastIndexOf(string, startIndex)\n }\n\n /**\n * Inserts the string representation of the
specified boolean [value] into this string builder at the specified [index] and returns this instance.\n *\n * The
overall effect is exactly as if the [value] were converted to a string by the `value.toString()` method,\n * and then
that string was inserted into this string builder at the specified [index].\n *\n * @throws
IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.\n *\n
@SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n actual fun insert(index: Int, value:
Boolean): StringBuilder {\n    AbstractList.checkPositionIndex(index, length)\n\n    string = string.substring(0,
index) + value + string.substring(index)\n    return this\n }\n\n /**\n * Inserts the specified character [value]
into this string builder at the specified [index] and returns this instance.\n *\n * @throws
IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.\n *\n
@SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n actual fun insert(index: Int, value:
Char): StringBuilder {\n    AbstractList.checkPositionIndex(index, length)\n\n    string = string.substring(0,
index) + value + string.substring(index)\n    return this\n }\n\n /**\n * Inserts characters in the specified
character array [value] into this string builder at the specified [index] and returns this instance.\n *\n * The
inserted characters go in same order as in the [value] character array, starting at [index].\n *\n * @throws
IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.\n *\n
@SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n actual fun insert(index: Int, value:
CharArray): StringBuilder {\n    AbstractList.checkPositionIndex(index, length)\n\n    string =
string.substring(0, index) + value.concatToString() + string.substring(index)\n    return this\n }\n\n /**\n *
Inserts characters in the specified character sequence [value] into this string builder at the specified [index] and
returns this instance.\n *\n * The inserted characters go in the same order as in the [value] character sequence,
starting at [index].\n *\n * @param index the position in this string builder to insert at.\n * @param value the
character sequence from which characters are inserted. If [value] is `null`, then the four characters `\"null\"` are
inserted.\n *\n * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of
this string builder.\n *\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n actual fun
insert(index: Int, value: CharSequence?): StringBuilder {\n    AbstractList.checkPositionIndex(index,
length)\n\n    string = string.substring(0, index) + value.toString() + string.substring(index)\n    return this\n
}\n\n /**\n * Inserts the string representation of the specified object [value] into this string builder at the
specified [index] and returns this instance.\n *\n * The overall effect is exactly as if the [value] were converted
to a string by the `value.toString()` method,\n * and then that string was inserted into this string builder at the
specified [index].\n *\n * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the
length of this string builder.\n *\n @SinceKotlin("1.4")\n
@WasExperimental(ExperimentalStdlibApi::class)\n actual fun insert(index: Int, value: Any?): StringBuilder {\n
    AbstractList.checkPositionIndex(index, length)\n\n    string = string.substring(0, index) + value.toString() +
string.substring(index)\n    return this\n }\n\n @Deprecated("Provided for binary compatibility.", level =
DeprecationLevel.HIDDEN)\n fun insert(index: Int, value: String): StringBuilder = insert(index, value)\n\n /**\n
* Inserts the string [value] into this string builder at the specified [index] and returns this instance.\n *\n * If
[value] is `null`, then the four characters `\"null\"` are inserted.\n *\n * @throws IndexOutOfBoundsException

```

```

if [index] is less than zero or greater than the length of this string builder.\n    *\n    @SinceKotlin("1.4")\n    @WasExperimental(ExperimentalStdlibApi::class)\n    actual fun insert(index: Int, value: String?): StringBuilder\n    {\n        AbstractList.checkPositionIndex(index, length)\n        val toInsert = value ?: "null"\n        this.string =\n        this.string.substring(0, index) + toInsert + this.string.substring(index)\n        return this\n    }\n    /**\n     * Sets\n     * the length of this string builder to the specified [newLength].\n     * *\n     * If the [newLength] is less than the current\n     * length, it is changed to the specified [newLength].\n     * *\n     * Otherwise, null characters '\u0000' are appended to this\n     * string builder until its length is less than the [newLength].\n     * *\n     * Note that in Kotlin/JS [set] operator function\n     * has non-constant execution time complexity.\n     * *\n     * Therefore, increasing length of this string builder and then\n     * updating each character by index may slow down your program.\n     * *\n     * @throws\n     * IndexOutOfBoundsException or [IllegalArgumentException] if [newLength] is less than zero.\n     *\n     *\n     @SinceKotlin("1.4")\n     @WasExperimental(ExperimentalStdlibApi::class)\n     actual fun setLength(newLength:\n     Int) {\n         if (newLength < 0) {\n             throw IllegalArgumentException("Negative new length:\n             $newLength.")\n         }\n         if (newLength <= length) {\n             string = string.substring(0, newLength)\n         }\n         else {\n             for (i in length until newLength) {\n                 string += "\u0000"\n             }\n         }\n     }\n     /**\n     * Returns a new [String] that contains characters in this string builder at [startIndex] (inclusive) and up to the\n     * [length] (exclusive).\n     * *\n     * @throws IndexOutOfBoundsException if [startIndex] is less than zero or greater\n     * than the length of this string builder.\n     *\n     *\n     @SinceKotlin("1.4")\n     @WasExperimental(ExperimentalStdlibApi::class)\n     actual fun substring(startIndex: Int): String {\n     AbstractList.checkPositionIndex(startIndex, length)\n     return string.substring(startIndex)\n     }\n     /**\n     * Returns a new [String] that contains characters in this string builder at [startIndex] (inclusive) and up to the\n     * [endIndex] (exclusive).\n     * *\n     * @throws IndexOutOfBoundsException or [IllegalArgumentException] when\n     * [startIndex] or [endIndex] is out of range of this string builder indices or when `startIndex > endIndex`.\n     *\n     *\n     @SinceKotlin("1.4")\n     @WasExperimental(ExperimentalStdlibApi::class)\n     actual fun substring(startIndex:\n     Int, endIndex: Int): String {\n         AbstractList.checkBoundsIndexes(startIndex, endIndex, length)\n         return\n         string.substring(startIndex, endIndex)\n     }\n     /**\n     * Attempts to reduce storage used for this string builder.\n     *\n     * *\n     * If the backing storage of this string builder is larger than necessary to hold its current contents,\n     * *\n     * then\n     * it may be resized to become more space efficient.\n     * *\n     * Calling this method may, but is not required to, affect the\n     * value of the [capacity] property.\n     * *\n     * In Kotlin/JS implementation of StringBuilder the size of the backing\n     * storage is always equal to the length of the string builder.\n     *\n     *\n     @SinceKotlin("1.4")\n     @WasExperimental(ExperimentalStdlibApi::class)\n     actual fun trimToSize() {\n     }\n     override fun toString():\n     String = string\n     /**\n     * Clears the content of this string builder making it empty and returns this instance.\n     *\n     *\n     * @sample samples.text.Strings.clearStringBuilder\n     *\n     *\n     @SinceKotlin("1.3")\n     public fun clear():\n     StringBuilder {\n         string = ""\n         return this\n     }\n     /**\n     * Sets the character at the specified [index]\n     * to the specified [value].\n     * *\n     * @throws IndexOutOfBoundsException if [index] is out of bounds of this string\n     * builder.\n     *\n     *\n     @SinceKotlin("1.4")\n     @WasExperimental(ExperimentalStdlibApi::class)\n     public\n     operator fun set(index: Int, value: Char) {\n         AbstractList.checkElementIndex(index, length)\n         string =\n         string.substring(0, index) + value + string.substring(index + 1)\n     }\n     /**\n     * Replaces characters in the\n     * specified range of this string builder with characters in the specified string [value] and returns this instance.\n     *\n     *\n     * @param startIndex the beginning (inclusive) of the range to replace.\n     * *\n     * @param endIndex the end (exclusive)\n     * of the range to replace.\n     * *\n     * @param value the string to replace with.\n     * *\n     * @throws\n     * IndexOutOfBoundsException or [IllegalArgumentException] if [startIndex] is less than zero, greater than the length\n     * of this string builder, or `startIndex > endIndex`.\n     *\n     *\n     @SinceKotlin("1.4")\n     @WasExperimental(ExperimentalStdlibApi::class)\n     public fun setRange(startIndex: Int, endIndex: Int, value:\n     String): StringBuilder {\n         checkReplaceRange(startIndex, endIndex, length)\n         this.string =\n         this.string.substring(0, startIndex) + value + this.string.substring(endIndex)\n         return this\n     }\n     private fun\n     checkReplaceRange(startIndex: Int, endIndex: Int, length: Int) {\n         if (startIndex < 0 || startIndex > length) {\n             throw IndexOutOfBoundsException("startIndex: $startIndex, length: $length")\n         }\n         if (startIndex >\n         endIndex) {\n             throw IllegalArgumentException("startIndex($startIndex) > endIndex($endIndex)")\n         }\n     }\n
```

```

    }\n\n /**\n * Removes the character at the specified [index] from this string builder and returns this instance.\n *\n * If the `Char` at the specified [index] is part of a supplementary code point, this method does not remove the entire supplementary character.\n *\n * @param index the index of `Char` to remove.\n *\n * @throws IndexOutOfBoundsException if [index] is out of bounds of this string builder.\n *\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n public fun deleteAt(index: Int): StringBuilder {\n    AbstractList.checkElementIndex(index, length)\n    string = string.substring(0, index) + string.substring(index + 1)\n    return this\n }\n\n /**\n * Removes characters in the specified range from this string builder and returns this instance.\n *\n * @param startIndex the beginning (inclusive) of the range to remove.\n * @param endIndex the end (exclusive) of the range to remove.\n *\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] is out of range of this string builder indices or when `startIndex > endIndex`.\n *\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n public fun deleteRange(startIndex: Int, endIndex: Int): StringBuilder {\n    checkReplaceRange(startIndex, endIndex, length)\n    string = string.substring(0, startIndex) + string.substring(endIndex)\n    return this\n }\n\n /**\n * Copies characters from this string builder into the [destination] character array.\n *\n * @param destination the array to copy to.\n * @param destinationOffset the position in the array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the range to copy, 0 by default.\n * @param endIndex the end (exclusive) of the range to copy, length of this string builder by default.\n *\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this string builder indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n *\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n public fun toCharArray(destination: CharArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = this.length) {\n    AbstractList.checkBoundsIndexes(startIndex, endIndex, length)\n    AbstractList.checkBoundsIndexes(destinationOffset, destinationOffset + endIndex - startIndex, destination.size)\n    var dstIndex = destinationOffset\n    for (index in startIndex until endIndex) {\n        destination[dstIndex++] = string[index]\n    }\n }\n\n /**\n * Appends characters in a subarray of the specified character array [value] to this string builder and returns this instance.\n *\n * Characters are appended in order, starting at specified [startIndex].\n *\n * @param value the array from which characters are appended.\n * @param startIndex the beginning (inclusive) of the subarray to append.\n * @param endIndex the end (exclusive) of the subarray to append.\n *\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of the [value] array indices or when `startIndex > endIndex`.\n *\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n public fun appendRange(value: CharArray, startIndex: Int, endIndex: Int): StringBuilder {\n    string += value.concatToString(startIndex, endIndex)\n    return this\n }\n\n /**\n * Appends a subsequence of the specified character sequence [value] to this string builder and returns this instance.\n *\n * @param value the character sequence from which a subsequence is appended.\n * @param startIndex the beginning (inclusive) of the subsequence to append.\n * @param endIndex the end (exclusive) of the subsequence to append.\n *\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of the [value] character sequence indices or when `startIndex > endIndex`.\n *\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n public fun appendRange(value: CharSequence, startIndex: Int, endIndex: Int): StringBuilder {\n    val stringCsq = value.toString()\n    AbstractList.checkBoundsIndexes(startIndex, endIndex, stringCsq.length)\n    string += stringCsq.substring(startIndex, endIndex)\n    return this\n }\n\n /**\n * Inserts characters in a subarray of the specified character array [value] into this string builder at the specified [index] and returns this instance.\n *\n * The inserted characters go in same order as in the [value] array, starting at [index].\n *\n * @param index the position in this string builder to insert at.\n * @param value the array from which characters are inserted.\n * @param startIndex the beginning (inclusive) of the subarray to insert.\n * @param endIndex the end (exclusive)

```



```

DOWED_BY_MEMBER", "NOTHING_TO_INLINE")\npublic actual inline fun
StringBuilder.deleteRange(startIndex: Int, endIndex: Int): StringBuilder = this.deleteRange(startIndex,
endIndex)\n\n/**\n * Copies characters from this string builder into the [destination] character array.\n *\n * @param destination the array to copy to.\n * @param destinationOffset the position in the array to copy to, 0 by
default.\n * @param startIndex the beginning (inclusive) of the range to copy, 0 by default.\n * @param endIndex
the end (exclusive) of the range to copy, length of this string builder by default.\n *\n * @throws
IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this
string builder indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the
subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is
out of the [destination] array indices range.\n
*/\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@Suppress("EXTENSION_SHA
DOWED_BY_MEMBER", "NOTHING_TO_INLINE",
"ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual inline fun
StringBuilder.toCharArray(destination: CharArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int =
this.length) =\n    this.toCharArray(destination, destinationOffset, startIndex, endIndex)\n\n/**\n * Appends
characters in a subarray of the specified character array [value] to this string builder and returns this instance.\n *\n * Characters are appended in order, starting at specified [startIndex].\n *\n * @param value the array from which
characters are appended.\n * @param startIndex the beginning (inclusive) of the subarray to append.\n * @param
endIndex the end (exclusive) of the subarray to append.\n *\n * @throws IndexOutOfBoundsException or
[IllegalArgumentException] when [startIndex] or [endIndex] is out of range of the [value] array indices or when
`startIndex > endIndex`.\n
*/\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@Suppress("EXTENSION_SHA
DOWED_BY_MEMBER", "NOTHING_TO_INLINE")\npublic actual inline fun
StringBuilder.appendRange(value: CharArray, startIndex: Int, endIndex: Int): StringBuilder =\n    this.appendRange(value, startIndex, endIndex)\n\n/**\n * Appends a subsequence of the specified character
sequence [value] to this string builder and returns this instance.\n *\n * @param value the character sequence from
which a subsequence is appended.\n * @param startIndex the beginning (inclusive) of the subsequence to append.\n
* @param endIndex the end (exclusive) of the subsequence to append.\n *\n * @throws
IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of the
[value] character sequence indices or when `startIndex > endIndex`.\n
*/\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@Suppress("EXTENSION_SHA
DOWED_BY_MEMBER", "NOTHING_TO_INLINE")\npublic actual inline fun
StringBuilder.appendRange(value: CharSequence, startIndex: Int, endIndex: Int): StringBuilder =\n    this.appendRange(value, startIndex, endIndex)\n\n/**\n * Inserts characters in a subarray of the specified character
array [value] into this string builder at the specified [index] and returns this instance.\n *\n * The inserted characters
go in same order as in the [value] array, starting at [index].\n *\n * @param index the position in this string builder
to insert at.\n * @param value the array from which characters are inserted.\n * @param startIndex the beginning
(inclusive) of the subarray to insert.\n * @param endIndex the end (exclusive) of the subarray to insert.\n *\n *
@throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of
range of the [value] array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException if
[index] is less than zero or greater than the length of this string builder.\n
*/\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@Suppress("EXTENSION_SHA
DOWED_BY_MEMBER", "NOTHING_TO_INLINE")\npublic actual inline fun
StringBuilder.insertRange(index: Int, value: CharArray, startIndex: Int, endIndex: Int): StringBuilder =\n    this.insertRange(index, value, startIndex, endIndex)\n\n/**\n * Inserts characters in a subsequence of the specified
character sequence [value] into this string builder at the specified [index] and returns this instance.\n *\n * The
inserted characters go in the same order as in the [value] character sequence, starting at [index].\n *\n * @param
index the position in this string builder to insert at.\n * @param value the character sequence from which a

```

subsequence is inserted.\n * @param startIndex the beginning (inclusive) of the subsequence to insert.\n * @param endIndex the end (exclusive) of the subsequence to insert.\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of the [value] character sequence indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.\n

```

*\/n@SinceKotlin("1.4")n@WasExperimental(ExperimentalStdlibApi::class)n@Suppress("EXTENSION_SHA
DOWED_BY_MEMBER", "NOTHING_TO_INLINE")npublic actual inline fun
StringBuilder.insertRange(index: Int, value: CharSequence, startIndex: Int, endIndex: Int): StringBuilder =n
this.insertRange(index, value, startIndex, endIndex)n"/n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin
Programming Language contributors.n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.n */npackage kotlin.textn/n/**n * Returns `true` if the content of this
string is equal to the word `true`, ignoring case, and `false` otherwise.n */n@Deprecated("Use Kotlin compiler
1.4 to avoid deprecation warning.")n@DeprecatedSinceKotlin(hiddenSince =
"1.4")n@kotlin.internal.InlineOnlynpublic actual inline fun String.toBoolean(): Boolean =
this.toBoolean()n/n/**n * Returns `true` if this string is not `null` and its content is equal to the word `true`,
ignoring case, and `false` otherwise.n */n * There are also strict versions of the function available on non-nullable
String, [toBooleanStrict] and [toBooleanStrictOrNull].n */n@SinceKotlin("1.4")npublic actual fun
String?.toBoolean(): Boolean = this != null && this.lowercase() == "true"n/n/**n * Parses the string as a signed
[Byte] number and returns the result.n * @throws NumberFormatException if the string is not a valid
representation of a number.n */npublic actual fun String.toByte(): Byte = toByteOrNull() ?:
numberFormatError(this)n/n/**n * Parses the string as a signed [Byte] number and returns the result.n * @throws
NumberFormatException if the string is not a valid representation of a number.n * @throws
IllegalArgumentException when [radix] is not a valid radix for string to number conversion.n */npublic actual fun
String.toByte(radix: Int): Byte = toByteOrNull(radix) ?: numberFormatError(this)n/n/**n * Parses the string as a
[Short] number and returns the result.n * @throws NumberFormatException if the string is not a valid
representation of a number.n */npublic actual fun String.toShort(): Short = toShortOrNull() ?:
numberFormatError(this)n/n/**n * Parses the string as a [Short] number and returns the result.n * @throws
NumberFormatException if the string is not a valid representation of a number.n * @throws
IllegalArgumentException when [radix] is not a valid radix for string to number conversion.n */npublic actual fun
String.toShort(radix: Int): Short = toShortOrNull(radix) ?: numberFormatError(this)n/n/**n * Parses the string as
an [Int] number and returns the result.n * @throws NumberFormatException if the string is not a valid
representation of a number.n */npublic actual fun String.toInt(): Int = toIntOrNull() ?:
numberFormatError(this)n/n/**n * Parses the string as an [Int] number and returns the result.n * @throws
NumberFormatException if the string is not a valid representation of a number.n * @throws
IllegalArgumentException when [radix] is not a valid radix for string to number conversion.n */npublic actual fun
String.toInt(radix: Int): Int = toIntOrNull(radix) ?: numberFormatError(this)n/n/**n * Parses the string as a [Long]
number and returns the result.n * @throws NumberFormatException if the string is not a valid representation of a
number.n */npublic actual fun String.toLong(): Long = toLongOrNull() ?: numberFormatError(this)n/n/**n *
Parses the string as a [Long] number and returns the result.n * @throws NumberFormatException if the string is not
a valid representation of a number.n * @throws IllegalArgumentException when [radix] is not a valid radix for
string to number conversion.n */npublic actual fun String.toLong(radix: Int): Long = toLongOrNull(radix) ?:
numberFormatError(this)n/n/**n * Parses the string as a [Double] number and returns the result.n * @throws
NumberFormatException if the string is not a valid representation of a number.n */npublic actual fun
String.toDouble(): Double = +(this.asDynamic()).unsafeCast<Double>().also {n if (it.isNaN()) && !this.isNaN()
|| it == 0.0 && this.isBlank()n numberFormatError(this)n}n/n/**n * Parses the string as a [Float] number
and returns the result.n * @throws NumberFormatException if the string is not a valid representation of a
number.n */n@kotlin.internal.InlineOnlynpublic actual inline fun String.toFloat(): Float =
toDouble().unsafeCast<Float>()n/n/**n * Parses the string as a [Double] number and returns the resultn * or `null`

```

if the string is not a valid representation of a number.

```

public actual fun String.toDoubleOrNull(): Double? =
    +(this.asDynamic()).unsafeCast<Double>().takeIf { !it.isNaN() && !this.isNaN() || it == 0.0 &&
    this.isBlank() }

```

* Parses the string as a [Float] number and returns the result or `null` if the string is not a valid representation of a number.

```

@kotlin.internal.InlineOnly
public actual inline fun
String.toFloatOrNull(): Float? = toDoubleOrNull().unsafeCast<Float?>()

```

* Returns a string representation of this [Byte] value in the specified [radix].

```

@throws IllegalArgumentException when [radix] is not a valid
radix for number to string conversion.
@SinceKotlin("1.2")
@kotlin.internal.InlineOnly
public actual
inline fun Byte.toString(radix: Int): String = this.toInt().toString(radix)

```

* Returns a string representation of this [Short] value in the specified [radix].

```

@throws IllegalArgumentException when [radix] is not a valid
radix for number to string conversion.
@SinceKotlin("1.2")
@kotlin.internal.InlineOnly
public actual
inline fun Short.toString(radix: Int): String = this.toInt().toString(radix)

```

* Returns a string representation of this [Int] value in the specified [radix].

```

@throws IllegalArgumentException when [radix] is not a valid
radix for number to string conversion.
@SinceKotlin("1.2")
public actual fun Int.toString(radix: Int): String =
    asDynamic().toString(checkRadix(radix))
private fun String.isNaN(): Boolean = when (this.lowercase()) {
    "nan", "+nan", "-nan" -> true
    else -> false
}

```

* Checks whether the given [radix] is valid radix for string to number and number to string conversion.

```

@PublishedApi
internal actual fun checkRadix(radix: Int):
Int {
    if (radix !in 2..36) {
        throw IllegalArgumentException("radix $radix was not in valid range 2..36")
    }
    return radix
}
internal actual fun digitOf(char: Char, radix: Int): Int = when {
    char >= '0' && char <= '9' -> char - '0'
    char >= 'A' && char <= 'Z' -> char - 'A' + 10
    char >= 'a' && char <= 'z' -> char - 'a' + 10
    char < "\u0080" -> -1
    char >= "\uFF21" && char <= "\uFF3A" -> char - "\uFF21" + 10 // full-width latin capital
    letter
    char >= "\uFF41" && char <= "\uFF5A" -> char - "\uFF41" + 10 // full-width latin small letter
    else ->
    char.digitToIntImpl()
}.let { if (it >= radix) -1 else it }

```

* Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.

* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

```

package kotlin.text
import kotlin.js.RegExp

```

* Provides enumeration values to use to set regular expression options.

```

public actual enum class RegexOptions(val value:
String) {
    /** Enables case-insensitive matching. */
    IGNORE_CASE("i"),
    /** Enables multiline
mode.
    * In multiline mode the expressions `^` and `$` match just after or just before,
    * respectively, a
    line terminator or the end of the input sequence. */
    MULTILINE("m")
}
private fun
Iterable<RegexOption>.toFlags(prepend: String): String = joinToString("\", prefix = prepend) { it.value
}

```

* Represents the results from a single capturing group within a [MatchResult] of [Regex].

```

@param value The value of captured group.
public actual data class MatchGroup(actual val value:
String)

```

* Represents a compiled regular expression.

* Provides functions to match strings in text with a pattern, replace the found occurrences and split text around matches.

* For pattern syntax reference see [MDN RegExp](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/RegExp#Special_characters_meaning_in_regular_expressions) and [http://www.w3schools.com/jsref/jsref_obj_regexp.asp](https://www.w3schools.com/jsref/jsref_obj_regexp.asp).

* Note that `RegExp` objects under the hood are constructed with [the `u` flag](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/RegExp/unicode) that enables Unicode-related features in regular expressions. This also makes the pattern syntax more strict, for example, prohibiting unnecessary escape sequences.

```

@constructor Creates a regular expression from the
specified [pattern] string and the specified set of [options].
public actual class Regex actual
constructor(pattern: String, options: Set<RegexOption>) {
    /** Creates a regular expression from the specified
[pattern] string and the specified single [option]. */
    public actual constructor(pattern: String, option:
RegexOption) : this(pattern, setOf(option))
    /** Creates a regular expression from the specified [pattern] string
and the default options. */
    public actual constructor(pattern: String) : this(pattern, emptySet())
    /** The
pattern string of this regular expression. */
    public actual val pattern: String = pattern
    /** The set of options
that were used to create this regular expression. */
    public actual val options: Set<RegexOption> =

```

```

options.toSet()\n private val nativePattern: RegExp = RegExp(pattern, options.toFlags("gu"))\n private var
nativeStickyPattern: RegExp? = null\n private fun initStickyPattern(): RegExp =\n nativeStickyPattern ?:
RegExp(pattern, options.toFlags("yu")).also { nativeStickyPattern = it }\n\n private var
nativeMatchesEntirePattern: RegExp? = null\n private fun initMatchesEntirePattern(): RegExp =\n
nativeMatchesEntirePattern ?: run {\n     if (pattern.startsWith('^') && pattern.endsWith('$'))\n
nativePattern\n     else\n         return RegExp("^${pattern.trimStart('^').trimEnd('$')}\$"),
options.toFlags("gu"))\n     }.also { nativeMatchesEntirePattern = it }\n\n /** Indicates whether the regular
expression matches the entire [input]. */\n public actual infix fun matches(input: CharSequence): Boolean {\n
nativePattern.reset()\n     val match = nativePattern.exec(input.toString())\n     return match != null &&
match.index == 0 && nativePattern.lastIndex == input.length\n } \n\n /** Indicates whether the regular
expression can find at least one match in the specified [input]. */\n public actual fun containsMatchIn(input:
CharSequence): Boolean {\n     nativePattern.reset()\n     return nativePattern.test(input.toString())\n } \n\n
@SinceKotlin("1.5")\n @ExperimentalStdlibApi\n public actual fun matchesAt(input: CharSequence, index:
Int): Boolean {\n     if (index < 0 || index > input.length) {\n         throw IndexOutOfBoundsException("index
out of bounds: $index, input length: ${input.length}")\n     }\n     val pattern = initStickyPattern()\n
pattern.lastIndex = index\n     return pattern.test(input.toString())\n } \n\n /**\n * Returns the first match of a
regular expression in the [input], beginning at the specified [startIndex].\n * \n * @param startIndex An index to
start search with, by default 0. Must be not less than zero and not greater than `input.length()`\n * @return An
instance of [MatchResult] if match was found or `null` otherwise.\n * @throws IndexOutOfBoundsException if
[startIndex] is less than zero or greater than the length of the [input] char sequence.\n * @sample
samples.text.Regexps.find\n */\n
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\n public actual fun find(input:
CharSequence, startIndex: Int = 0): MatchResult? {\n     if (startIndex < 0 || startIndex > input.length) {\n
throw IndexOutOfBoundsException("Start index out of bounds: $startIndex, input length: ${input.length}")\n
}\n     return nativePattern.findNext(input.toString(), startIndex, nativePattern)\n } \n\n /**\n * Returns a
sequence of all occurrences of a regular expression within the [input] string, beginning at the specified
[startIndex].\n * \n * @throws IndexOutOfBoundsException if [startIndex] is less than zero or greater than the
length of the [input] char sequence.\n * \n * @sample samples.text.Regexps.findAll\n */\n
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\n public actual fun findAll(input:
CharSequence, startIndex: Int = 0): Sequence<MatchResult> {\n     if (startIndex < 0 || startIndex > input.length)
{\n         throw IndexOutOfBoundsException("Start index out of bounds: $startIndex, input length:
${input.length}")\n     }\n     return generateSequence({ find(input, startIndex) }, { match -> match.next() })\n
}\n\n /**\n * Attempts to match the entire [input] CharSequence against the pattern.\n * \n * @return An
instance of [MatchResult] if the entire input matches or `null` otherwise.\n */\n public actual fun
matchEntire(input: CharSequence): MatchResult? =\n     initMatchesEntirePattern().findNext(input.toString(), 0,
nativePattern)\n\n @SinceKotlin("1.5")\n @ExperimentalStdlibApi\n public actual fun matchAt(input:
CharSequence, index: Int): MatchResult? {\n     if (index < 0 || index > input.length) {\n         throw
IndexOutOfBoundsException("index out of bounds: $index, input length: ${input.length}")\n     }\n     return
initStickyPattern().findNext(input.toString(), index, nativePattern)\n } \n\n\n /**\n * Replaces all occurrences
of this regular expression in the specified [input] string with specified [replacement] expression.\n * \n * The
replacement string may contain references to the captured groups during a match. Occurrences of `\$index`\n * in
the replacement string will be substituted with the subsequences corresponding to the captured groups with the
specified index.\n * \n * The first digit after '$' is always treated as part of group reference. Subsequent digits
are incorporated\n * into `index` only if they would form a valid group reference. Only the digits '0'..'9' are considered
as potential components\n * of the group reference. Note that indexes of captured groups start from 1, and the
group with index 0 is the whole match.\n * \n * Backslash character '\\' can be used to include the succeeding
character as a literal in the replacement string, e.g. `\\$` or `\\\\`.\n * [Regex.escapeReplacement] can be used if
[replacement] have to be treated as a literal string.\n * \n * Note that referring named capturing groups by name

```

is currently not supported in Kotlin/JS.

`* However, you can still refer them by index.`

`* @param input the char sequence to find matches of this regular expression in`

`* @param replacement the expression to replace found matches with`

`* @return the result of replacing each occurrence of this regular expression in [input] with the result of evaluating the [replacement] expression`

`* @throws RuntimeException if [replacement] expression is malformed, or capturing group with specified `name` or `index` does not exist`

```

public actual fun
replace(input: CharSequence, replacement: String): String {
    if (!replacement.contains("\\\\") &&
        !replacement.contains('$')) {
        return input.toString().nativeReplace(nativePattern, replacement)
    }
    return replace(input) { substituteGroupRefs(it, replacement) }
}
/**
 * Replaces all occurrences of this
regular expression in the specified [input] string with the result of
 * the given function [transform] that takes
[MatchResult] and returns a string to be used as a
 * replacement for that match.
*/
public actual fun
replace(input: CharSequence, transform: (MatchResult) -> CharSequence): String {
    var match = find(input)
    if (match == null) return input.toString()
    var lastStart = 0
    val length = input.length
    val sb =
StringBuilder(length)
    do {
        val foundMatch = match!!
        sb.append(input, lastStart,
foundMatch.range.start)
        sb.append(transform(foundMatch))
        lastStart =
foundMatch.range.endInclusive + 1
        match = foundMatch.next()
    } while (lastStart < length && match
!= null)
    if (lastStart < length) {
        sb.append(input, lastStart, length)
    }
    return
sb.toString()
}
/**
 * Replaces the first occurrence of this regular expression in the specified [input]
string with specified [replacement] expression.
 * The replacement string may contain references to the
captured groups during a match. Occurrences of `index`
 * in the replacement string will be substituted with the
subsequences corresponding to the captured groups with the specified index.
 * The first digit after '$' is always
treated as part of group reference. Subsequent digits are incorporated
 * into `index` only if they would form a
valid group reference. Only the digits '0'..'9' are considered as potential components
 * of the group reference.
Note that indexes of captured groups start from 1, and the group with index 0 is the whole match.
 * Backslash character '\\' can be used to include the succeeding character as a literal in the replacement string, e.g. '\\$'
or '\\\\'.
 * [Regex.escapeReplacement] can be used if [replacement] have to be treated as a literal string.
 * Note that referring named capturing groups by name is not supported currently in Kotlin/JS.
 * However, you
can still refer them by index.
 * @param input the char sequence to find a match of this regular expression
in
 * @param replacement the expression to replace the found match with
 * @return the result of replacing
the first occurrence of this regular expression in [input] with the result of evaluating the [replacement] expression
 * @throws RuntimeException if [replacement] expression is malformed, or capturing group with specified `name`
or `index` does not exist
*/
public actual fun replaceFirst(input: CharSequence, replacement: String): String
{
    if (!replacement.contains("\\\\") && !replacement.contains('$')) {
        val nonGlobalOptions =
options.toFlags("u")
        return input.toString().nativeReplace(RegExp(pattern, nonGlobalOptions),
replacement)
    }
    val match = find(input) ?: return input.toString()
    return buildString {
        append(input.substring(0, match.range.first))
        append(substituteGroupRefs(match, replacement))
        append(input.substring(match.range.last + 1, input.length))
    }
}
/**
 * Splits the [input]
CharSequence to a list of strings around matches of this regular expression.
 * @param limit Non-negative
value specifying the maximum number of substrings the string can be split to.
 * Zero by default means no limit
is set.
 * @Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun
split(input: CharSequence, limit: Int = 0): List<String> {
    requireNonNegativeLimit(limit)
    val matches =
findAll(input).let { if (limit == 0) it else it.take(limit - 1) }
    val result = mutableListOf<String>()
    var
lastStart = 0
    for (match in matches) {
        result.add(input.subSequence(lastStart,
match.range.start).toString())
        lastStart = match.range.endInclusive + 1
    }
    result.add(input.subSequence(lastStart, input.length).toString())
    return result
}
/**
 * Splits the
[input] CharSequence to a sequence of strings around matches of this regular expression.
 * @param limit
Non-negative value specifying the maximum number of substrings the string can be split to.
 * Zero by default
means no limit is set.
 * @sample samples.text.Regexps.splitToSequence
 * @SinceKotlin("1.6")
 * @WasExperimental(ExperimentalStdlibApi::class)

```



```

reference\)\n\n        val endIndex = replacement.readGroupIndex(index, match.groupValues.size)\n        val\n        groupIndex = replacement.substring(index, endIndex).toInt()\n        if (groupIndex >=\n        match.groupValues.size)\n            throw IndexOutOfBoundsException("Group with index $groupIndex does not\n            exist")\n        result.append(match.groupValues[groupIndex])\n        index = endIndex\n    } else {\n    result.append(char)\n    }\n    }\n    return result.toString()\n}\n\nprivate fun String.readGroupIndex(startIndex:\nInt, groupCount: Int): Int {\n    // at least one digit after '$' is always captured\n    var index = startIndex + 1\n    var\n    groupIndex = this[startIndex] - '0'\n    // capture the largest valid group index\n    while (index < length &&\n    this[index] in '0'..'9') {\n        val newGroupIndex = (groupIndex * 10) + (this[index] - '0')\n        if (newGroupIndex\n        in 0 until groupCount) {\n            groupIndex = newGroupIndex\n            index++\n        } else {\n            break\n        }\n    }\n    return index\n}"/**\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language\n        contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the\n        license/LICENSE.txt file.\n
```

```

*/\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("StringsKt")\n@file:Suppress("EXTENSI\nON_SHADOWED_BY_MEMBER")\n\npackage kotlin.text\n\nimport kotlin.contracts.*\n\n/**\n * A mutable\n        sequence of characters.\n * String builder can be used to efficiently perform multiple string manipulation\n        operations.\n */\nexpect class StringBuilder : Appendable, CharSequence {\n    /** Constructs an empty string\n        builder. *\n    constructor()\n    /** Constructs an empty string builder with the specified initial [capacity]. *\n    constructor(capacity: Int)\n    /** Constructs a string builder that contains the same characters as the specified\n        [content] char sequence. *\n    constructor(content: CharSequence)\n    /** Constructs a string builder that\n        contains the same characters as the specified [content] string. *\n    @SinceKotlin("1.3")\n\n    @ExperimentalStdlibApi\n    constructor(content: String)\n    override val length: Int\n    override operator fun\n        get(index: Int): Char\n    override fun subSequence(startIndex: Int, endIndex: Int): CharSequence\n    override\n        fun append(value: Char): StringBuilder\n    override fun append(value: CharSequence?): StringBuilder\n    override\n        fun append(value: CharSequence?, startIndex: Int, endIndex: Int): StringBuilder\n    /**\n        * Reverses the\n        contents of this string builder and returns this instance.\n        *\n        * Surrogate pairs included in this string builder\n        are treated as single characters.\n        *\n        * Therefore, the order of the high-low surrogates is never reversed.\n        *\n        * Note that the reverse operation may produce new surrogate pairs that were unpaired low-surrogates and high-\n        surrogates before the operation.\n        *\n        * For example, reversing `"\uDC00\uD800"` produces `"\uD800\uDC00"`\n        which is a valid surrogate pair.\n        *\n        fun reverse(): StringBuilder\n    /**\n        * Appends the string\n        representation of the specified object [value] to this string builder and returns this instance.\n        *\n        * The overall\n        effect is exactly as if the [value] were converted to a string by the `value.toString()` method,\n        *\n        * and then that\n        string was appended to this string builder.\n        *\n        fun append(value: Any?): StringBuilder\n    /**\n        * Appends the string representation of the specified boolean [value] to this string builder and returns this instance.\n        *\n        * The overall\n        effect is exactly as if the [value] were converted to a string by the `value.toString()` method,\n        *\n        * and then that\n        string was appended to this string builder.\n        *\n        @SinceKotlin("1.3")\n        fun append(value:\n        Boolean): StringBuilder\n    /**\n        * Appends characters in the specified character array [value] to this string\n        builder and returns this instance.\n        *\n        * Characters are appended in order, starting at the index 0.\n        *\n        @SinceKotlin("1.4")\n        @WasExperimental(ExperimentalStdlibApi::class)\n        fun append(value: CharArray):\n        StringBuilder\n    /**\n        * Appends the specified string [value] to this string builder and returns this instance.\n        *\n        * If [value] is `null`, then the four characters `null` are appended.\n        *\n        @SinceKotlin("1.3")\n        fun\n        append(value: String?): StringBuilder\n    /**\n        * Returns the current capacity of this string builder.\n        *\n        * The\n        capacity is the maximum length this string builder can have before an allocation occurs.\n        *\n        @SinceKotlin("1.3")\n        @ExperimentalStdlibApi\n        @Deprecated("Obtaining StringBuilder capacity is not\n        supported in JS and common code.", level = DeprecationLevel.ERROR)\n        fun capacity(): Int\n    /**\n        * Ensures that the capacity of this string builder is at least equal to the specified [minimumCapacity].\n        *\n        * If\n        the current capacity is less than the [minimumCapacity], a new backing storage is allocated with greater capacity.\n        *\n        * Otherwise, this method takes no action and simply returns.\n        *\n        @SinceKotlin("1.4")\n        @WasExperimental(ExperimentalStdlibApi::class)\n        fun ensureCapacity(minimumCapacity: Int)\n    /**\n        *

```

Returns the index within this string builder of the first occurrence of the specified [string].\n * Returns -1 if the specified [string] does not occur in this string builder.\n */\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n fun indexOf(string: String): Int\n /**\n * Returns the index within this string builder of the first occurrence of the specified [string],\n * starting at the specified [startIndex].\n * Returns -1 if the specified [string] does not occur in this string builder starting at the specified [startIndex].\n */\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n fun indexOf(string: String, startIndex: Int): Int\n /**\n * Returns the index within this string builder of the last occurrence of the specified [string].\n * The last occurrence of empty string `""` is considered to be at the index equal to `this.length`.\n * Returns -1 if the specified [string] does not occur in this string builder.\n */\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n fun lastIndexOf(string: String): Int\n /**\n * Returns the index within this string builder of the last occurrence of the specified [string],\n * starting from the specified [startIndex] toward the beginning.\n * Returns -1 if the specified [string] does not occur in this string builder starting at the specified [startIndex].\n */\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n fun lastIndexOf(string: String, startIndex: Int): Int\n /**\n * Inserts the string representation of the specified boolean [value] into this string builder at the specified [index] and returns this instance.\n * The overall effect is exactly as if the [value] were converted to a string by the `value.toString()` method,\n * and then that string was inserted into this string builder at the specified [index].\n * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.\n */\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n fun insert(index: Int, value: Boolean): String\n /**\n * Inserts the specified character [value] into this string builder at the specified [index] and returns this instance.\n * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.\n */\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n fun insert(index: Int, value: Char): String\n /**\n * Inserts characters in the specified character array [value] into this string builder at the specified [index] and returns this instance.\n * The inserted characters go in same order as in the [value] character array, starting at [index].\n * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.\n */\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n fun insert(index: Int, value: CharArray): String\n /**\n * Inserts characters in the specified character sequence [value] into this string builder at the specified [index] and returns this instance.\n * The inserted characters go in the same order as in the [value] character sequence, starting at [index].\n * @param index the position in this string builder to insert at.\n * @param value the character sequence from which characters are inserted. If [value] is `null`, then the four characters `"\u0000\u0000\u0000\u0000"` are inserted.\n * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.\n */\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n fun insert(index: Int, value: CharSequence?): String\n /**\n * Inserts the string representation of the specified object [value] into this string builder at the specified [index] and returns this instance.\n * The overall effect is exactly as if the [value] were converted to a string by the `value.toString()` method,\n * and then that string was inserted into this string builder at the specified [index].\n */\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n fun insert(index: Int, value: Any?): String\n /**\n * Inserts the string [value] into this string builder at the specified [index] and returns this instance.\n * If [value] is `null`, then the four characters `"\u0000\u0000\u0000\u0000"` are inserted.\n * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.\n */\n @SinceKotlin("1.4")\n @WasExperimental(ExperimentalStdlibApi::class)\n fun insert(index: Int, value: String?): String\n /**\n * Sets the length of this string builder to the specified [newLength].\n * If the [newLength] is less than the current length, it is changed to the specified [newLength].\n * Otherwise, null characters `'\u0000'` are appended to this string builder until its length is less than the [newLength].\n * Note that in Kotlin/JS [set] operator function has non-constant execution time complexity.\n * Therefore, increasing length of this string

builder and then updating each character by index may slow down your program.

```

    * @throws IndexOutOfBoundsException or [IllegalArgumentException] if [newLength] is less than zero.
    *
    * @SinceKotlin("1.4")
    * @WasExperimental(ExperimentalStdlibApi::class)
    fun setLength(newLength: Int)
    /**
     * Returns a new [String] that contains characters in this string builder at [startIndex] (inclusive) and up to the [length] (exclusive).
     *
     * @throws IndexOutOfBoundsException if [startIndex] is less than zero or greater than the length of this string builder.
     *
     * @SinceKotlin("1.4")
     * @WasExperimental(ExperimentalStdlibApi::class)
    fun substring(startIndex: Int): String
    /**
     * Returns a new [String] that contains characters in this string builder at [startIndex] (inclusive) and up to the [endIndex] (exclusive).
     *
     * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this string builder indices or when `startIndex > endIndex`.
     *
     * @SinceKotlin("1.4")
     * @WasExperimental(ExperimentalStdlibApi::class)
    fun substring(startIndex: Int, endIndex: Int): String
    /**
     * Attempts to reduce storage used for this string builder.
     *
     * If the backing storage of this string builder is larger than necessary to hold its current contents, then it may be resized to become more space efficient.
     *
     * Calling this method may, but is not required to, affect the value of the [capacity] property.
     *
     * @SinceKotlin("1.4")
     * @WasExperimental(ExperimentalStdlibApi::class)
    fun trimToSize()
    /**
     * Clears the content of this string builder making it empty and returns this instance.
     *
     * @sample samples.text.Strings.clearStringBuilder
     *
     * @SinceKotlin("1.3")
    public expect fun StringBuilder.clear(): StringBuilder
    /**
     * Sets the character at the specified [index] to the specified [value].
     *
     * @throws IndexOutOfBoundsException if [index] is out of bounds of this string builder.
     *
     * @SinceKotlin("1.4")
     * @WasExperimental(ExperimentalStdlibApi::class)
    public expect operator fun StringBuilder.set(index: Int, value: Char)
    /**
     * Replaces characters in the specified range of this string builder with characters in the specified string [value] and returns this instance.
     *
     * @param startIndex the beginning (inclusive) of the range to replace.
     * @param endIndex the end (exclusive) of the range to replace.
     * @param value the string to replace with.
     *
     * @throws IndexOutOfBoundsException or [IllegalArgumentException] if [startIndex] is less than zero, greater than the length of this string builder, or `startIndex > endIndex`.
     *
     * @SinceKotlin("1.4")
     * @WasExperimental(ExperimentalStdlibApi::class)
    public expect fun StringBuilder.setRange(startIndex: Int, endIndex: Int, value: String): StringBuilder
    /**
     * Removes the character at the specified [index] from this string builder and returns this instance.
     *
     * If the `Char` at the specified [index] is part of a supplementary code point, this method does not remove the entire supplementary character.
     *
     * @param index the index of `Char` to remove.
     *
     * @throws IndexOutOfBoundsException if [index] is out of bounds of this string builder.
     *
     * @SinceKotlin("1.4")
     * @WasExperimental(ExperimentalStdlibApi::class)
    public expect fun StringBuilder.deleteAt(index: Int): StringBuilder
    /**
     * Removes characters in the specified range from this string builder and returns this instance.
     *
     * @param startIndex the beginning (inclusive) of the range to remove.
     * @param endIndex the end (exclusive) of the range to remove.
     *
     * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] is out of range of this string builder indices or when `startIndex > endIndex`.
     *
     * @SinceKotlin("1.4")
     * @WasExperimental(ExperimentalStdlibApi::class)
    public expect fun StringBuilder.deleteRange(startIndex: Int, endIndex: Int): StringBuilder
    /**
     * Copies characters from this string builder into the [destination] character array.
     *
     * @param destination the array to copy to.
     * @param destinationOffset the position in the array to copy to, 0 by default.
     * @param startIndex the beginning (inclusive) of the range to copy, 0 by default.
     * @param endIndex the end (exclusive) of the range to copy, length of this string builder by default.
     *
     * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this string builder indices or when `startIndex > endIndex`.
     *
     * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset], or when that index is out of the [destination] array indices range.
     *
     * @SinceKotlin("1.4")
     * @WasExperimental(ExperimentalStdlibApi::class)
    public expect fun StringBuilder.toCharArray(destination: CharArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int =

```

this.length)\n\n/**\n * Appends characters in a subarray of the specified character array [value] to this string builder and returns this instance.\n * Characters are appended in order, starting at specified [startIndex].\n * @param value the array from which characters are appended.\n * @param startIndex the beginning (inclusive) of the subarray to append.\n * @param endIndex the end (exclusive) of the subarray to append.\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of the [value] array indices or when `startIndex > endIndex`.\n

*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
StringBuilder.appendRange(value: CharArray, startIndex: Int, endIndex: Int): StringBuilder\n\n/**\n * Appends a subsequence of the specified character sequence [value] to this string builder and returns this instance.\n * @param value the character sequence from which a subsequence is appended.\n * @param startIndex the beginning (inclusive) of the subsequence to append.\n * @param endIndex the end (exclusive) of the subsequence to append.\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of the [value] character sequence indices or when `startIndex > endIndex`.\n

*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
StringBuilder.appendRange(value: CharSequence, startIndex: Int, endIndex: Int): StringBuilder\n\n/**\n * Inserts characters in a subarray of the specified character array [value] into this string builder at the specified [index] and returns this instance.\n * The inserted characters go in same order as in the [value] array, starting at [index].\n * @param index the position in this string builder to insert at.\n * @param value the array from which characters are inserted.\n * @param startIndex the beginning (inclusive) of the subarray to insert.\n * @param endIndex the end (exclusive) of the subarray to insert.\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of the [value] array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.\n

*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
StringBuilder.insertRange(index: Int, value: CharArray, startIndex: Int, endIndex: Int): StringBuilder\n\n/**\n * Inserts characters in a subsequence of the specified character sequence [value] into this string builder at the specified [index] and returns this instance.\n * The inserted characters go in the same order as in the [value] character sequence, starting at [index].\n * @param index the position in this string builder to insert at.\n * @param value the character sequence from which a subsequence is inserted.\n * @param startIndex the beginning (inclusive) of the subsequence to insert.\n * @param endIndex the end (exclusive) of the subsequence to insert.\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of the [value] character sequence indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.\n

*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
StringBuilder.insertRange(index: Int, value: CharSequence, startIndex: Int, endIndex: Int):
StringBuilder\n\n@Suppress("EXTENSION_SHADOWED_BY_MEMBER")\n@Deprecated("Use append(value: Any?) instead", ReplaceWith("append(value = obj)"),
DeprecationLevel.WARNING)\n@kotlin.internal.InlineOnly\npublic inline fun StringBuilder.append(obj: Any?):
StringBuilder = this.append(obj)\n\n/**\n * Builds new string by populating newly created [StringBuilder] using provided [builderAction]\n * and then converting it to [String].\n * @kotlin.internal.InlineOnly\npublic inline fun
buildString(builderAction: StringBuilder.() -> Unit): String {\n contract { callsInPlace(builderAction, InvocationKind.EXACTLY_ONCE) }\n return StringBuilder().apply(builderAction).toString()\n}\n\n/**\n * Builds new string by populating newly created [StringBuilder] initialized with the given [capacity]\n * using provided [builderAction] and then converting it to [String].\n

*\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline fun buildString(capacity: Int, builderAction: StringBuilder.() -> Unit): String {\n contract { callsInPlace(builderAction, InvocationKind.EXACTLY_ONCE) }\n return StringBuilder(capacity).apply(builderAction).toString()\n}\n\n/**\n * Appends all arguments to the given String Builder.\n * @public fun StringBuilder.append(vararg value: String?): StringBuilder {\n for (item in

```

value)\n    append(item)\n    return this\n}\n\n/**\n * Appends all arguments to the given StringBuilder.\n *\npublic fun StringBuilder.append(vararg value: Any?): StringBuilder {\n    for (item in value)\n        append(item)\n    return this\n}\n\n/** Appends a line feed character (`\n`) to this StringBuilder.\n *\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun StringBuilder.appendLine():\n    StringBuilder = append("\n")\n\n/** Appends [value] to this [StringBuilder], followed by a line feed character\n    (`\n`). *\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun StringBuilder.appendLine(value:\n    CharSequence?): StringBuilder = append(value).appendLine()\n\n/** Appends [value] to this [StringBuilder],\n    followed by a line feed character (`\n`). *\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun\n    StringBuilder.appendLine(value: String?): StringBuilder = append(value).appendLine()\n\n/** Appends [value] to\n    this [StringBuilder], followed by a line feed character (`\n`).\n *\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun StringBuilder.appendLine(value: Any?):\n    StringBuilder = append(value).appendLine()\n\n/** Appends [value] to this [StringBuilder], followed by a line feed\n    character (`\n`). *\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun\n    StringBuilder.appendLine(value: CharArray): StringBuilder = append(value).appendLine()\n\n/** Appends [value]\n    to this [StringBuilder], followed by a line feed character (`\n`).\n *\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun StringBuilder.appendLine(value: Char):\n    StringBuilder = append(value).appendLine()\n\n/** Appends [value] to this [StringBuilder], followed by a line feed\n    character (`\n`). *\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun\n    StringBuilder.appendLine(value: Boolean): StringBuilder = append(value).appendLine()\n\n"/*\n * Copyright 2010-\n    2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n    *\n    Use of this source code is governed by the\n    Apache 2.0 license that can be found in the license/LICENSE.txt file.\n *\npackage kotlin.text\n\nimport\n    kotlin.js.RegExp\n\n@kotlin.internal.InlineOnly\ninternal actual inline fun String.nativeIndexOf(ch: Char,\n    fromIndex: Int): Int = nativeIndexOf(ch.toString(), fromIndex)\n\n@kotlin.internal.InlineOnly\ninternal actual\n    inline fun String.nativeLastIndexOf(ch: Char, fromIndex: Int): Int = nativeLastIndexOf(ch.toString(),\n    fromIndex)\n\n/**\n * Returns `true` if this string starts with the specified prefix.\n *\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual fun\n    String.startsWith(prefix: String, ignoreCase: Boolean = false): Boolean {\n        if (!ignoreCase)\n            return\n                nativeStartsWith(prefix, 0)\n        else\n            return regionMatches(0, prefix, 0, prefix.length, ignoreCase)\n    }\n\n/**\n * Returns `true` if a substring of this string starting at the specified offset [startIndex] starts with the specified prefix.\n *\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual fun\n    String.startsWith(prefix: String, startIndex: Int, ignoreCase: Boolean = false): Boolean {\n        if (!ignoreCase)\n            return\n                nativeStartsWith(prefix, startIndex)\n        else\n            return regionMatches(startIndex, prefix, 0, prefix.length,\n                ignoreCase)\n    }\n\n/**\n * Returns `true` if this string ends with the specified suffix.\n *\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual fun\n    String.endsWith(suffix: String, ignoreCase: Boolean = false): Boolean {\n        if (!ignoreCase)\n            return\n                nativeEndsWith(suffix)\n        else\n            return regionMatches(length - suffix.length, suffix, 0, suffix.length,\n                ignoreCase)\n    }\n\n@Deprecated("Use Regex.matches() instead",\n    ReplaceWith("regex.toRegex().matches(this)"))\n@DeprecatedSinceKotlin(warningSince = "1.6")\npublic fun\n    String.matches(regex: String): Boolean {\n        @Suppress("DEPRECATION")\n        val result = this.match(regex)\n        return result != null && result.size != 0\n    }\n\n/**\n * Returns `true` if this string is empty or consists solely of\n    whitespace characters.\n *\n * @sample samples.text.Strings.stringIsBlank\n *\npublic actual fun\n    CharSequence.isBlank(): Boolean = length == 0 || indices.all { this[it].isWhitespace() }\n\n/**\n * Returns `true` if\n    this string is equal to [other], optionally ignoring character case.\n *\n * Two strings are considered to be equal if\n    they have the same length and the same character at the same index.\n *\n * If [ignoreCase] is true, the result of\n    `Char.toUpperCaseChar().toLowerCaseChar()` on each character is compared.\n *\n * @param ignoreCase `true` to ignore\n    character case when comparing strings. By default `false`.\n *\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual fun\n    String?.equals(other: String?, ignoreCase: Boolean = false): Boolean {\n        if (this == null) return other == null\n        if

```

```

(other == null) return false\n  if (!ignoreCase) return this == other\n\n  if (this.length != other.length) return
false\n\n  for (index in 0 until this.length) {\n    val thisChar = this[index]\n    val otherChar = other[index]\n
    if (!thisChar.equals(otherChar, ignoreCase)) {\n      return false\n    }\n  }\n  return
true\n}\n\n\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual fun
CharSequence.regionMatches(thisOffset: Int, other: CharSequence, otherOffset: Int, length: Int, ignoreCase:
Boolean = false): Boolean =\n  regionMatchesImpl(thisOffset, other, otherOffset, length, ignoreCase)\n\n\n/**\n *
Returns a copy of this string having its first letter titlecased using the rules of the default locale,\n * or the original
string if it's empty or already starts with a title case letter.\n * \n * The title case of a character is usually the same as
its upper case with several exceptions.\n * The particular list of characters with the special title case form depends
on the underlying platform.\n * \n * @sample samples.text.Strings.capitalize\n * \n * @Deprecated("Use
replaceFirstChar instead.", ReplaceWith("replaceFirstChar { if (it.isLowerCase()) it.titlecase() else it.toString()
}"))\n * \n * @DeprecatedSinceKotlin(warningSince = "1.5")\n * \n * public actual fun String.capitalize(): String {\n
return if (isEmpty()) substring(0, 1).uppercase() + substring(1) else this\n}\n\n\n/**\n * Returns a copy of this string having
its first letter lowercased using the rules of the default locale,\n * or the original string if it's empty or already starts
with a lower case letter.\n * \n * @sample samples.text.Strings.decapitalize\n * \n * @Deprecated("Use
replaceFirstChar instead.", ReplaceWith("replaceFirstChar { it.lowercase()
}"))\n * \n * @DeprecatedSinceKotlin(warningSince = "1.5")\n * \n * public actual fun String.decapitalize(): String {\n
return if (isEmpty()) substring(0, 1).lowercase() + substring(1) else this\n}\n\n\n/**\n * Returns a string containing this
char sequence repeated [n] times.\n * \n * @throws [IllegalArgumentException] when n < 0.\n * \n * @sample
samples.text.Strings.repeat\n * \n * public actual fun CharSequence.repeat(n: Int): String {\n  require(n >= 0) {\n
"Count 'n' must be non-negative, but was $n." }\n  return when (n) {\n    0 -> ""\n    1 -> this.toString()\n
else -> {\n      var result = ""\n      if (!isEmpty()) {\n        var s = this.toString()\n        var count =
n\n        while (true) {\n          if ((count and 1) == 1) {\n            result += s\n          }\n          count =
count ushr 1\n          if (count == 0) {\n            break\n          }\n          s +=
s\n        }\n      }\n      return result\n    }\n  }\n}\n\n\n/**\n * Returns a new string obtained by
replacing all occurrences of the [oldValue] substring in this string\n * with the specified [newValue] string.\n * \n *
@sample samples.text.Strings.replace\n * \n * @Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\n * \n * public actual fun
String.replace(oldValue: String, newValue: String, ignoreCase: Boolean = false): String =\n  nativeReplace(Regex(Regex.escape(oldValue), if (ignoreCase) "gui" else "gu"),
Regex.nativeEscapeReplacement(newValue))\n\n\n/**\n * Returns a new string with all occurrences of [oldChar]
replaced with [newChar].\n * \n * @sample samples.text.Strings.replace\n * \n * @Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\n * \n * public actual fun
String.replace(oldChar: Char, newChar: Char, ignoreCase: Boolean = false): String =\n  nativeReplace(Regex(Regex.escape(oldChar.toString()), if (ignoreCase) "gui" else "gu"),
newChar.toString())\n\n\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\npublic actual
fun String.replaceFirst(oldValue: String, newValue: String, ignoreCase: Boolean = false): String =\n  nativeReplace(Regex(Regex.escape(oldValue), if (ignoreCase) "ui" else "u"),
Regex.nativeEscapeReplacement(newValue))\n\n\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGU
MENTS")\npublic actual fun String.replaceFirst(oldChar: Char, newChar: Char, ignoreCase: Boolean = false):
String =\n  nativeReplace(Regex(Regex.escape(oldChar.toString()), if (ignoreCase) "ui" else "u"),
newChar.toString())\n", "/*\n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n * \n * \n * package kotlin.text\n * \n * /** Returns the negative [size] if [throwOnMalformed] is
false, throws [CharacterCodingException] otherwise. *\n * private fun malformed(size: Int, index: Int,
throwOnMalformed: Boolean): Int {\n  if (throwOnMalformed) throw CharacterCodingException("Malformed
sequence starting at ${index - 1}")\n  return -size\n}\n\n\n/**\n * Returns code point corresponding to UTF-16
surrogate pair,\n * where the first of the pair is the [high] and the second is in the [string] at the [index].\n * Returns

```

```

zero if the pair is malformed and [throwOnMalformed] is false.\n *\n * @throws CharacterCodingException if the
pair is malformed and [throwOnMalformed] is true.\n *\nprivate fun codePointFromSurrogate(string: String, high:
Int, index: Int, endIndex: Int, throwOnMalformed: Boolean): Int {\n    if (high !in 0xD800..0xDBFF || index >=
endIndex) {\n        return malformed(0, index, throwOnMalformed)\n    }\n    val low = string[index].code\n    if
(low !in 0xDC00..0xDFFF) {\n        return malformed(0, index, throwOnMalformed)\n    }\n    return 0x10000 +
((high and 0x3FF) shl 10) or (low and 0x3FF)\n}\n\n/**\n * Returns code point corresponding to UTF-8 sequence of
two bytes,\n * where the first byte of the sequence is the [byte1] and the second byte is in the [bytes] array at the
[index].\n * Returns zero if the sequence is malformed and [throwOnMalformed] is false.\n *\n * @throws
CharacterCodingException if the sequence of two bytes is malformed and [throwOnMalformed] is true.\n
*\nprivate fun codePointFrom2(bytes: ByteArray, byte1: Int, index: Int, endIndex: Int, throwOnMalformed:
Boolean): Int {\n    if (byte1 and 0x1E == 0 || index >= endIndex) {\n        return malformed(0, index,
throwOnMalformed)\n    }\n    val byte2 = bytes[index].toInt()\n    if (byte2 and 0xC0 != 0x80) {\n        return
malformed(0, index, throwOnMalformed)\n    }\n    return (byte1 shl 6) xor byte2 xor 0xF80\n}\n\n/**\n * Returns
code point corresponding to UTF-8 sequence of three bytes,\n * where the first byte of the sequence is the [byte1]
and the others are in the [bytes] array starting from the [index].\n * Returns a non-positive value indicating number
of bytes from [bytes] included in malformed sequence\n * if the sequence is malformed and [throwOnMalformed] is
false.\n *\n * @throws CharacterCodingException if the sequence of three bytes is malformed and
[throwOnMalformed] is true.\n *\nprivate fun codePointFrom3(bytes: ByteArray, byte1: Int, index: Int, endIndex:
Int, throwOnMalformed: Boolean): Int {\n    if (index >= endIndex) {\n        return malformed(0, index,
throwOnMalformed)\n    }\n    val byte2 = bytes[index].toInt()\n    if (byte1 and 0xF == 0) {\n        if (byte2 and
0xE0 != 0xA0) {\n            // Non-shortest form\n            return malformed(0, index, throwOnMalformed)\n        }\n    }
else if (byte1 and 0xF == 0xD) {\n        if (byte2 and 0xE0 != 0x80) {\n            // Surrogate code point\n
return malformed(0, index, throwOnMalformed)\n        }\n    } else if (byte2 and 0xC0 != 0x80) {\n        return
malformed(0, index, throwOnMalformed)\n    }\n    if (index + 1 == endIndex) {\n        return malformed(1, index,
throwOnMalformed)\n    }\n    val byte3 = bytes[index + 1].toInt()\n    if (byte3 and 0xC0 != 0x80) {\n        return
malformed(1, index, throwOnMalformed)\n    }\n    return (byte1 shl 12) xor (byte2 shl 6) xor byte3 xor -
0x1E080\n}\n\n/**\n * Returns code point corresponding to UTF-8 sequence of four bytes,\n * where the first byte
of the sequence is the [byte1] and the others are in the [bytes] array starting from the [index].\n * Returns a non-
positive value indicating number of bytes from [bytes] included in malformed sequence\n * if the sequence is
malformed and [throwOnMalformed] is false.\n *\n * @throws CharacterCodingException if the sequence of four
bytes is malformed and [throwOnMalformed] is true.\n *\nprivate fun codePointFrom4(bytes: ByteArray, byte1:
Int, index: Int, endIndex: Int, throwOnMalformed: Boolean): Int {\n    if (index >= endIndex) {\n        malformed(0,
index, throwOnMalformed)\n    }\n    val byte2 = bytes[index].toInt()\n    if (byte1 and 0xF == 0x0) {\n        if
(byte2 and 0xF0 <= 0x80) {\n            // Non-shortest form\n            return malformed(0, index,
throwOnMalformed)\n        }\n    } else if (byte1 and 0xF == 0x4) {\n        if (byte2 and 0xF0 != 0x80) {\n            //
Out of Unicode code points domain (larger than U+10FFFF)\n            return malformed(0, index,
throwOnMalformed)\n        }\n    } else if (byte1 and 0xF > 0x4) {\n        return malformed(0, index,
throwOnMalformed)\n    } else if (byte2 and 0xC0 != 0x80) {\n        return malformed(0, index,
throwOnMalformed)\n    }\n    if (index + 1 == endIndex) {\n        return malformed(1, index,
throwOnMalformed)\n    }\n    val byte3 = bytes[index + 1].toInt()\n    if (byte3 and 0xC0 != 0x80) {\n        return
malformed(1, index, throwOnMalformed)\n    }\n    if (index + 2 == endIndex) {\n        return malformed(2, index,
throwOnMalformed)\n    }\n    val byte4 = bytes[index + 2].toInt()\n    if (byte4 and 0xC0 != 0x80) {\n        return
malformed(2, index, throwOnMalformed)\n    }\n    return (byte1 shl 18) xor (byte2 shl 12) xor (byte3 shl 6) xor
byte4 xor 0x381F80\n}\n\n/**\n * Maximum number of bytes needed to encode a single char.\n *\n * Code points in
`0..0x7F` are encoded in a single byte.\n * Code points in `0x80..0x7FF` are encoded in two bytes.\n * Code points
in `0x800..0xD7FF` or in `0xE000..0xFFFF` are encoded in three bytes.\n * Surrogate code points in
`0xD800..0xDFFF` are not Unicode scalar values, therefore aren't encoded.\n * Code points in
`0x10000..0x10FFFF` are represented by a pair of surrogate `Char`s and are encoded in four bytes.\n *\nprivate

```

```

const val MAX_BYTES_PER_CHAR = 3\n\n/**\n * The byte sequence a malformed UTF-16 char sequence is
replaced by.\n */\nprivate val REPLACEMENT_BYTE_SEQUENCE: ByteArray = byteArrayOf(0xEF.toByte(),
0xBF.toByte(), 0xBD.toByte())\n\n/**\n * Encodes the [string] using UTF-8 and returns the resulting [ByteArray].\n
*\n * @param string the string to encode.\n * @param startIndex the start offset (inclusive) of the substring to
encode.\n * @param endIndex the end offset (exclusive) of the substring to encode.\n * @param
throwOnMalformed whether to throw on malformed char sequence or replace by the
[REPLACEMENT_BYTE_SEQUENCE].\n */\n * @throws CharacterCodingException if the char sequence is
malformed and [throwOnMalformed] is true.\n */\ninternal fun encodeUtf8(string: String, startIndex: Int, endIndex:
Int, throwOnMalformed: Boolean): ByteArray {\n    require(startIndex >= 0 && endIndex <= string.length &&
startIndex <= endIndex)\n    val bytes = ByteArray((endIndex - startIndex) * MAX_BYTES_PER_CHAR)\n    var
byteIndex = 0\n    var charIndex = startIndex\n    while (charIndex < endIndex) {\n        val code =
string[charIndex++].code\n        when {\n            code < 0x80 -> {\n                bytes[byteIndex++] = code.toByte()\n
            }\n            code < 0x800 -> {\n                bytes[byteIndex++] = ((code shr 6) or 0xC0).toByte()\n
            }\n            code < 0xD800 || code >= 0xE000 -> {\n                bytes[byteIndex++] = ((code and 0x3F) or 0x80).toByte()\n
            }\n            code < 0xD800 || code >= 0xE000 -> {\n                bytes[byteIndex++] = ((code shr 12) or 0xE0).toByte()\n
            }\n            code < 0xD800 || code >= 0xE000 -> {\n                bytes[byteIndex++] = (((code shr 6)
and 0x3F) or 0x80).toByte()\n            }\n            code < 0xD800 || code >= 0xE000 -> {\n                bytes[byteIndex++] = ((code and 0x3F) or 0x80).toByte()\n
            }\n            else -> { // Surrogate char value\n                val codePoint = codePointFromSurrogate(string, code, charIndex,
endIndex, throwOnMalformed)\n                if (codePoint <= 0) {\n                    bytes[byteIndex++] =
REPLACEMENT_BYTE_SEQUENCE[0]\n                }\n                if (codePoint <= 0) {\n                    bytes[byteIndex++] =
REPLACEMENT_BYTE_SEQUENCE[1]\n                }\n                if (codePoint <= 0) {\n                    bytes[byteIndex++] =
REPLACEMENT_BYTE_SEQUENCE[2]\n                }\n                } else {\n                    bytes[byteIndex++] = ((codePoint shr
18) or 0xF0).toByte()\n                }\n                bytes[byteIndex++] = (((codePoint shr 12) and 0x3F) or 0x80).toByte()\n
            }\n                bytes[byteIndex++] = (((codePoint shr 6) and 0x3F) or 0x80).toByte()\n                bytes[byteIndex++] =
(((codePoint and 0x3F) or 0x80).toByte()\n                charIndex++\n                }\n            }\n            }\n            }\n            }\n        }\n    }\n    return if (bytes.size == byteIndex) bytes else bytes.copyOf(byteIndex)\n}\n\n/**\n * The character a malformed
UTF-8 byte sequence is replaced by.\n */\nprivate const val REPLACEMENT_CHAR = "\uFFFF"\n\n/**\n * Decodes the UTF-8 [bytes] array and returns the resulting [String].\n
*\n * @param bytes the byte array to decode.\n * @param startIndex the start offset (inclusive) of the array to be decoded.\n * @param endIndex the end offset
(exclusive) of the array to be encoded.\n * @param throwOnMalformed whether to throw on malformed byte
sequence or replace by the [REPLACEMENT_CHAR].\n */\n * @throws CharacterCodingException if the array is
malformed UTF-8 byte sequence and [throwOnMalformed] is true.\n */\ninternal fun decodeUtf8(bytes: ByteArray,
startIndex: Int, endIndex: Int, throwOnMalformed: Boolean): String {\n    require(startIndex >= 0 && endIndex <=
bytes.size && startIndex <= endIndex)\n    var byteIndex = startIndex\n    val stringBuilder = StringBuilder()\n    while (byteIndex < endIndex) {\n        val byte = bytes[byteIndex++].toInt()\n        when {\n            byte >= 0 -> {\n
                stringBuilder.append(byte.toChar())\n            }\n            byte shr 5 == -2 -> {\n                val code =
codePointFrom2(bytes, byte, byteIndex, endIndex, throwOnMalformed)\n                if (code <= 0) {\n
                    stringBuilder.append(REPLACEMENT_CHAR)\n                }\n                byteIndex += -code\n            }\n            } else {\n
                stringBuilder.append(code.toChar())\n                byteIndex += 1\n            }\n            }\n            byte shr 4 == -2 -
> {\n                val code = codePointFrom3(bytes, byte, byteIndex, endIndex, throwOnMalformed)\n                if
(code <= 0) {\n                    stringBuilder.append(REPLACEMENT_CHAR)\n                }\n                byteIndex += -code\n
            }\n            } else {\n                stringBuilder.append(code.toChar())\n                byteIndex += 2\n            }\n
        }\n    }\n    byte shr 3 == -2 -> {\n        val code = codePointFrom4(bytes, byte, byteIndex, endIndex,
throwOnMalformed)\n        if (code <= 0) {\n            stringBuilder.append(REPLACEMENT_CHAR)\n        }\n        byteIndex += -code\n    }\n    } else {\n        val high = (code - 0x10000) shr 10 or 0xD800\n
        val low = (code and 0x3FF) or 0xDC00\n        stringBuilder.append(high.toChar())\n        stringBuilder.append(low.toChar())\n        byteIndex += 3\n    }\n    }\n    }\n    }\n    return stringBuilder.toString()\n}\n\n"/**\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin

```

```

Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n */\n\npackage kotlin\n\n/**\n * Returns the detailed description of this
throwable with its stack trace.\n */\n * The detailed description includes:\n * - the short description (see
[Throwable.toString]) of this throwable;\n * - the complete stack trace;\n * - detailed descriptions of the exceptions
that were [suppressed][suppressedExceptions] in order to deliver this exception;\n * - the detailed description of
each throwable in the [Throwable.cause] chain.\n */\n\n@SinceKotlin("1.4")\npublic actual fun
Throwable.stackTraceToString(): String = ExceptionTraceBuilder().buildFor(this)\n\n/**\n * Prints the [detailed
description][Throwable.stackTraceToString] of this throwable to console error output.\n
*/\n\n@SinceKotlin("1.4")\npublic actual fun Throwable.printStackTrace() {\n
console.error(this.stackTraceToString())\n}\n\n/**\n * Adds the specified exception to the list of exceptions that
were\n * suppressed in order to deliver this exception.\n */\n\n@SinceKotlin("1.4")\npublic actual fun
Throwable.addSuppressed(exception: Throwable) {\n    if (this !== exception) {\n        val suppressed =
this.asDynamic()._suppressed.unsafeCast<MutableList<Throwable>?>()\n        if (suppressed == null) {\n
this.asDynamic()._suppressed = mutableListOf(exception)\n        } else {\n            suppressed.add(exception)\n
}\n    }\n}\n\n/**\n * Returns a list of all exceptions that were suppressed in order to deliver this exception.\n
*/\n\n@SinceKotlin("1.4")\npublic actual val Throwable.suppressedExceptions: List<Throwable>\n    get() {\n
return this.asDynamic()._suppressed?.unsafeCast<List<Throwable>>() ?: emptyList()\n    }\n}\n\nprivate class
ExceptionTraceBuilder {\n    private val target = StringBuilder()\n    private val visited = arrayOf<Throwable>()\n
private var topStack: String = ""\n    private var topStackStart: Int = 0\n    fun buildFor(exception: Throwable):
String {\n        exception.dumpFullTrace("", "")\n        return target.toString()\n    }\n\n    private fun
hasSeen(exception: Throwable): Boolean = visited.any { it === exception }\n\n    private fun
Throwable.dumpFullTrace(indent: String, qualifier: String) {\n        this.dumpSelfTrace(indent, qualifier) ||
return\n\n        var cause = this.cause\n        while (cause != null) {\n            cause.dumpSelfTrace(indent, "Caused
by: ") || return\n            cause = cause.cause\n        }\n    }\n\n    private fun Throwable.dumpSelfTrace(indent:
String, qualifier: String): Boolean {\n        target.append(indent).append(qualifier)\n        val shortInfo =
this.toString()\n        if (hasSeen(this)) {\n            target.append("[CIRCULAR REFERENCE, SEE ABOVE:
").append(shortInfo).append("]\n\n")\n            return false\n        }\n        visited.asDynamic().push(this)\n        var
stack = this.asDynamic().stack as String?\n        if (stack != null) {\n            val stackStart =
stack.indexOf(shortInfo).let { if (it < 0) 0 else it + shortInfo.length }\n            if (stackStart == 0)\n                target.append(shortInfo).append("\n\n")\n            if (topStack.isEmpty()) {\n                topStack = stack\n
topStackStart = stackStart\n            } else {\n                stack = dropCommonFrames(stack, stackStart)\n            }\n
            if (indent.isNotEmpty()) {\n                // indent stack, but avoid indenting exception message lines\n                val
messageLines = if (stackStart == 0) 0 else 1 + shortInfo.count { c -> c == '\n' }\n                stack.lineSequence().forEachIndexed { index: Int, line: String -> }\n                if (index >= messageLines)\n                    target.append(indent)\n                    target.append(line).append("\n\n")\n                } else {\n                    target.append(stack).append("\n\n")\n                }\n            } else {\n                target.append(shortInfo).append("\n\n")\n            }\n        }\n        val suppressed = suppressedExceptions\n        if (suppressed.isNotEmpty()) {\n            val
suppressedIndent = indent + "\n    " for (s in suppressed) {\n                s.dumpFullTrace(suppressedIndent,
"Suppressed: ")\n            }\n        }\n        return true\n    }\n\n    private fun dropCommonFrames(stack: String,
stackStart: Int): String {\n        var commonFrames: Int = 0\n        var lastBreak: Int = 0\n        var preLastBreak: Int
= 0\n        for (pos in 0 until minOf(topStack.length - topStackStart, stack.length - stackStart)) {\n            val c =
stack[stack.lastIndex - pos]\n            if (c != topStack[topStack.lastIndex - pos]) break\n            if (c == '\n') {\n
                commonFrames += 1\n                preLastBreak = lastBreak\n                lastBreak = pos\n            }\n        }\n
if (commonFrames <= 1) return stack\n        while (preLastBreak > 0 && stack[stack.lastIndex - (preLastBreak - 1)]
== ' ')\n            preLastBreak -= 1\n        // leave 1 common frame to ease matching with the top exception stack\n
return stack.dropLast(preLastBreak) + "... and ${commonFrames - 1} more common stack frames skipped"\n    }\n}\n\n"/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this
source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n

```

```

*/\n\npackage kotlin.time\n\nimport kotlin.js.json\nimport kotlin.math.*\n\ninternal actual inline val
durationAssertionsEnabled: Boolean get() = true\n\ninternal actual fun formatToExactDecimals(value: Double,
decimals: Int): String {\n    val rounded = if (decimals == 0) {\n        value\n    } else {\n        val pow =
10.0.pow(decimals)\n        JsMath.round(abs(value) * pow) / pow * sign(value)\n    }\n    return if (abs(rounded) <
1e21) {\n        // toFixed switches to scientific format after 1e21\n
rounded.asDynamic().toFixed(decimals).unsafeCast<String>()\n    } else {\n        // toPrecision outputs the specified
number of digits, but only for positive numbers\n        val positive = abs(rounded)\n        val positiveString =
positive.asDynamic().toPrecision(ceil(log10(positive)) + decimals).unsafeCast<String>()\n        if (rounded < 0) \"-
$positiveString\" else positiveString\n    }\n}\n\ninternal actual fun formatUpToDecimals(value: Double, decimals:
Int): String {\n    return value.asDynamic().toLocaleString(\"en-us\", json(\"maximumFractionDigits\" to
decimals)).unsafeCast<String>()\n}\n\n\"/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\npackage
kotlin.time\n\n@SinceKotlin(\"1.6\")\n@WasExperimental(ExperimentalTime::class)\npublic actual enum class
DurationUnit(internal val scale: Double) {\n    /**\n     * Time unit representing one nanosecond, which is 1/1000 of
a microsecond.\n     */\n    NANOSECONDS(1e0),\n    /**\n     * Time unit representing one microsecond, which is
1/1000 of a millisecond.\n     */\n    MICROSECONDS(1e3),\n    /**\n     * Time unit representing one millisecond,
which is 1/1000 of a second.\n     */\n    MILLISECONDS(1e6),\n    /**\n     * Time unit representing one second.\n
*/\n    SECONDS(1e9),\n    /**\n     * Time unit representing one minute.\n     */\n    MINUTES(60e9),\n    /**\n
     * Time unit representing one hour.\n     */\n    HOURS(3600e9),\n    /**\n     * Time unit representing one day,
which is always equal to 24 hours.\n     */\n    DAYS(86400e9);\n}\n\n@SinceKotlin(\"1.3\")\ninternal actual fun
convertDurationUnit(value: Double, sourceUnit: DurationUnit, targetUnit: DurationUnit): Double {\n    val
sourceCompareTarget = sourceUnit.scale.compareTo(targetUnit.scale)\n    return when {\n
sourceCompareTarget > 0 -> value * (sourceUnit.scale / targetUnit.scale)\n        sourceCompareTarget < 0 -> value /
(targetUnit.scale / sourceUnit.scale)\n        else -> value\n    }\n}\n\n@SinceKotlin(\"1.5\")\ninternal actual fun
convertDurationUnitOverflow(value: Long, sourceUnit: DurationUnit, targetUnit: DurationUnit): Long {\n    val
sourceCompareTarget = sourceUnit.scale.compareTo(targetUnit.scale)\n    return when {\n
sourceCompareTarget > 0 -> value * (sourceUnit.scale / targetUnit.scale).toLong()\n        sourceCompareTarget < 0
-> value / (targetUnit.scale / sourceUnit.scale).toLong()\n        else -> value\n    }\n}\n\n@SinceKotlin(\"1.5\")\ninternal actual fun convertDurationUnit(value: Long, sourceUnit: DurationUnit,
targetUnit: DurationUnit): Long {\n    val sourceCompareTarget = sourceUnit.scale.compareTo(targetUnit.scale)\n
return when {\n        sourceCompareTarget > 0 -> {\n            val scale = (sourceUnit.scale /
targetUnit.scale).toLong()\n            val result = value * scale\n            when {\n                result / scale == value ->
result\n                value > 0 -> Long.MAX_VALUE\n                else -> Long.MIN_VALUE\n            }\n        }\n
sourceCompareTarget < 0 -> value / (targetUnit.scale / sourceUnit.scale).toLong()\n        else -> value\n    }\n}\n\n\"/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of
this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.time\n\nimport org.w3c.performance.GlobalPerformance\nimport
org.w3c.performance.Performance\n\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\ninternal actual object
MonotonicTimeSource : TimeSource {\n\n    private val actualSource: TimeSource = run {\n        val isNode:
Boolean = js(\"typeof process !== 'undefined' && process.versions && !!process.versions.node\")\n        if
(isNode)\n            HrTimeSource(js(\"process\").unsafeCast<Process>())\n        else\n            js(\"self\").unsafeCast<GlobalPerformance?>()?.performance?.let(::PerformanceTimeSource)\n    }\n\n    ?:
DateNowTimeSource\n}\n\n    override fun markNow(): TimeMark = actualSource.markNow()\n}\n\ninternal
external interface Process {\n    fun hrtime(time: Array<Double> = definedExternally):
Array<Double>\n}\n\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\ninternal class HrTimeSource(val process:
Process) : TimeSource {\n    override fun markNow(): TimeMark = object : TimeMark() {\n        val startedAt =
process.hrtime()\n        override fun elapsedNow(): Duration =\n            process.hrtime(startedAt).let { (seconds,

```



```

whether this node is an [Element].\n *\n@SinceKotlin("1.4")\npublic val Node.isElement: Boolean\n    get() =
nodeType == Node.ELEMENT_NODE\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n *\n\npackage kotlinx.dom\n\nimport org.w3c.dom.*\n\n/** Removes all the children
from this node. *\n@SinceKotlin("1.4")\npublic fun Node.clear() {\n    while (hasChildNodes()) {\n
removeChild(firstChild!!)\n    }\n}\n\n/**\n * Creates text node and append it to the element.\n *\n * @return this
element\n *\n@SinceKotlin("1.4")\nfun Element.appendText(text: String): Element {\n
appendChild(ownerDocument!!.createTextNode(text))\n    return this\n}\n","/*\n * Copyright 2010-2019 JetBrains
s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0
license that can be found in the license/LICENSE.txt file.\n *\n\npackage org.w3c.dom\n\n@Deprecated("Use
UnionMessagePortOrWindowProxy instead.", ReplaceWith("UnionMessagePortOrWindowProxy"))\ntypealias
UnionMessagePortOrWindow = UnionMessagePortOrWindowProxy\n\n@Deprecated("Use `as` instead.",
ReplaceWith("`as`"))\nvar HTMLLinkElement.as_\n    get() = `as`\n    set(value) {\n        `as` = value\n
}\n\n@Deprecated("Use `is` instead.", ReplaceWith("`is`"))\nvar ElementCreationOptions.is_\n    get() = `is`\n
set(value) {\n        `is` = value\n    }"/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n *\n\n// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!\n// See
github.com/kotlin/dukat for details\n\npackage org.khronos.webgl\n\nimport kotlin.js.*\nimport
org.w3c.dom.*\nimport org.w3c.dom.events.*\n\npublic external interface WebGLContextAttributes {\n    var
alpha: Boolean? /* = true */\n        get() = definedExternally\n        set(value) = definedExternally\n    var depth:
Boolean? /* = true */\n        get() = definedExternally\n        set(value) = definedExternally\n    var stencil: Boolean?
/* = false */\n        get() = definedExternally\n        set(value) = definedExternally\n    var antialias: Boolean? /* =
true */\n        get() = definedExternally\n        set(value) = definedExternally\n    var premultipliedAlpha: Boolean?
/* = true */\n        get() = definedExternally\n        set(value) = definedExternally\n    var preserveDrawingBuffer:
Boolean? /* = false */\n        get() = definedExternally\n        set(value) = definedExternally\n    var
preferLowPowerToHighPerformance: Boolean? /* = false */\n        get() = definedExternally\n        set(value) =
definedExternally\n    var failIfMajorPerformanceCaveat: Boolean? /* = false */\n        get() = definedExternally\n
set(value) = definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n\n@kotlin.internal.InlineOnly\npublic inline fun WebGLContextAttributes(alpha:
Boolean? = true, depth: Boolean? = true, stencil: Boolean? = false, antialias: Boolean? = true, premultipliedAlpha:
Boolean? = true, preserveDrawingBuffer: Boolean? = false, preferLowPowerToHighPerformance: Boolean? = false,
failIfMajorPerformanceCaveat: Boolean? = false): WebGLContextAttributes {\n    val o = js("{}")\n
o["alpha"] = alpha\n    o["depth"] = depth\n    o["stencil"] = stencil\n    o["antialias"] = antialias\n
o["premultipliedAlpha"] = premultipliedAlpha\n    o["preserveDrawingBuffer"] = preserveDrawingBuffer\n
o["preferLowPowerToHighPerformance"] = preferLowPowerToHighPerformance\n
o["failIfMajorPerformanceCaveat"] = failIfMajorPerformanceCaveat\n    return o\n}\n\npublic external abstract
class WebGLObject\n\n/**\n * Exposes the JavaScript
[WebGLBuffer](https://developer.mozilla.org/en/docs/Web/API/WebGLBuffer) to Kotlin\n *\n\npublic external
abstract class WebGLBuffer : WebGLObject\n\n/**\n * Exposes the JavaScript
[WebGLFramebuffer](https://developer.mozilla.org/en/docs/Web/API/WebGLFramebuffer) to Kotlin\n *\n\npublic
external abstract class WebGLFramebuffer : WebGLObject\n\n/**\n * Exposes the JavaScript
[WebGLProgram](https://developer.mozilla.org/en/docs/Web/API/WebGLProgram) to Kotlin\n *\n\npublic external
abstract class WebGLProgram : WebGLObject\n\n/**\n * Exposes the JavaScript
[WebGLRenderbuffer](https://developer.mozilla.org/en/docs/Web/API/WebGLRenderbuffer) to Kotlin\n *\n\npublic
external abstract class WebGLRenderbuffer : WebGLObject\n\n/**\n * Exposes the JavaScript
[WebGLShader](https://developer.mozilla.org/en/docs/Web/API/WebGLShader) to Kotlin\n *\n\npublic external
abstract class WebGLShader : WebGLObject\n\n/**\n * Exposes the JavaScript
[WebGLTexture](https://developer.mozilla.org/en/docs/Web/API/WebGLTexture) to Kotlin\n *\n\npublic external

```

```

abstract class WebGLTexture : WebGLObject\n\n**\n * Exposes the JavaScript
[WebGLUniformLocation](https://developer.mozilla.org/en/docs/Web/API/WebGLUniformLocation) to Kotlin\n
*\npublic external abstract class WebGLUniformLocation\n\n**\n * Exposes the JavaScript
[WebGLActiveInfo](https://developer.mozilla.org/en/docs/Web/API/WebGLActiveInfo) to Kotlin\n *\npublic
external abstract class WebGLActiveInfo {\n  open val size: Int\n  open val type: Int\n  open val name:
String\n}\n\n**\n * Exposes the JavaScript
[WebGLShaderPrecisionFormat](https://developer.mozilla.org/en/docs/Web/API/WebGLShaderPrecisionFormat) to
Kotlin\n *\npublic external abstract class WebGLShaderPrecisionFormat {\n  open val rangeMin: Int\n  open val
rangeMax: Int\n  open val precision:
Int\n}\n\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\npublic external interface
WebGLRenderingContextBase {\n  val canvas: HTMLCanvasElement\n  val drawingBufferWidth: Int\n  val
drawingBufferHeight: Int\n  fun getContextAttributes(): WebGLContextAttributes?\n  fun isContextLost():
Boolean\n  fun getSupportedExtensions(): Array<String>?\n  fun getExtension(name: String): dynamic\n  fun
activeTexture(texture: Int)\n  fun attachShader(program: WebGLProgram?, shader: WebGLShader?)\n  fun
bindAttribLocation(program: WebGLProgram?, index: Int, name: String)\n  fun bindBuffer(target: Int, buffer:
WebGLBuffer?)\n  fun bindFramebuffer(target: Int, framebuffer: WebGLFramebuffer?)\n  fun
bindRenderbuffer(target: Int, renderbuffer: WebGLRenderbuffer?)\n  fun bindTexture(target: Int, texture:
WebGLTexture?)\n  fun blendColor(red: Float, green: Float, blue: Float, alpha: Float)\n  fun
blendEquation(mode: Int)\n  fun blendEquationSeparate(modeRGB: Int, modeAlpha: Int)\n  fun
blendFunc(sfactor: Int, dfactor: Int)\n  fun blendFuncSeparate(srcRGB: Int, dstRGB: Int, srcAlpha: Int, dstAlpha:
Int)\n  fun bufferData(target: Int, size: Int, usage: Int)\n  fun bufferData(target: Int, data: BufferDataSource?,
usage: Int)\n  fun bufferSubData(target: Int, offset: Int, data: BufferDataSource?)\n  fun
checkFramebufferStatus(target: Int): Int\n  fun clear(mask: Int)\n  fun clearColor(red: Float, green: Float, blue:
Float, alpha: Float)\n  fun clearDepth(depth: Float)\n  fun clearStencil(s: Int)\n  fun colorMask(red: Boolean,
green: Boolean, blue: Boolean, alpha: Boolean)\n  fun compileShader(shader: WebGLShader?)\n  fun
compressedTexImage2D(target: Int, level: Int, internalformat: Int, width: Int, height: Int, border: Int, data:
ArrayBufferView)\n  fun compressedTexSubImage2D(target: Int, level: Int, xoffset: Int, yoffset: Int, width: Int,
height: Int, format: Int, data: ArrayBufferView)\n  fun copyTexImage2D(target: Int, level: Int, internalformat: Int,
x: Int, y: Int, width: Int, height: Int, border: Int)\n  fun copyTexSubImage2D(target: Int, level: Int, xoffset: Int,
yoffset: Int, x: Int, y: Int, width: Int, height: Int)\n  fun createBuffer(): WebGLBuffer?\n  fun createFramebuffer():
WebGLFramebuffer?\n  fun createProgram(): WebGLProgram?\n  fun createRenderbuffer():
WebGLRenderbuffer?\n  fun createShader(type: Int): WebGLShader?\n  fun createTexture(): WebGLTexture?\n
fun cullFace(mode: Int)\n  fun deleteBuffer(buffer: WebGLBuffer?)\n  fun deleteFramebuffer(framebuffer:
WebGLFramebuffer?)\n  fun deleteProgram(program: WebGLProgram?)\n  fun deleteRenderbuffer(renderbuffer:
WebGLRenderbuffer?)\n  fun deleteShader(shader: WebGLShader?)\n  fun deleteTexture(texture:
WebGLTexture?)\n  fun depthFunc(func: Int)\n  fun depthMask(flag: Boolean)\n  fun depthRange(zNear: Float,
zFar: Float)\n  fun detachShader(program: WebGLProgram?, shader: WebGLShader?)\n  fun disable(cap: Int)\n
fun disableVertexAttribArray(index: Int)\n  fun drawArrays(mode: Int, first: Int, count: Int)\n  fun
drawElements(mode: Int, count: Int, type: Int, offset: Int)\n  fun enable(cap: Int)\n  fun
enableVertexAttribArray(index: Int)\n  fun finish()\n  fun flush()\n  fun framebufferRenderbuffer(target: Int,
attachment: Int, renderbuffertarget: Int, renderbuffer: WebGLRenderbuffer?)\n  fun framebufferTexture2D(target:
Int, attachment: Int, textarget: Int, texture: WebGLTexture?, level: Int)\n  fun frontFace(mode: Int)\n  fun
generateMipmap(target: Int)\n  fun getActiveAttrib(program: WebGLProgram?, index: Int): WebGLActiveInfo?\n
fun getActiveUniform(program: WebGLProgram?, index: Int): WebGLActiveInfo?\n  fun
getAttachedShaders(program: WebGLProgram?): Array<WebGLShader>?\n  fun getAttribLocation(program:
WebGLProgram?, name: String): Int\n  fun getBufferParameter(target: Int, pname: Int): Any?\n  fun
getParameter(pname: Int): Any?\n  fun getError(): Int\n  fun getFramebufferAttachmentParameter(target: Int,
attachment: Int, pname: Int): Any?\n  fun getProgramParameter(program: WebGLProgram?, pname: Int): Any?\n

```

```

fun getProgramInfoLog(program: WebGLProgram?): String?\n fun getRenderbufferParameter(target: Int, pname:
Int): Any?\n fun getShaderParameter(shader: WebGLShader?, pname: Int): Any?\n fun
getShaderPrecisionFormat(shadertype: Int, precisiontype: Int): WebGLShaderPrecisionFormat?\n fun
getShaderInfoLog(shader: WebGLShader?): String?\n fun getShaderSource(shader: WebGLShader?): String?\n
fun getTexParameter(target: Int, pname: Int): Any?\n fun getUniform(program: WebGLProgram?, location:
WebGLUniformLocation?): Any?\n fun getUniformLocation(program: WebGLProgram?, name: String):
WebGLUniformLocation?\n fun getVertexAttrib(index: Int, pname: Int): Any?\n fun
getVertexAttribOffset(index: Int, pname: Int): Int\n fun hint(target: Int, mode: Int)\n fun isBuffer(buffer:
WebGLBuffer?): Boolean\n fun isEnabled(cap: Int): Boolean\n fun isFramebuffer(framebuffer:
WebGLFramebuffer?): Boolean\n fun isProgram(program: WebGLProgram?): Boolean\n fun
isRenderbuffer(renderbuffer: WebGLRenderbuffer?): Boolean\n fun isShader(shader: WebGLShader?): Boolean\n
fun isTexture(texture: WebGLTexture?): Boolean\n fun lineWidth(width: Float)\n fun linkProgram(program:
WebGLProgram?)\n fun pixelStorei(pname: Int, param: Int)\n fun polygonOffset(factor: Float, units: Float)\n
fun readPixels(x: Int, y: Int, width: Int, height: Int, format: Int, type: Int, pixels: ArrayBufferView?)\n fun
renderbufferStorage(target: Int, internalformat: Int, width: Int, height: Int)\n fun sampleCoverage(value: Float,
invert: Boolean)\n fun scissor(x: Int, y: Int, width: Int, height: Int)\n fun shaderSource(shader: WebGLShader?,
source: String)\n fun stencilFunc(func: Int, ref: Int, mask: Int)\n fun stencilFuncSeparate(face: Int, func: Int, ref:
Int, mask: Int)\n fun stencilMask(mask: Int)\n fun stencilMaskSeparate(face: Int, mask: Int)\n fun
stencilOp(fail: Int, zfail: Int, zpass: Int)\n fun stencilOpSeparate(face: Int, fail: Int, zfail: Int, zpass: Int)\n fun
texImage2D(target: Int, level: Int, internalformat: Int, width: Int, height: Int, border: Int, format: Int, type: Int, pixels:
ArrayBufferView?)\n fun texImage2D(target: Int, level: Int, internalformat: Int, format: Int, type: Int, source:
TexImageSource?)\n fun texParameterf(target: Int, pname: Int, param: Float)\n fun texParameteri(target: Int,
pname: Int, param: Int)\n fun texSubImage2D(target: Int, level: Int, xoffset: Int, yoffset: Int, width: Int, height: Int,
format: Int, type: Int, pixels: ArrayBufferView?)\n fun texSubImage2D(target: Int, level: Int, xoffset: Int, yoffset:
Int, format: Int, type: Int, source: TexImageSource?)\n fun uniform1f(location: WebGLUniformLocation?, x:
Float)\n fun uniform1fv(location: WebGLUniformLocation?, v: Float32Array)\n fun uniform1fv(location:
WebGLUniformLocation?, v: Array<Float>)\n fun uniform1i(location: WebGLUniformLocation?, x: Int)\n fun
uniform1iv(location: WebGLUniformLocation?, v: Int32Array)\n fun uniform1iv(location:
WebGLUniformLocation?, v: Array<Int>)\n fun uniform2f(location: WebGLUniformLocation?, x: Float, y:
Float)\n fun uniform2fv(location: WebGLUniformLocation?, v: Float32Array)\n fun uniform2fv(location:
WebGLUniformLocation?, v: Array<Float>)\n fun uniform2i(location: WebGLUniformLocation?, x: Int, y: Int)\n
fun uniform2iv(location: WebGLUniformLocation?, v: Int32Array)\n fun uniform2iv(location:
WebGLUniformLocation?, v: Array<Int>)\n fun uniform3f(location: WebGLUniformLocation?, x: Float, y: Float,
z: Float)\n fun uniform3fv(location: WebGLUniformLocation?, v: Float32Array)\n fun uniform3fv(location:
WebGLUniformLocation?, v: Array<Float>)\n fun uniform3i(location: WebGLUniformLocation?, x: Int, y: Int, z:
Int)\n fun uniform3iv(location: WebGLUniformLocation?, v: Int32Array)\n fun uniform3iv(location:
WebGLUniformLocation?, v: Array<Int>)\n fun uniform4f(location: WebGLUniformLocation?, x: Float, y: Float,
z: Float, w: Float)\n fun uniform4fv(location: WebGLUniformLocation?, v: Float32Array)\n fun
uniform4fv(location: WebGLUniformLocation?, v: Array<Float>)\n fun uniform4i(location:
WebGLUniformLocation?, x: Int, y: Int, z: Int, w: Int)\n fun uniform4iv(location: WebGLUniformLocation?, v:
Int32Array)\n fun uniform4iv(location: WebGLUniformLocation?, v: Array<Int>)\n fun
uniformMatrix2fv(location: WebGLUniformLocation?, transpose: Boolean, value: Float32Array)\n fun
uniformMatrix2fv(location: WebGLUniformLocation?, transpose: Boolean, value: Array<Float>)\n fun
uniformMatrix3fv(location: WebGLUniformLocation?, transpose: Boolean, value: Float32Array)\n fun
uniformMatrix3fv(location: WebGLUniformLocation?, transpose: Boolean, value: Array<Float>)\n fun
uniformMatrix4fv(location: WebGLUniformLocation?, transpose: Boolean, value: Float32Array)\n fun
uniformMatrix4fv(location: WebGLUniformLocation?, transpose: Boolean, value: Array<Float>)\n fun
useProgram(program: WebGLProgram?)\n fun validateProgram(program: WebGLProgram?)\n fun

```

```

vertexAttrib1f(index: Int, x: Float)\n fun vertexAttrib1fv(index: Int, values: dynamic)\n fun
vertexAttrib2f(index: Int, x: Float, y: Float)\n fun vertexAttrib2fv(index: Int, values: dynamic)\n fun
vertexAttrib3f(index: Int, x: Float, y: Float, z: Float)\n fun vertexAttrib3fv(index: Int, values: dynamic)\n fun
vertexAttrib4f(index: Int, x: Float, y: Float, z: Float, w: Float)\n fun vertexAttrib4fv(index: Int, values: dynamic)\n
fun vertexAttribPointer(index: Int, size: Int, type: Int, normalized: Boolean, stride: Int, offset: Int)\n fun
viewport(x: Int, y: Int, width: Int, height: Int)\n\n companion object {\n val DEPTH_BUFFER_BIT: Int\n
val STENCIL_BUFFER_BIT: Int\n val COLOR_BUFFER_BIT: Int\n val POINTS: Int\n val LINES:
Int\n val LINE_LOOP: Int\n val LINE_STRIP: Int\n val TRIANGLES: Int\n val
TRIANGLE_STRIP: Int\n val TRIANGLE_FAN: Int\n val ZERO: Int\n val ONE: Int\n val
SRC_COLOR: Int\n val ONE_MINUS_SRC_COLOR: Int\n val SRC_ALPHA: Int\n val
ONE_MINUS_SRC_ALPHA: Int\n val DST_ALPHA: Int\n val ONE_MINUS_DST_ALPHA: Int\n
val DST_COLOR: Int\n val ONE_MINUS_DST_COLOR: Int\n val SRC_ALPHA_SATURATE: Int\n
val FUNC_ADD: Int\n val BLEND_EQUATION: Int\n val BLEND_EQUATION_RGB: Int\n val
BLEND_EQUATION_ALPHA: Int\n val FUNC_SUBTRACT: Int\n val FUNC_REVERSE_SUBTRACT:
Int\n val BLEND_DST_RGB: Int\n val BLEND_SRC_RGB: Int\n val BLEND_DST_ALPHA: Int\n
val BLEND_SRC_ALPHA: Int\n val CONSTANT_COLOR: Int\n val
ONE_MINUS_CONSTANT_COLOR: Int\n val CONSTANT_ALPHA: Int\n val
ONE_MINUS_CONSTANT_ALPHA: Int\n val BLEND_COLOR: Int\n val ARRAY_BUFFER: Int\n
val ELEMENT_ARRAY_BUFFER: Int\n val ARRAY_BUFFER_BINDING: Int\n val
ELEMENT_ARRAY_BUFFER_BINDING: Int\n val STREAM_DRAW: Int\n val STATIC_DRAW: Int\n
val DYNAMIC_DRAW: Int\n val BUFFER_SIZE: Int\n val BUFFER_USAGE: Int\n val
CURRENT_VERTEX_ATTRIB: Int\n val FRONT: Int\n val BACK: Int\n val FRONT_AND_BACK:
Int\n val CULL_FACE: Int\n val BLEND: Int\n val DITHER: Int\n val STENCIL_TEST: Int\n
val DEPTH_TEST: Int\n val SCISSOR_TEST: Int\n val POLYGON_OFFSET_FILL: Int\n val
SAMPLE_ALPHA_TO_COVERAGE: Int\n val SAMPLE_COVERAGE: Int\n val NO_ERROR: Int\n
val INVALID_ENUM: Int\n val INVALID_VALUE: Int\n val INVALID_OPERATION: Int\n val
OUT_OF_MEMORY: Int\n val CW: Int\n val CCW: Int\n val LINE_WIDTH: Int\n val
ALIASED_POINT_SIZE_RANGE: Int\n val ALIASED_LINE_WIDTH_RANGE: Int\n val
CULL_FACE_MODE: Int\n val FRONT_FACE: Int\n val DEPTH_RANGE: Int\n val
DEPTH_WRITEMASK: Int\n val DEPTH_CLEAR_VALUE: Int\n val DEPTH_FUNC: Int\n val
STENCIL_CLEAR_VALUE: Int\n val STENCIL_FUNC: Int\n val STENCIL_FAIL: Int\n val
STENCIL_PASS_DEPTH_FAIL: Int\n val STENCIL_PASS_DEPTH_PASS: Int\n val STENCIL_REF:
Int\n val STENCIL_VALUE_MASK: Int\n val STENCIL_WRITEMASK: Int\n val
STENCIL_BACK_FUNC: Int\n val STENCIL_BACK_FAIL: Int\n val
STENCIL_BACK_PASS_DEPTH_FAIL: Int\n val STENCIL_BACK_PASS_DEPTH_PASS: Int\n val
STENCIL_BACK_REF: Int\n val STENCIL_BACK_VALUE_MASK: Int\n val
STENCIL_BACK_WRITEMASK: Int\n val VIEWPORT: Int\n val SCISSOR_BOX: Int\n val
COLOR_CLEAR_VALUE: Int\n val COLOR_WRITEMASK: Int\n val UNPACK_ALIGNMENT: Int\n
val PACK_ALIGNMENT: Int\n val MAX_TEXTURE_SIZE: Int\n val MAX_VIEWPORT_DIMS: Int\n
val SUBPIXEL_BITS: Int\n val RED_BITS: Int\n val GREEN_BITS: Int\n val BLUE_BITS: Int\n
val ALPHA_BITS: Int\n val DEPTH_BITS: Int\n val STENCIL_BITS: Int\n val
POLYGON_OFFSET_UNITS: Int\n val POLYGON_OFFSET_FACTOR: Int\n val
TEXTURE_BINDING_2D: Int\n val SAMPLE_BUFFERS: Int\n val SAMPLES: Int\n val
SAMPLE_COVERAGE_VALUE: Int\n val SAMPLE_COVERAGE_INVERT: Int\n val
COMPRESSED_TEXTURE_FORMATS: Int\n val DONT_CARE: Int\n val FASTEST: Int\n val
NICEST: Int\n val GENERATE_MIPMAP_HINT: Int\n val BYTE: Int\n val UNSIGNED_BYTE:
Int\n val SHORT: Int\n val UNSIGNED_SHORT: Int\n val INT: Int\n val UNSIGNED_INT: Int\n
val FLOAT: Int\n val DEPTH_COMPONENT: Int\n val ALPHA: Int\n val RGB: Int\n val

```

RGBA: Int\n val LUMINANCE: Int\n val LUMINANCE_ALPHA: Int\n val
 UNSIGNED_SHORT_4_4_4_4: Int\n val UNSIGNED_SHORT_5_5_5_1: Int\n val
 UNSIGNED_SHORT_5_6_5: Int\n val FRAGMENT_SHADER: Int\n val VERTEX_SHADER: Int\n
 val MAX_VERTEX_ATTRIBS: Int\n val MAX_VERTEX_UNIFORM_VECTORS: Int\n val
 MAX_VARYING_VECTORS: Int\n val MAX_COMBINED_TEXTURE_IMAGE_UNITS: Int\n val
 MAX_VERTEX_TEXTURE_IMAGE_UNITS: Int\n val MAX_TEXTURE_IMAGE_UNITS: Int\n val
 MAX_FRAGMENT_UNIFORM_VECTORS: Int\n val SHADER_TYPE: Int\n val DELETE_STATUS:
 Int\n val LINK_STATUS: Int\n val VALIDATE_STATUS: Int\n val ATTACHED_SHADERS: Int\n
 val ACTIVE_UNIFORMS: Int\n val ACTIVE_ATTRIBUTES: Int\n val
 SHADING_LANGUAGE_VERSION: Int\n val CURRENT_PROGRAM: Int\n val NEVER: Int\n val
 LESS: Int\n val EQUAL: Int\n val LEQUAL: Int\n val GREATER: Int\n val NOTEQUAL: Int\n
 val GEQUAL: Int\n val ALWAYS: Int\n val KEEP: Int\n val REPLACE: Int\n val INCR: Int\n
 val DECR: Int\n val INVERT: Int\n val INCR_WRAP: Int\n val DECR_WRAP: Int\n val
 VENDOR: Int\n val RENDERER: Int\n val VERSION: Int\n val NEAREST: Int\n val LINEAR:
 Int\n val NEAREST_MIPMAP_NEAREST: Int\n val LINEAR_MIPMAP_NEAREST: Int\n val
 NEAREST_MIPMAP_LINEAR: Int\n val LINEAR_MIPMAP_LINEAR: Int\n val
 TEXTURE_MAG_FILTER: Int\n val TEXTURE_MIN_FILTER: Int\n val TEXTURE_WRAP_S: Int\n
 val TEXTURE_WRAP_T: Int\n val TEXTURE_2D: Int\n val TEXTURE: Int\n val
 TEXTURE_CUBE_MAP: Int\n val TEXTURE_BINDING_CUBE_MAP: Int\n val
 TEXTURE_CUBE_MAP_POSITIVE_X: Int\n val TEXTURE_CUBE_MAP_NEGATIVE_X: Int\n val
 TEXTURE_CUBE_MAP_POSITIVE_Y: Int\n val TEXTURE_CUBE_MAP_NEGATIVE_Y: Int\n val
 TEXTURE_CUBE_MAP_POSITIVE_Z: Int\n val TEXTURE_CUBE_MAP_NEGATIVE_Z: Int\n val
 MAX_CUBE_MAP_TEXTURE_SIZE: Int\n val TEXTURE0: Int\n val TEXTURE1: Int\n val
 TEXTURE2: Int\n val TEXTURE3: Int\n val TEXTURE4: Int\n val TEXTURE5: Int\n val
 TEXTURE6: Int\n val TEXTURE7: Int\n val TEXTURE8: Int\n val TEXTURE9: Int\n val
 TEXTURE10: Int\n val TEXTURE11: Int\n val TEXTURE12: Int\n val TEXTURE13: Int\n val
 TEXTURE14: Int\n val TEXTURE15: Int\n val TEXTURE16: Int\n val TEXTURE17: Int\n val
 TEXTURE18: Int\n val TEXTURE19: Int\n val TEXTURE20: Int\n val TEXTURE21: Int\n val
 TEXTURE22: Int\n val TEXTURE23: Int\n val TEXTURE24: Int\n val TEXTURE25: Int\n val
 TEXTURE26: Int\n val TEXTURE27: Int\n val TEXTURE28: Int\n val TEXTURE29: Int\n val
 TEXTURE30: Int\n val TEXTURE31: Int\n val ACTIVE_TEXTURE: Int\n val REPEAT: Int\n
 val CLAMP_TO_EDGE: Int\n val MIRRORED_REPEAT: Int\n val FLOAT_VEC2: Int\n val
 FLOAT_VEC3: Int\n val FLOAT_VEC4: Int\n val INT_VEC2: Int\n val INT_VEC3: Int\n val
 INT_VEC4: Int\n val BOOL: Int\n val BOOL_VEC2: Int\n val BOOL_VEC3: Int\n val
 BOOL_VEC4: Int\n val FLOAT_MAT2: Int\n val FLOAT_MAT3: Int\n val FLOAT_MAT4: Int\n
 val SAMPLER_2D: Int\n val SAMPLER_CUBE: Int\n val VERTEX_ATTRIB_ARRAY_ENABLED:
 Int\n val VERTEX_ATTRIB_ARRAY_SIZE: Int\n val VERTEX_ATTRIB_ARRAY_STRIDE: Int\n
 val VERTEX_ATTRIB_ARRAY_TYPE: Int\n val VERTEX_ATTRIB_ARRAY_NORMALIZED: Int\n
 val VERTEX_ATTRIB_ARRAY_POINTER: Int\n val VERTEX_ATTRIB_ARRAY_BUFFER_BINDING:
 Int\n val IMPLEMENTATION_COLOR_READ_TYPE: Int\n val
 IMPLEMENTATION_COLOR_READ_FORMAT: Int\n val COMPILE_STATUS: Int\n val
 LOW_FLOAT: Int\n val MEDIUM_FLOAT: Int\n val HIGH_FLOAT: Int\n val LOW_INT: Int\n
 val MEDIUM_INT: Int\n val HIGH_INT: Int\n val FRAMEBUFFER: Int\n val RENDERBUFFER:
 Int\n val RGBA4: Int\n val RGB5_A1: Int\n val RGB565: Int\n val DEPTH_COMPONENT16:
 Int\n val STENCIL_INDEX: Int\n val STENCIL_INDEX8: Int\n val DEPTH_STENCIL: Int\n val
 RENDERBUFFER_WIDTH: Int\n val RENDERBUFFER_HEIGHT: Int\n val
 RENDERBUFFER_INTERNAL_FORMAT: Int\n val RENDERBUFFER_RED_SIZE: Int\n val
 RENDERBUFFER_GREEN_SIZE: Int\n val RENDERBUFFER_BLUE_SIZE: Int\n val

```

RENDERBUFFER_ALPHA_SIZE: Int\n    val RENDERBUFFER_DEPTH_SIZE: Int\n    val
RENDERBUFFER_STENCIL_SIZE: Int\n    val FRAMEBUFFER_ATTACHMENT_OBJECT_TYPE: Int\n
val FRAMEBUFFER_ATTACHMENT_OBJECT_NAME: Int\n    val
FRAMEBUFFER_ATTACHMENT_TEXTURE_LEVEL: Int\n    val
FRAMEBUFFER_ATTACHMENT_TEXTURE_CUBE_MAP_FACE: Int\n    val COLOR_ATTACHMENT0:
Int\n    val DEPTH_ATTACHMENT: Int\n    val STENCIL_ATTACHMENT: Int\n    val
DEPTH_STENCIL_ATTACHMENT: Int\n    val NONE: Int\n    val FRAMEBUFFER_COMPLETE: Int\n
val FRAMEBUFFER_INCOMPLETE_ATTACHMENT: Int\n    val
FRAMEBUFFER_INCOMPLETE_DIMENSIONS: Int\n    val FRAMEBUFFER_UNSUPPORTED: Int\n
val FRAMEBUFFER_BINDING: Int\n    val RENDERBUFFER_BINDING: Int\n    val
MAX_RENDERBUFFER_SIZE: Int\n    val INVALID_FRAMEBUFFER_OPERATION: Int\n    val
UNPACK_FLIP_Y_WEBGL: Int\n    val UNPACK_PREMULTIPLY_ALPHA_WEBGL: Int\n    val
CONTEXT_LOST_WEBGL: Int\n    val UNPACK_COLORSPACE_CONVERSION_WEBGL: Int\n    val
BROWSER_DEFAULT_WEBGL: Int\n    }\n}\n\n/**\n * Exposes the JavaScript
[WebGLRenderingContext](https://developer.mozilla.org/en/docs/Web/API/WebGLRenderingContext) to Kotlin\n
*\npublic external abstract class WebGLRenderingContext : WebGLRenderingContextBase, RenderingContext {\n
companion object {\n    val DEPTH_BUFFER_BIT: Int\n    val STENCIL_BUFFER_BIT: Int\n    val
COLOR_BUFFER_BIT: Int\n    val POINTS: Int\n    val LINES: Int\n    val LINE_LOOP: Int\n    val
LINE_STRIP: Int\n    val TRIANGLES: Int\n    val TRIANGLE_STRIP: Int\n    val TRIANGLE_FAN:
Int\n    val ZERO: Int\n    val ONE: Int\n    val SRC_COLOR: Int\n    val ONE_MINUS_SRC_COLOR:
Int\n    val SRC_ALPHA: Int\n    val ONE_MINUS_SRC_ALPHA: Int\n    val DST_ALPHA: Int\n    val
ONE_MINUS_DST_ALPHA: Int\n    val DST_COLOR: Int\n    val ONE_MINUS_DST_COLOR: Int\n
val SRC_ALPHA_SATURATE: Int\n    val FUNC_ADD: Int\n    val BLEND_EQUATION: Int\n    val
BLEND_EQUATION_RGB: Int\n    val BLEND_EQUATION_ALPHA: Int\n    val FUNC_SUBTRACT:
Int\n    val FUNC_REVERSE_SUBTRACT: Int\n    val BLEND_DST_RGB: Int\n    val
BLEND_SRC_RGB: Int\n    val BLEND_DST_ALPHA: Int\n    val BLEND_SRC_ALPHA: Int\n    val
CONSTANT_COLOR: Int\n    val ONE_MINUS_CONSTANT_COLOR: Int\n    val CONSTANT_ALPHA:
Int\n    val ONE_MINUS_CONSTANT_ALPHA: Int\n    val BLEND_COLOR: Int\n    val
ARRAY_BUFFER: Int\n    val ELEMENT_ARRAY_BUFFER: Int\n    val ARRAY_BUFFER_BINDING:
Int\n    val ELEMENT_ARRAY_BUFFER_BINDING: Int\n    val STREAM_DRAW: Int\n    val
STATIC_DRAW: Int\n    val DYNAMIC_DRAW: Int\n    val BUFFER_SIZE: Int\n    val
BUFFER_USAGE: Int\n    val CURRENT_VERTEX_ATTRIB: Int\n    val FRONT: Int\n    val BACK:
Int\n    val FRONT_AND_BACK: Int\n    val CULL_FACE: Int\n    val BLEND: Int\n    val DITHER:
Int\n    val STENCIL_TEST: Int\n    val DEPTH_TEST: Int\n    val SCISSOR_TEST: Int\n    val
POLYGON_OFFSET_FILL: Int\n    val SAMPLE_ALPHA_TO_COVERAGE: Int\n    val
SAMPLE_COVERAGE: Int\n    val NO_ERROR: Int\n    val INVALID_ENUM: Int\n    val
INVALID_VALUE: Int\n    val INVALID_OPERATION: Int\n    val OUT_OF_MEMORY: Int\n    val CW:
Int\n    val CCW: Int\n    val LINE_WIDTH: Int\n    val ALIASED_POINT_SIZE_RANGE: Int\n    val
ALIASED_LINE_WIDTH_RANGE: Int\n    val CULL_FACE_MODE: Int\n    val FRONT_FACE: Int\n
val DEPTH_RANGE: Int\n    val DEPTH_WRITEMASK: Int\n    val DEPTH_CLEAR_VALUE: Int\n    val
DEPTH_FUNC: Int\n    val STENCIL_CLEAR_VALUE: Int\n    val STENCIL_FUNC: Int\n    val
STENCIL_FAIL: Int\n    val STENCIL_PASS_DEPTH_FAIL: Int\n    val STENCIL_PASS_DEPTH_PASS:
Int\n    val STENCIL_REF: Int\n    val STENCIL_VALUE_MASK: Int\n    val STENCIL_WRITEMASK:
Int\n    val STENCIL_BACK_FUNC: Int\n    val STENCIL_BACK_FAIL: Int\n    val
STENCIL_BACK_PASS_DEPTH_FAIL: Int\n    val STENCIL_BACK_PASS_DEPTH_PASS: Int\n    val
STENCIL_BACK_REF: Int\n    val STENCIL_BACK_VALUE_MASK: Int\n    val
STENCIL_BACK_WRITEMASK: Int\n    val VIEWPORT: Int\n    val SCISSOR_BOX: Int\n    val

```

COLOR_CLEAR_VALUE: Int\n val COLOR_WRITEMASK: Int\n val UNPACK_ALIGNMENT: Int\n
 val PACK_ALIGNMENT: Int\n val MAX_TEXTURE_SIZE: Int\n val MAX_VIEWPORT_DIMS: Int\n
 val SUBPIXEL_BITS: Int\n val RED_BITS: Int\n val GREEN_BITS: Int\n val BLUE_BITS: Int\n
 val ALPHA_BITS: Int\n val DEPTH_BITS: Int\n val STENCIL_BITS: Int\n val
 POLYGON_OFFSET_UNITS: Int\n val POLYGON_OFFSET_FACTOR: Int\n val
 TEXTURE_BINDING_2D: Int\n val SAMPLE_BUFFERS: Int\n val SAMPLES: Int\n val
 SAMPLE_COVERAGE_VALUE: Int\n val SAMPLE_COVERAGE_INVERT: Int\n val
 COMPRESSED_TEXTURE_FORMATS: Int\n val DONT_CARE: Int\n val FASTEST: Int\n val
 NICEST: Int\n val GENERATE_MIPMAP_HINT: Int\n val BYTE: Int\n val UNSIGNED_BYTE:
 Int\n val SHORT: Int\n val UNSIGNED_SHORT: Int\n val INT: Int\n val UNSIGNED_INT: Int\n
 val FLOAT: Int\n val DEPTH_COMPONENT: Int\n val ALPHA: Int\n val RGB: Int\n val
 RGBA: Int\n val LUMINANCE: Int\n val LUMINANCE_ALPHA: Int\n val
 UNSIGNED_SHORT_4_4_4_4: Int\n val UNSIGNED_SHORT_5_5_5_1: Int\n val
 UNSIGNED_SHORT_5_6_5: Int\n val FRAGMENT_SHADER: Int\n val VERTEX_SHADER: Int\n
 val MAX_VERTEX_ATTRIBS: Int\n val MAX_VERTEX_UNIFORM_VECTORS: Int\n val
 MAX_VARYING_VECTORS: Int\n val MAX_COMBINED_TEXTURE_IMAGE_UNITS: Int\n val
 MAX_VERTEX_TEXTURE_IMAGE_UNITS: Int\n val MAX_TEXTURE_IMAGE_UNITS: Int\n val
 MAX_FRAGMENT_UNIFORM_VECTORS: Int\n val SHADER_TYPE: Int\n val DELETE_STATUS:
 Int\n val LINK_STATUS: Int\n val VALIDATE_STATUS: Int\n val ATTACHED_SHADERS: Int\n
 val ACTIVE_UNIFORMS: Int\n val ACTIVE_ATTRIBUTES: Int\n val
 SHADING_LANGUAGE_VERSION: Int\n val CURRENT_PROGRAM: Int\n val NEVER: Int\n val
 LESS: Int\n val EQUAL: Int\n val LEQUAL: Int\n val GREATER: Int\n val NOTEQUAL: Int\n
 val GEQUAL: Int\n val ALWAYS: Int\n val KEEP: Int\n val REPLACE: Int\n val INCR: Int\n
 val DECR: Int\n val INVERT: Int\n val INCR_WRAP: Int\n val DECR_WRAP: Int\n val
 VENDOR: Int\n val RENDERER: Int\n val VERSION: Int\n val NEAREST: Int\n val LINEAR:
 Int\n val NEAREST_MIPMAP_NEAREST: Int\n val LINEAR_MIPMAP_NEAREST: Int\n val
 NEAREST_MIPMAP_LINEAR: Int\n val LINEAR_MIPMAP_LINEAR: Int\n val
 TEXTURE_MAG_FILTER: Int\n val TEXTURE_MIN_FILTER: Int\n val TEXTURE_WRAP_S: Int\n
 val TEXTURE_WRAP_T: Int\n val TEXTURE_2D: Int\n val TEXTURE: Int\n val
 TEXTURE_CUBE_MAP: Int\n val TEXTURE_BINDING_CUBE_MAP: Int\n val
 TEXTURE_CUBE_MAP_POSITIVE_X: Int\n val TEXTURE_CUBE_MAP_NEGATIVE_X: Int\n val
 TEXTURE_CUBE_MAP_POSITIVE_Y: Int\n val TEXTURE_CUBE_MAP_NEGATIVE_Y: Int\n val
 TEXTURE_CUBE_MAP_POSITIVE_Z: Int\n val TEXTURE_CUBE_MAP_NEGATIVE_Z: Int\n val
 MAX_CUBE_MAP_TEXTURE_SIZE: Int\n val TEXTURE0: Int\n val TEXTURE1: Int\n val
 TEXTURE2: Int\n val TEXTURE3: Int\n val TEXTURE4: Int\n val TEXTURE5: Int\n val
 TEXTURE6: Int\n val TEXTURE7: Int\n val TEXTURE8: Int\n val TEXTURE9: Int\n val
 TEXTURE10: Int\n val TEXTURE11: Int\n val TEXTURE12: Int\n val TEXTURE13: Int\n val
 TEXTURE14: Int\n val TEXTURE15: Int\n val TEXTURE16: Int\n val TEXTURE17: Int\n val
 TEXTURE18: Int\n val TEXTURE19: Int\n val TEXTURE20: Int\n val TEXTURE21: Int\n val
 TEXTURE22: Int\n val TEXTURE23: Int\n val TEXTURE24: Int\n val TEXTURE25: Int\n val
 TEXTURE26: Int\n val TEXTURE27: Int\n val TEXTURE28: Int\n val TEXTURE29: Int\n val
 TEXTURE30: Int\n val TEXTURE31: Int\n val ACTIVE_TEXTURE: Int\n val REPEAT: Int\n
 val CLAMP_TO_EDGE: Int\n val MIRRORED_REPEAT: Int\n val FLOAT_VEC2: Int\n val
 FLOAT_VEC3: Int\n val FLOAT_VEC4: Int\n val INT_VEC2: Int\n val INT_VEC3: Int\n val
 INT_VEC4: Int\n val BOOL: Int\n val BOOL_VEC2: Int\n val BOOL_VEC3: Int\n val
 BOOL_VEC4: Int\n val FLOAT_MAT2: Int\n val FLOAT_MAT3: Int\n val FLOAT_MAT4: Int\n
 val SAMPLER_2D: Int\n val SAMPLER_CUBE: Int\n val VERTEX_ATTRIB_ARRAY_ENABLED:
 Int\n val VERTEX_ATTRIB_ARRAY_SIZE: Int\n val VERTEX_ATTRIB_ARRAY_STRIDE: Int\n


```

override val byteLength: Int\n fun set(array: Int8Array, offset: Int = definedExternally)\n fun set(array:
Array<Byte>, offset: Int = definedExternally)\n fun subarray(start: Int, end: Int): Int8Array\n\n companion
object {\n val BYTES_PER_ELEMENT: Int\n }\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Int8Array.get(index: Int):
Byte = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Int8Array.set(index: Int,
value: Byte) { asDynamic()[index] = value }\n\n/**\n * Exposes the JavaScript
[Uint8Array](https://developer.mozilla.org/en/docs/Web/API/Uint8Array) to Kotlin\n */\npublic external open class
Uint8Array : ArrayBufferView {\n constructor(length: Int)\n constructor(array: Uint8Array)\n
constructor(array: Array<Byte>)\n constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length:
Int = definedExternally)\n open val length: Int\n override val buffer: ArrayBuffer\n override val byteOffset:
Int\n override val byteLength: Int\n fun set(array: Uint8Array, offset: Int = definedExternally)\n fun set(array:
Array<Byte>, offset: Int = definedExternally)\n fun subarray(start: Int, end: Int): Uint8Array\n\n companion
object {\n val BYTES_PER_ELEMENT: Int\n }\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Uint8Array.get(index: Int):
Byte = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Uint8Array.set(index: Int,
value: Byte) { asDynamic()[index] = value }\n\n/**\n * Exposes the JavaScript
[Uint8ClampedArray](https://developer.mozilla.org/en/docs/Web/API/Uint8ClampedArray) to Kotlin\n */\npublic
external open class Uint8ClampedArray : ArrayBufferView {\n constructor(length: Int)\n constructor(array:
Uint8ClampedArray)\n constructor(array: Array<Byte>)\n constructor(buffer: ArrayBuffer, byteOffset: Int =
definedExternally, length: Int = definedExternally)\n open val length: Int\n override val buffer: ArrayBuffer\n
override val byteOffset: Int\n override val byteLength: Int\n fun set(array: Uint8ClampedArray, offset: Int =
definedExternally)\n fun set(array: Array<Byte>, offset: Int = definedExternally)\n fun subarray(start: Int, end:
Int): Uint8ClampedArray\n\n companion object {\n val BYTES_PER_ELEMENT: Int\n
}\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun
Uint8ClampedArray.get(index: Int): Byte = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun
Uint8ClampedArray.set(index: Int, value: Byte) { asDynamic()[index] = value }\n\n/**\n * Exposes the JavaScript
[Int16Array](https://developer.mozilla.org/en/docs/Web/API/Int16Array) to Kotlin\n */\npublic external open class
Int16Array : ArrayBufferView {\n constructor(length: Int)\n constructor(array: Int16Array)\n
constructor(array: Array<Short>)\n constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length:
Int = definedExternally)\n open val length: Int\n override val buffer: ArrayBuffer\n override val byteOffset:
Int\n override val byteLength: Int\n fun set(array: Int16Array, offset: Int = definedExternally)\n fun set(array:
Array<Short>, offset: Int = definedExternally)\n fun subarray(start: Int, end: Int): Int16Array\n\n companion
object {\n val BYTES_PER_ELEMENT: Int\n }\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Int16Array.get(index: Int):
Short = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Int16Array.set(index: Int,
value: Short) { asDynamic()[index] = value }\n\n/**\n * Exposes the JavaScript
[Uint16Array](https://developer.mozilla.org/en/docs/Web/API/Uint16Array) to Kotlin\n */\npublic external open
class Uint16Array : ArrayBufferView {\n constructor(length: Int)\n constructor(array: Uint16Array)\n
constructor(array: Array<Short>)\n constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length:
Int = definedExternally)\n open val length: Int\n override val buffer: ArrayBuffer\n override val byteOffset:
Int\n override val byteLength: Int\n fun set(array: Uint16Array, offset: Int = definedExternally)\n fun set(array:
Array<Short>, offset: Int = definedExternally)\n fun subarray(start: Int, end: Int): Uint16Array\n\n companion
object {\n val BYTES_PER_ELEMENT: Int\n }\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",

```

```

\ "INVISIBLE_MEMBER" ) \n @kotlin.internal.InlineOnly \n public inline operator fun Uint16Array.get(index: Int): Short = asDynamic()[index] \n \n @Suppress("INVISIBLE_REFERENCE",
\ "INVISIBLE_MEMBER" ) \n @kotlin.internal.InlineOnly \n public inline operator fun Uint16Array.set(index: Int, value: Short) { asDynamic()[index] = value } \n \n /** \n * Exposes the JavaScript
[Int32Array](https://developer.mozilla.org/en/docs/Web/API/Int32Array) to Kotlin \n * \n \n public external open class Int32Array : ArrayBufferView { \n    constructor(length: Int) \n    constructor(array: Int32Array) \n    constructor(array: Array<Int>) \n    constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length: Int = definedExternally) \n    open val length: Int \n    override val buffer: ArrayBuffer \n    override val byteOffset: Int \n    override val byteLength: Int \n    fun set(array: Int32Array, offset: Int = definedExternally) \n    fun set(array: Array<Int>, offset: Int = definedExternally) \n    fun subarray(start: Int, end: Int): Int32Array \n \n    companion object { \n        val BYTES_PER_ELEMENT: Int \n    } \n } \n \n @Suppress("INVISIBLE_REFERENCE",
\ "INVISIBLE_MEMBER" ) \n @kotlin.internal.InlineOnly \n public inline operator fun Int32Array.get(index: Int): Int = asDynamic()[index] \n \n @Suppress("INVISIBLE_REFERENCE",
\ "INVISIBLE_MEMBER" ) \n @kotlin.internal.InlineOnly \n public inline operator fun Int32Array.set(index: Int, value: Int) { asDynamic()[index] = value } \n \n /** \n * Exposes the JavaScript
[Uint32Array](https://developer.mozilla.org/en/docs/Web/API/Uint32Array) to Kotlin \n * \n \n public external open class Uint32Array : ArrayBufferView { \n    constructor(length: Int) \n    constructor(array: Uint32Array) \n    constructor(array: Array<Int>) \n    constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length: Int = definedExternally) \n    open val length: Int \n    override val buffer: ArrayBuffer \n    override val byteOffset: Int \n    override val byteLength: Int \n    fun set(array: Uint32Array, offset: Int = definedExternally) \n    fun set(array: Array<Int>, offset: Int = definedExternally) \n    fun subarray(start: Int, end: Int): Uint32Array \n \n    companion object { \n        val BYTES_PER_ELEMENT: Int \n    } \n } \n \n @Suppress("INVISIBLE_REFERENCE",
\ "INVISIBLE_MEMBER" ) \n @kotlin.internal.InlineOnly \n public inline operator fun Uint32Array.get(index: Int): Int = asDynamic()[index] \n \n @Suppress("INVISIBLE_REFERENCE",
\ "INVISIBLE_MEMBER" ) \n @kotlin.internal.InlineOnly \n public inline operator fun Uint32Array.set(index: Int, value: Int) { asDynamic()[index] = value } \n \n /** \n * Exposes the JavaScript
[Float32Array](https://developer.mozilla.org/en/docs/Web/API/Float32Array) to Kotlin \n * \n \n public external open class Float32Array : ArrayBufferView { \n    constructor(length: Int) \n    constructor(array: Float32Array) \n    constructor(array: Array<Float>) \n    constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length: Int = definedExternally) \n    open val length: Int \n    override val buffer: ArrayBuffer \n    override val byteOffset: Int \n    override val byteLength: Int \n    fun set(array: Float32Array, offset: Int = definedExternally) \n    fun set(array: Array<Float>, offset: Int = definedExternally) \n    fun subarray(start: Int, end: Int): Float32Array \n \n    companion object { \n        val BYTES_PER_ELEMENT: Int \n    } \n } \n \n @Suppress("INVISIBLE_REFERENCE",
\ "INVISIBLE_MEMBER" ) \n @kotlin.internal.InlineOnly \n public inline operator fun Float32Array.get(index: Int): Float = asDynamic()[index] \n \n @Suppress("INVISIBLE_REFERENCE",
\ "INVISIBLE_MEMBER" ) \n @kotlin.internal.InlineOnly \n public inline operator fun Float32Array.set(index: Int, value: Float) { asDynamic()[index] = value } \n \n /** \n * Exposes the JavaScript
[Float64Array](https://developer.mozilla.org/en/docs/Web/API/Float64Array) to Kotlin \n * \n \n public external open class Float64Array : ArrayBufferView { \n    constructor(length: Int) \n    constructor(array: Float64Array) \n    constructor(array: Array<Double>) \n    constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length: Int = definedExternally) \n    open val length: Int \n    override val buffer: ArrayBuffer \n    override val byteOffset: Int \n    override val byteLength: Int \n    fun set(array: Float64Array, offset: Int = definedExternally) \n    fun set(array: Array<Double>, offset: Int = definedExternally) \n    fun subarray(start: Int, end: Int): Float64Array \n \n    companion object { \n        val BYTES_PER_ELEMENT: Int \n    } \n } \n \n @Suppress("INVISIBLE_REFERENCE",
\ "INVISIBLE_MEMBER" ) \n @kotlin.internal.InlineOnly \n public inline operator fun Float64Array.get(index: Int): Double = asDynamic()[index] \n \n @Suppress("INVISIBLE_REFERENCE",

```

```

\ "INVISIBLE_MEMBER" ) \n @kotlin.internal.InlineOnly \n public inline operator fun Float64Array.set(index: Int,
value: Double) { asDynamic()[index] = value } \n \n /** \n * Exposes the JavaScript
[DataView](https://developer.mozilla.org/en/docs/Web/API/DataView) to Kotlin \n * \n \n public external open class
DataView(buffer: ArrayBuffer, byteOffset: Int = definedExternally, byteLength: Int = definedExternally) :
ArrayBufferView { \n override val buffer: ArrayBuffer \n override val byteOffset: Int \n override val
byteLength: Int \n fun getInt8(byteOffset: Int): Byte \n fun getUint8(byteOffset: Int): Byte \n fun
getInt16(byteOffset: Int, littleEndian: Boolean = definedExternally): Short \n fun getUint16(byteOffset: Int,
littleEndian: Boolean = definedExternally): Short \n fun getInt32(byteOffset: Int, littleEndian: Boolean =
definedExternally): Int \n fun getUint32(byteOffset: Int, littleEndian: Boolean = definedExternally): Int \n fun
getFloat32(byteOffset: Int, littleEndian: Boolean = definedExternally): Float \n fun getFloat64(byteOffset: Int,
littleEndian: Boolean = definedExternally): Double \n fun setInt8(byteOffset: Int, value: Byte) \n fun
setUint8(byteOffset: Int, value: Byte) \n fun setInt16(byteOffset: Int, value: Short, littleEndian: Boolean =
definedExternally) \n fun setUint16(byteOffset: Int, value: Short, littleEndian: Boolean = definedExternally) \n
fun setInt32(byteOffset: Int, value: Int, littleEndian: Boolean = definedExternally) \n fun setUint32(byteOffset: Int,
value: Int, littleEndian: Boolean = definedExternally) \n fun setFloat32(byteOffset: Int, value: Float, littleEndian:
Boolean = definedExternally) \n fun setFloat64(byteOffset: Int, value: Double, littleEndian: Boolean =
definedExternally) \n } \n \n public external interface BufferDataSource \n \n public external interface
TexImageSource", /* \n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors. \n *
Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file. \n
* \n \n // NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT! \n // See github.com/kotlin/dukat for
details \n \n package org.w3c.dom.clipboard \n \n import kotlin.js.* \n import org.khronos.webgl.* \n import
org.w3c.dom.* \n import org.w3c.dom.events.* \n \n public external interface ClipboardEventInit : EventInit { \n var
clipboardData: DataTransfer? /* = null */ \n get() = definedExternally \n set(value) =
definedExternally \n } \n \n @Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER") \n @kotlin.internal.InlineOnly \n public inline fun ClipboardEventInit(clipboardData:
DataTransfer? = null, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false):
ClipboardEventInit { \n val o = js("{}") \n o["clipboardData"] = clipboardData \n o["bubbles"] = bubbles \n
o["cancelable"] = cancelable \n o["composed"] = composed \n return o \n } \n \n /** \n * Exposes the JavaScript
[ClipboardEvent](https://developer.mozilla.org/en/docs/Web/API/ClipboardEvent) to Kotlin \n * \n \n public external
open class ClipboardEvent(type: String, eventInitDict: ClipboardEventInit = definedExternally) : Event { \n open
val clipboardData: DataTransfer? \n \n companion object { \n val NONE: Short \n val
CAPTURING_PHASE: Short \n val AT_TARGET: Short \n val BUBBLING_PHASE: Short \n
} \n } \n \n /** \n * Exposes the JavaScript [Clipboard](https://developer.mozilla.org/en/docs/Web/API/Clipboard) to
Kotlin \n * \n \n public external abstract class Clipboard : EventTarget { \n fun read(): Promise<DataTransfer> \n fun
readText(): Promise<String> \n fun write(data: DataTransfer): Promise<Unit> \n fun writeText(data: String):
Promise<Unit> \n } \n \n public external interface ClipboardPermissionDescriptor { \n var allowWithoutGesture:
Boolean? /* = false */ \n get() = definedExternally \n set(value) =
definedExternally \n } \n \n @Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER") \n @kotlin.internal.InlineOnly \n public inline fun
ClipboardPermissionDescriptor(allowWithoutGesture: Boolean? = false): ClipboardPermissionDescriptor { \n val
o = js("{}") \n o["allowWithoutGesture"] = allowWithoutGesture \n return o \n }", /* \n * Copyright 2010-
2021 JetBrains s.r.o. and Kotlin Programming Language contributors. \n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file. \n * \n \n // NOTE: THIS FILE IS AUTO-
GENERATED, DO NOT EDIT! \n // See github.com/kotlin/dukat for details \n \n package org.w3c.dom.css \n \n import
kotlin.js.* \n import org.khronos.webgl.* \n import org.w3c.dom.* \n \n public external abstract class MediaList :
ItemArrayLike<String> { \n open var mediaText: String \n fun appendMedium(medium: String) \n fun
deleteMedium(medium: String) \n override fun item(index: Int):
String? \n } \n \n @Suppress("INVISIBLE_REFERENCE",

```

```

\@kotlin.internal.InlineOnly\npublic inline operator fun MediaList.get(index: Int):
String? = asDynamic()[index]\n\n/**\n * Exposes the JavaScript
[StyleSheet](https://developer.mozilla.org/en/docs/Web/API/StyleSheet) to Kotlin\n *\npublic external abstract
class StyleSheet {\n  open val type: String\n  open val href: String?\n  open val ownerNode:
UnionElementOrProcessingInstruction?\n  open val parentStyleSheet: StyleSheet?\n  open val title: String?\n
open val media: MediaList\n  open var disabled: Boolean\n}\n\n/**\n * Exposes the JavaScript
[CSSStyleSheet](https://developer.mozilla.org/en/docs/Web/API/CSSStyleSheet) to Kotlin\n *\npublic external
abstract class CSSStyleSheet : StyleSheet {\n  open val ownerRule: CSSRule?\n  open val cssRules:
CSSRuleList\n  fun insertRule(rule: String, index: Int): Int\n  fun deleteRule(index: Int)\n}\n\n/**\n * Exposes the
JavaScript [StyleSheetList](https://developer.mozilla.org/en/docs/Web/API/StyleSheetList) to Kotlin\n *\npublic
external abstract class StyleSheetList : ItemArrayLike<StyleSheet> {\n  override fun item(index: Int):
StyleSheet?\n}\n\n@Suppress(\\"INVISIBLE_REFERENCE\"),
\@kotlin.internal.InlineOnly\npublic inline operator fun StyleSheetList.get(index: Int):
StyleSheet? = asDynamic()[index]\n\n/**\n * Exposes the JavaScript
[LinkStyle](https://developer.mozilla.org/en/docs/Web/API/LinkStyle) to Kotlin\n *\npublic external interface
LinkStyle {\n  val sheet: StyleSheet?\n  get() = definedExternally\n}\n\n/**\n * Exposes the JavaScript
[CSSRuleList](https://developer.mozilla.org/en/docs/Web/API/CSSRuleList) to Kotlin\n *\npublic external abstract
class CSSRuleList : ItemArrayLike<CSSRule> {\n  override fun item(index: Int):
CSSRule?\n}\n\n@Suppress(\\"INVISIBLE_REFERENCE\"),
\@kotlin.internal.InlineOnly\npublic inline operator fun CSSRuleList.get(index: Int):
CSSRule? = asDynamic()[index]\n\n/**\n * Exposes the JavaScript
[CSSRule](https://developer.mozilla.org/en/docs/Web/API/CSSRule) to Kotlin\n *\npublic external abstract class
CSSRule {\n  open val type: Short\n  open var cssText: String\n  open val parentRule: CSSRule?\n  open val
parentStyleSheet: CSSStyleSheet?\n\n  companion object {\n    val STYLE_RULE: Short\n    val
CHARSET_RULE: Short\n    val IMPORT_RULE: Short\n    val MEDIA_RULE: Short\n    val
FONT_FACE_RULE: Short\n    val PAGE_RULE: Short\n    val MARGIN_RULE: Short\n    val
NAMESPACE_RULE: Short\n  }\n}\n\n/**\n * Exposes the JavaScript
[CSSStyleRule](https://developer.mozilla.org/en/docs/Web/API/CSSStyleRule) to Kotlin\n *\npublic external
abstract class CSSStyleRule : CSSRule {\n  open var selectorText: String\n  open val style:
CSSStyleDeclaration\n\n  companion object {\n    val STYLE_RULE: Short\n    val CHARSET_RULE:
Short\n    val IMPORT_RULE: Short\n    val MEDIA_RULE: Short\n    val FONT_FACE_RULE: Short\n
    val PAGE_RULE: Short\n    val MARGIN_RULE: Short\n    val NAMESPACE_RULE: Short\n  }\n}\n\npublic external abstract class CSSImportRule : CSSRule {\n  open val href: String\n  open val media:
MediaList\n  open val styleSheet: CSSStyleSheet\n\n  companion object {\n    val STYLE_RULE: Short\n
    val CHARSET_RULE: Short\n    val IMPORT_RULE: Short\n    val MEDIA_RULE: Short\n    val
FONT_FACE_RULE: Short\n    val PAGE_RULE: Short\n    val MARGIN_RULE: Short\n    val
NAMESPACE_RULE: Short\n  }\n}\n\n/**\n * Exposes the JavaScript
[CSSGroupingRule](https://developer.mozilla.org/en/docs/Web/API/CSSGroupingRule) to Kotlin\n *\npublic
external abstract class CSSGroupingRule : CSSRule {\n  open val cssRules: CSSRuleList\n  fun insertRule(rule:
String, index: Int): Int\n  fun deleteRule(index: Int)\n\n  companion object {\n    val STYLE_RULE: Short\n
    val CHARSET_RULE: Short\n    val IMPORT_RULE: Short\n    val MEDIA_RULE: Short\n    val
FONT_FACE_RULE: Short\n    val PAGE_RULE: Short\n    val MARGIN_RULE: Short\n    val
NAMESPACE_RULE: Short\n  }\n}\n\n/**\n * Exposes the JavaScript
[CSSMediaRule](https://developer.mozilla.org/en/docs/Web/API/CSSMediaRule) to Kotlin\n *\npublic external
abstract class CSSMediaRule : CSSGroupingRule {\n  open val media: MediaList\n\n  companion object {\n
    val STYLE_RULE: Short\n    val CHARSET_RULE: Short\n    val IMPORT_RULE: Short\n    val
MEDIA_RULE: Short\n    val FONT_FACE_RULE: Short\n    val PAGE_RULE: Short\n    val
MARGIN_RULE: Short\n    val NAMESPACE_RULE: Short\n  }\n}\n\n/**\n * Exposes the JavaScript

```

[CSSPageRule](https://developer.mozilla.org/en/docs/Web/API/CSSPageRule) to Kotlin\n *\npublic external abstract class CSSPageRule : CSSGroupingRule {\n open var selectorText: String\n open val style: CSSStyleDeclaration\n\n companion object {\n val STYLE_RULE: Short\n val CHARSET_RULE: Short\n val IMPORT_RULE: Short\n val MEDIA_RULE: Short\n val FONT_FACE_RULE: Short\n val PAGE_RULE: Short\n val MARGIN_RULE: Short\n val NAMESPACE_RULE: Short\n }\n}\n\npublic external abstract class CSSMarginRule : CSSRule {\n open val name: String\n open val style: CSSStyleDeclaration\n\n companion object {\n val STYLE_RULE: Short\n val CHARSET_RULE: Short\n val IMPORT_RULE: Short\n val MEDIA_RULE: Short\n val FONT_FACE_RULE: Short\n val PAGE_RULE: Short\n val MARGIN_RULE: Short\n val NAMESPACE_RULE: Short\n }\n}\n\n/**\n * Exposes the JavaScript

[CSSNamespaceRule](https://developer.mozilla.org/en/docs/Web/API/CSSNamespaceRule) to Kotlin\n *\npublic external abstract class CSSNamespaceRule : CSSRule {\n open val namespaceURI: String\n open val prefix: String\n\n companion object {\n val STYLE_RULE: Short\n val CHARSET_RULE: Short\n val IMPORT_RULE: Short\n val MEDIA_RULE: Short\n val FONT_FACE_RULE: Short\n val PAGE_RULE: Short\n val MARGIN_RULE: Short\n val NAMESPACE_RULE: Short\n }\n}\n\n/**\n * Exposes the JavaScript

[CSSStyleDeclaration](https://developer.mozilla.org/en/docs/Web/API/CSSStyleDeclaration) to Kotlin\n *\npublic external abstract class CSSStyleDeclaration : ItemArrayLike<String> {\n open var cssText: String\n open val parentRule: CSSRule?\n open var cssFloat: String\n open var alignContent: String\n open var alignItems: String\n open var alignSelf: String\n open var animation: String\n open var animationDelay: String\n open var animationDirection: String\n open var animationDuration: String\n open var animationFillMode: String\n open var animationIterationCount: String\n open var animationName: String\n open var animationPlayState: String\n open var animationTimingFunction: String\n open var backfaceVisibility: String\n open var background: String\n open var backgroundAttachment: String\n open var backgroundClip: String\n open var backgroundColor: String\n open var backgroundImage: String\n open var backgroundOrigin: String\n open var backgroundPosition: String\n open var backgroundRepeat: String\n open var backgroundSize: String\n open var border: String\n open var borderBottom: String\n open var borderBottomColor: String\n open var borderBottomLeftRadius: String\n open var borderBottomRightRadius: String\n open var borderBottomStyle: String\n open var borderBottomWidth: String\n open var borderCollapse: String\n open var borderColor: String\n open var borderImage: String\n open var borderImageOutset: String\n open var borderImageRepeat: String\n open var borderImageSlice: String\n open var borderImageSource: String\n open var borderImageWidth: String\n open var borderLeft: String\n open var borderLeftColor: String\n open var borderLeftStyle: String\n open var borderLeftWidth: String\n open var borderRadius: String\n open var borderRight: String\n open var borderRightColor: String\n open var borderRightStyle: String\n open var borderRightWidth: String\n open var borderSpacing: String\n open var borderStyle: String\n open var borderTop: String\n open var borderTopColor: String\n open var borderTopLeftRadius: String\n open var borderTopRightRadius: String\n open var borderTopStyle: String\n open var borderTopWidth: String\n open var borderWidth: String\n open var bottom: String\n open var boxDecorationBreak: String\n open var boxShadow: String\n open var boxSizing: String\n open var breakAfter: String\n open var breakBefore: String\n open var breakInside: String\n open var captionSide: String\n open var clear: String\n open var clip: String\n open var color: String\n open var columnCount: String\n open var columnFill: String\n open var columnGap: String\n open var columnRule: String\n open var columnRuleColor: String\n open var columnRuleStyle: String\n open var columnRuleWidth: String\n open var columnSpan: String\n open var columnWidth: String\n open var columns: String\n open var content: String\n open var counterIncrement: String\n open var counterReset: String\n open var cursor: String\n open var direction: String\n open var display: String\n open var emptyCells: String\n open var filter: String\n open var flex: String\n open var flexBasis: String\n open var flexDirection: String\n open var flexFlow: String\n open var flexGrow: String\n open var flexShrink: String\n open var flexWrap: String\n open var font: String\n open var fontFamily:

```

String\n open var fontFeatureSettings: String\n open var fontKerning: String\n open var
fontLanguageOverride: String\n open var fontSize: String\n open var fontSizeAdjust: String\n open var
fontStretch: String\n open var fontStyle: String\n open var fontSynthesis: String\n open var fontVariant:
String\n open var fontVariantAlternates: String\n open var fontVariantCaps: String\n open var
fontVariantEastAsian: String\n open var fontVariantLigatures: String\n open var fontVariantNumeric: String\n
open var fontVariantPosition: String\n open var fontWeight: String\n open var hangingPunctuation: String\n
open var height: String\n open var hyphens: String\n open var imageOrientation: String\n open var
imageRendering: String\n open var imageResolution: String\n open var imeMode: String\n open var
justifyContent: String\n open var left: String\n open var letterSpacing: String\n open var lineBreak: String\n
open var lineHeight: String\n open var listStyle: String\n open var listStyleImage: String\n open var
listStylePosition: String\n open var listStyleType: String\n open var margin: String\n open var marginBottom:
String\n open var marginLeft: String\n open var marginRight: String\n open var marginTop: String\n open
var mark: String\n open var markAfter: String\n open var markBefore: String\n open var marks: String\n
open var marqueeDirection: String\n open var marqueePlayCount: String\n open var marqueeSpeed: String\n
open var marqueeStyle: String\n open var mask: String\n open var maskType: String\n open var maxHeight:
String\n open var maxWidth: String\n open var minHeight: String\n open var minWidth: String\n open var
navDown: String\n open var navIndex: String\n open var navLeft: String\n open var navRight: String\n open
var navUp: String\n open var objectFit: String\n open var objectPosition: String\n open var opacity: String\n
open var order: String\n open var orphans: String\n open var outline: String\n open var outlineColor: String\n
open var outlineOffset: String\n open var outlineStyle: String\n open var outlineWidth: String\n open var
overflowWrap: String\n open var overflowX: String\n open var overflowY: String\n open var padding:
String\n open var paddingBottom: String\n open var paddingLeft: String\n open var paddingRight: String\n
open var paddingTop: String\n open var pageBreakAfter: String\n open var pageBreakBefore: String\n open
var pageBreakInside: String\n open var perspective: String\n open var perspectiveOrigin: String\n open var
phonemes: String\n open var position: String\n open var quotes: String\n open var resize: String\n open var
rest: String\n open var restAfter: String\n open var restBefore: String\n open var right: String\n open var
tabSize: String\n open var tableLayout: String\n open var textAlign: String\n open var textAlignLast: String\n
open var textCombineUpright: String\n open var textDecoration: String\n open var textDecorationColor:
String\n open var textDecorationLine: String\n open var textDecorationStyle: String\n open var textIndent:
String\n open var textJustify: String\n open var textOrientation: String\n open var textOverflow: String\n
open var textShadow: String\n open var textTransform: String\n open var textUnderlinePosition: String\n open
var top: String\n open var transform: String\n open var transformOrigin: String\n open var transformStyle:
String\n open var transition: String\n open var transitionDelay: String\n open var transitionDuration: String\n
open var transitionProperty: String\n open var transitionTimingFunction: String\n open var unicodeBidi:
String\n open var verticalAlign: String\n open var visibility: String\n open var voiceBalance: String\n open
var voiceDuration: String\n open var voicePitch: String\n open var voicePitchRange: String\n open var
voiceRate: String\n open var voiceStress: String\n open var voiceVolume: String\n open var whiteSpace:
String\n open var widows: String\n open var width: String\n open var wordBreak: String\n open var
wordSpacing: String\n open var wordWrap: String\n open var writingMode: String\n open var zIndex: String\n
open var _dashed_attribute: String\n open var _camel_cased_attribute: String\n open var
_webkit_cased_attribute: String\n fun getPropertyValue(property: String): String\n fun
getPropertyPriority(property: String): String\n fun setProperty(property: String, value: String, priority: String =
definedExternally)\n fun setPropertyValue(property: String, value: String)\n fun setPropertyPriority(property:
String, priority: String)\n fun removeProperty(property: String): String\n override fun item(index: Int):
String\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun
CSSStyleDeclaration.get(index: Int): String? = asDynamic()[index]\n\npublic external interface
ElementCSSInlineStyle {\n val style: CSSStyleDeclaration\n}\n\n/**\n * Exposes the JavaScript

```

```

[CSS](https://developer.mozilla.org/en/docs/Web/API/CSS) to Kotlin\n *^npublic external abstract class CSS {\n
companion object {\n    fun escape(ident: String): String\n    }\n}\n\npublic external interface
UnionElementOrProcessingInstruction", "/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n *^n\n// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!\n// See
github.com/kotlin/dukat for details\n\npackage org.w3c.dom.encryptedmedia\n\nimport kotlin.js.*\nimport
org.khronos.webgl.*\nimport org.w3c.dom.*\nimport org.w3c.dom.events.*\n\n/**\n * Exposes the JavaScript
[MediaKeySystemConfiguration](https://developer.mozilla.org/en/docs/Web/API/MediaKeySystemConfiguration)
to Kotlin\n *^npublic external interface MediaKeySystemConfiguration {\n    var label: String? /* = \"\" *\n
get() = definedExternally\n    set(value) = definedExternally\n    var initDataTypes: Array<String>? /* = arrayOf()\n
*\n    get() = definedExternally\n    set(value) = definedExternally\n    var audioCapabilities:
Array<MediaKeySystemMediaCapability>? /* = arrayOf() *\n    get() = definedExternally\n    set(value) =
definedExternally\n    var videoCapabilities: Array<MediaKeySystemMediaCapability>? /* = arrayOf() *\n
get() = definedExternally\n    set(value) = definedExternally\n    var distinctiveIdentifier:
MediaKeysRequirement? /* = MediaKeysRequirement.OPTIONAL *\n    get() = definedExternally\n
set(value) = definedExternally\n    var persistentState: MediaKeysRequirement? /* =
MediaKeysRequirement.OPTIONAL *\n    get() = definedExternally\n    set(value) = definedExternally\n
var sessionTypes: Array<String>?\n    get() = definedExternally\n    set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun MediaKeySystemConfiguration(label:
String? = \"\", initDataTypes: Array<String>? = arrayOf(), audioCapabilities:
Array<MediaKeySystemMediaCapability>? = arrayOf(), videoCapabilities:
Array<MediaKeySystemMediaCapability>? = arrayOf(), distinctiveIdentifier: MediaKeysRequirement? =
MediaKeysRequirement.OPTIONAL, persistentState: MediaKeysRequirement? =
MediaKeysRequirement.OPTIONAL, sessionTypes: Array<String>? = undefined): MediaKeySystemConfiguration
{\n    val o = js(\"({})\")\n    o[\"label\"] = label\n    o[\"initDataTypes\"] = initDataTypes\n
o[\"audioCapabilities\"] = audioCapabilities\n    o[\"videoCapabilities\"] = videoCapabilities\n
o[\"distinctiveIdentifier\"] = distinctiveIdentifier\n    o[\"persistentState\"] = persistentState\n    o[\"sessionTypes\"]
= sessionTypes\n    return o\n}\n\npublic external interface MediaKeySystemMediaCapability {\n    var
contentType: String? /* = \"\" *\n    get() = definedExternally\n    set(value) = definedExternally\n    var
robustness: String? /* = \"\" *\n    get() = definedExternally\n    set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun
MediaKeySystemMediaCapability(contentType: String? = \"\", robustness: String? = \"\"):
MediaKeySystemMediaCapability {\n    val o = js(\"({})\")\n    o[\"contentType\"] = contentType\n
o[\"robustness\"] = robustness\n    return o\n}\n\n/**\n * Exposes the JavaScript
[MediaKeySystemAccess](https://developer.mozilla.org/en/docs/Web/API/MediaKeySystemAccess) to Kotlin\n
*\npublic external abstract class MediaKeySystemAccess {\n    open val keySystem: String\n    fun
getConfiguration(): MediaKeySystemConfiguration\n    fun createMediaKeys(): Promise<MediaKeys>\n}\n\n/**\n
* Exposes the JavaScript [MediaKeys](https://developer.mozilla.org/en/docs/Web/API/MediaKeys) to Kotlin\n
*\npublic external abstract class MediaKeys {\n    fun createSession(sessionType: MediaKeySessionType =
definedExternally): MediaKeySession\n    fun setServerCertificate(serverCertificate: dynamic):
Promise<Boolean>\n}\n\n/**\n * Exposes the JavaScript
[MediaKeySession](https://developer.mozilla.org/en/docs/Web/API/MediaKeySession) to Kotlin\n *^npublic
external abstract class MediaKeySession : EventTarget {\n    open val sessionId: String\n    open val expiration:
Double\n    open val closed: Promise<Unit>\n    open val keyStatuses: MediaKeyStatusMap\n    open var
onkeystatuschange: ((Event) -> dynamic)?\n    open var onmessage: ((MessageEvent) -> dynamic)?\n    fun
generateRequest(initDataType: String, initData: dynamic): Promise<Unit>\n    fun load(sessionId: String):

```

```

Promise<Boolean>\n fun update(response: dynamic): Promise<Unit>\n fun close(): Promise<Unit>\n fun
remove(): Promise<Unit>\n}\n\n/**\n * Exposes the JavaScript
[MediaKeyStatusMap](https://developer.mozilla.org/en/docs/Web/API/MediaKeyStatusMap) to Kotlin\n */\npublic
external abstract class MediaKeyStatusMap {\n open val size: Int\n fun has(keyId: dynamic): Boolean\n fun
get(keyId: dynamic): Any?\n}\n\n/**\n * Exposes the JavaScript
[MediaKeyMessageEvent](https://developer.mozilla.org/en/docs/Web/API/MediaKeyMessageEvent) to Kotlin\n
*/\npublic external open class MediaKeyMessageEvent(type: String, eventInitDict: MediaKeyMessageEventInit) :
Event {\n open val messageType: MediaKeyMessageType\n open val message: ArrayBuffer\n\n companion
object {\n val NONE: Short\n val CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val
BUBBLING_PHASE: Short\n }\n}\n\npublic external interface MediaKeyMessageEventInit : EventInit {\n var
messageType: MediaKeyMessageType?\n var message:
ArrayBuffer?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun
MediaKeyMessageEventInit(messageType: MediaKeyMessageType?, message: ArrayBuffer?, bubbles: Boolean? =
false, cancelable: Boolean? = false, composed: Boolean? = false): MediaKeyMessageEventInit {\n val o =
js(\"({})\")\n o[\"messageType\"] = messageType\n o[\"message\"] = message\n o[\"bubbles\"] = bubbles\n
o[\"cancelable\"] = cancelable\n o[\"composed\"] = composed\n return o\n}\n\npublic external open class
MediaEncryptedEvent(type: String, eventInitDict: MediaEncryptedEventInit = definedExternally) : Event {\n
open val initDataType: String\n open val initData: ArrayBuffer?\n\n companion object {\n val NONE:
Short\n val CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val BUBBLING_PHASE:
Short\n }\n}\n\npublic external interface MediaEncryptedEventInit : EventInit {\n var initDataType: String? /* =
\"\" */\n get() = definedExternally\n set(value) = definedExternally\n var initData: ArrayBuffer? /* = null
*/\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun
MediaEncryptedEventInit(initDataType: String? = \"\", initData: ArrayBuffer? = null, bubbles: Boolean? = false,
cancelable: Boolean? = false, composed: Boolean? = false): MediaEncryptedEventInit {\n val o = js(\"({})\")\n
o[\"initDataType\"] = initData\n o[\"initData\"] = initData\n o[\"bubbles\"] = bubbles\n o[\"cancelable\"]
= cancelable\n o[\"composed\"] = composed\n return o\n}\n\n/* please, don't implement this interface!
*/\n\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external
interface MediaKeysRequirement {\n companion object\n}\n\npublic inline val
MediaKeysRequirement.Companion.REQUIRED: MediaKeysRequirement get() =
\"required\".asDynamic().unsafeCast<MediaKeysRequirement>()\n\npublic inline val
MediaKeysRequirement.Companion.OPTIONAL: MediaKeysRequirement get() =
\"optional\".asDynamic().unsafeCast<MediaKeysRequirement>()\n\npublic inline val
MediaKeysRequirement.Companion.NOT_ALLOWED: MediaKeysRequirement get() = \"not-
allowed\".asDynamic().unsafeCast<MediaKeysRequirement>()\n\n/* please, don't implement this interface!
*/\n\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external
interface MediaKeySessionType {\n companion object\n}\n\npublic inline val
MediaKeySessionType.Companion.TEMPORARY: MediaKeySessionType get() =
\"temporary\".asDynamic().unsafeCast<MediaKeySessionType>()\n\npublic inline val
MediaKeySessionType.Companion.PERSISTENT_LICENSE: MediaKeySessionType get() = \"persistent-
license\".asDynamic().unsafeCast<MediaKeySessionType>()\n\n/* please, don't implement this interface!
*/\n\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external
interface MediaKeyStatus {\n companion object\n}\n\npublic inline val MediaKeyStatus.Companion.USABLE:
MediaKeyStatus get() = \"usable\".asDynamic().unsafeCast<MediaKeyStatus>()\n\npublic inline val
MediaKeyStatus.Companion.EXPIRED: MediaKeyStatus get() =
\"expired\".asDynamic().unsafeCast<MediaKeyStatus>()\n\npublic inline val

```

```

MediaKeyStatus.Companion.RELEASED: MediaKeyStatus get() =
  `released`.asDynamic().unsafeCast<MediaKeyStatus>()\n\npublic inline val
MediaKeyStatus.Companion.OUTPUT_RESTRICTED: MediaKeyStatus get() = `output-
restricted`.asDynamic().unsafeCast<MediaKeyStatus>()\n\npublic inline val
MediaKeyStatus.Companion.OUTPUT_DOWNSCALED: MediaKeyStatus get() = `output-
downscaled`.asDynamic().unsafeCast<MediaKeyStatus>()\n\npublic inline val
MediaKeyStatus.Companion.STATUS_PENDING: MediaKeyStatus get() = `status-
pending`.asDynamic().unsafeCast<MediaKeyStatus>()\n\npublic inline val
MediaKeyStatus.Companion.INTERNAL_ERROR: MediaKeyStatus get() = `internal-
error`.asDynamic().unsafeCast<MediaKeyStatus>()\n\n/* please, don't implement this interface!
*/\n\n@JsName("null")\n\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\n\npublic external
interface MediaKeyMessageType {\n  companion object\n}\n\npublic inline val
MediaKeyMessageType.Companion.LICENSE_REQUEST: MediaKeyMessageType get() = `license-
request`.asDynamic().unsafeCast<MediaKeyMessageType>()\n\npublic inline val
MediaKeyMessageType.Companion.LICENSE_RENEWAL: MediaKeyMessageType get() = `license-
renewal`.asDynamic().unsafeCast<MediaKeyMessageType>()\n\npublic inline val
MediaKeyMessageType.Companion.LICENSE_RELEASE: MediaKeyMessageType get() = `license-
release`.asDynamic().unsafeCast<MediaKeyMessageType>()\n\npublic inline val
MediaKeyMessageType.Companion.INDIVIDUALIZATION_REQUEST: MediaKeyMessageType get() =
`individualization-request`.asDynamic().unsafeCast<MediaKeyMessageType>(), /*\n\n * Copyright 2010-2021
JetBrains s.r.o. and Kotlin Programming Language contributors.\n\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n\n */\n\n// NOTE: THIS FILE IS AUTO-
GENERATED, DO NOT EDIT!\n\n// See github.com/kotlin/dukat for details\n\npackage
org.w3c.dom.events\n\nimport kotlin.js.*\n\nimport org.khronos.webgl.*\n\nimport org.w3c.dom.*\n\n/**\n\n * Exposes
the JavaScript [UIEvent](https://developer.mozilla.org/en/docs/Web/API/UIEvent) to Kotlin\n\n */\n\npublic external
open class UIEvent(type: String, eventInitDict: UIEventInit = definedExternally) : Event {\n  open val view:
Window?\n  open val detail: Int\n\n  companion object {\n    val NONE: Short\n    val
CAPTURING_PHASE: Short\n    val AT_TARGET: Short\n    val BUBBLING_PHASE: Short\n
}\n}\n\npublic external interface UIEventInit : EventInit {\n  var view: Window? /* = null */\n  get() =
definedExternally\n  set(value) = definedExternally\n  var detail: Int? /* = 0 */\n  get() =
definedExternally\n  set(value) = definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n\n@kotlin.internal.InlineOnly\n\npublic inline fun UIEventInit(view: Window? = null,
detail: Int? = 0, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): UIEventInit
{\n  val o = js("{}")\n  o["view"] = view\n  o["detail"] = detail\n  o["bubbles"] = bubbles\n
o["cancelable"] = cancelable\n  o["composed"] = composed\n  return o\n}\n\n/**\n\n * Exposes the JavaScript
[FocusEvent](https://developer.mozilla.org/en/docs/Web/API/FocusEvent) to Kotlin\n\n */\n\npublic external open class
FocusEvent(type: String, eventInitDict: FocusEventInit = definedExternally) : UIEvent {\n  open val relatedTarget:
EventTarget?\n\n  companion object {\n    val NONE: Short\n    val CAPTURING_PHASE: Short\n    val
AT_TARGET: Short\n    val BUBBLING_PHASE: Short\n  }\n}\n\npublic external interface FocusEventInit :
UIEventInit {\n  var relatedTarget: EventTarget? /* = null */\n  get() = definedExternally\n  set(value) =
definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n\n@kotlin.internal.InlineOnly\n\npublic inline fun FocusEventInit(relatedTarget:
EventTarget? = null, view: Window? = null, detail: Int? = 0, bubbles: Boolean? = false, cancelable: Boolean? =
false, composed: Boolean? = false): FocusEventInit {\n  val o = js("{}")\n  o["relatedTarget"] =
relatedTarget\n  o["view"] = view\n  o["detail"] = detail\n  o["bubbles"] = bubbles\n  o["cancelable"] =
cancelable\n  o["composed"] = composed\n  return o\n}\n\n/**\n\n * Exposes the JavaScript
[MouseEvent](https://developer.mozilla.org/en/docs/Web/API/MouseEvent) to Kotlin\n\n */\n\npublic external open
class MouseEvent(type: String, eventInitDict: MouseEventInit = definedExternally) : UIEvent,

```

```

UnionElementOrMouseEvent {\n  open val screenX: Int\n  open val screenY: Int\n  open val clientX: Int\n
open val clientY: Int\n  open val ctrlKey: Boolean\n  open val shiftKey: Boolean\n  open val altKey: Boolean\n
open val metaKey: Boolean\n  open val button: Short\n  open val buttons: Short\n  open val relatedTarget:
EventTarget?\n  open val region: String?\n  open val pageX: Double\n  open val pageY: Double\n  open val x:
Double\n  open val y: Double\n  open val offsetX: Double\n  open val offsetY: Double\n  fun
getModifierState(keyArg: String): Boolean\n\n  companion object {\n    val NONE: Short\n    val
CAPTURING_PHASE: Short\n    val AT_TARGET: Short\n    val BUBBLING_PHASE: Short\n
}\n}\n\npublic external interface MouseEventInit : EventModifierInit {\n  var screenX: Int? /* = 0 */\n    get() =
definedExternally\n    set(value) = definedExternally\n  var screenY: Int? /* = 0 */\n    get() =
definedExternally\n    set(value) = definedExternally\n  var clientX: Int? /* = 0 */\n    get() =
definedExternally\n    set(value) = definedExternally\n  var clientY: Int? /* = 0 */\n    get() =
definedExternally\n    set(value) = definedExternally\n  var button: Short? /* = 0 */\n    get() =
definedExternally\n    set(value) = definedExternally\n  var buttons: Short? /* = 0 */\n    get() =
definedExternally\n    set(value) = definedExternally\n  var relatedTarget: EventTarget? /* = null */\n    get()
= definedExternally\n    set(value) = definedExternally\n  var region: String? /* = null */\n    get() =
definedExternally\n    set(value) = definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun MouseEventInit(screenX: Int? = 0,
screenY: Int? = 0, clientX: Int? = 0, clientY: Int? = 0, button: Short? = 0, buttons: Short? = 0, relatedTarget:
EventTarget? = null, region: String? = null, ctrlKey: Boolean? = false, shiftKey: Boolean? = false, altKey: Boolean?
= false, metaKey: Boolean? = false, modifierAltGraph: Boolean? = false, modifierCapsLock: Boolean? = false,
modifierFn: Boolean? = false, modifierFnLock: Boolean? = false, modifierHyper: Boolean? = false,
modifierNumLock: Boolean? = false, modifierScrollLock: Boolean? = false, modifierSuper: Boolean? = false,
modifierSymbol: Boolean? = false, modifierSymbolLock: Boolean? = false, view: Window? = null, detail: Int? = 0,
bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): MouseEventInit {\n  val o =
js("{}")\n  o["screenX"] = screenX\n  o["screenY"] = screenY\n  o["clientX"] = clientX\n  o["clientY"]
= clientY\n  o["button"] = button\n  o["buttons"] = buttons\n  o["relatedTarget"] = relatedTarget\n
o["region"] = region\n  o["ctrlKey"] = ctrlKey\n  o["shiftKey"] = shiftKey\n  o["altKey"] = altKey\n
o["metaKey"] = metaKey\n  o["modifierAltGraph"] = modifierAltGraph\n  o["modifierCapsLock"] =
modifierCapsLock\n  o["modifierFn"] = modifierFn\n  o["modifierFnLock"] = modifierFnLock\n
o["modifierHyper"] = modifierHyper\n  o["modifierNumLock"] = modifierNumLock\n
o["modifierScrollLock"] = modifierScrollLock\n  o["modifierSuper"] = modifierSuper\n
o["modifierSymbol"] = modifierSymbol\n  o["modifierSymbolLock"] = modifierSymbolLock\n  o["view"] =
view\n  o["detail"] = detail\n  o["bubbles"] = bubbles\n  o["cancelable"] = cancelable\n  o["composed"] =
composed\n  return o\n}\n\npublic external interface EventModifierInit : UIEventInit {\n  var ctrlKey: Boolean?
/* = false */\n    get() = definedExternally\n    set(value) = definedExternally\n  var shiftKey: Boolean? /* =
false */\n    get() = definedExternally\n    set(value) = definedExternally\n  var altKey: Boolean? /* = false
*/\n    get() = definedExternally\n    set(value) = definedExternally\n  var metaKey: Boolean? /* = false */\n
get() = definedExternally\n    set(value) = definedExternally\n  var modifierAltGraph: Boolean? /* = false */\n
get() = definedExternally\n    set(value) = definedExternally\n  var modifierCapsLock: Boolean? /* = false
*/\n    get() = definedExternally\n    set(value) = definedExternally\n  var modifierFn: Boolean? /* = false */\n
get() = definedExternally\n    set(value) = definedExternally\n  var modifierFnLock: Boolean? /* = false */\n
get() = definedExternally\n    set(value) = definedExternally\n  var modifierHyper: Boolean? /* = false */\n
get() = definedExternally\n    set(value) = definedExternally\n  var modifierNumLock: Boolean? /* = false */\n
get() = definedExternally\n    set(value) = definedExternally\n  var modifierScrollLock: Boolean? /* = false
*/\n    get() = definedExternally\n    set(value) = definedExternally\n  var modifierSuper: Boolean? /* = false
*/\n    get() = definedExternally\n    set(value) = definedExternally\n  var modifierSymbol: Boolean? /* = false
*/\n    get() = definedExternally\n    set(value) = definedExternally\n  var modifierSymbolLock: Boolean? /* =
false */\n    get() = definedExternally\n    set(value) =

```

```

definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun EventModifierInit(ctrlKey: Boolean? =
false, shiftKey: Boolean? = false, altKey: Boolean? = false, metaKey: Boolean? = false, modifierAltGraph:
Boolean? = false, modifierCapsLock: Boolean? = false, modifierFn: Boolean? = false, modifierFnLock: Boolean? =
false, modifierHyper: Boolean? = false, modifierNumLock: Boolean? = false, modifierScrollLock: Boolean? = false,
modifierSuper: Boolean? = false, modifierSymbol: Boolean? = false, modifierSymbolLock: Boolean? = false, view:
Window? = null, detail: Int? = 0, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? =
false): EventModifierInit {\n    val o = js(\"({})\")\n    o[\"ctrlKey\"] = ctrlKey\n    o[\"shiftKey\"] = shiftKey\n    o[\"altKey\"] = altKey\n    o[\"metaKey\"] = metaKey\n    o[\"modifierAltGraph\"] = modifierAltGraph\n    o[\"modifierCapsLock\"] = modifierCapsLock\n    o[\"modifierFn\"] = modifierFn\n    o[\"modifierFnLock\"] =
modifierFnLock\n    o[\"modifierHyper\"] = modifierHyper\n    o[\"modifierNumLock\"] = modifierNumLock\n    o[\"modifierScrollLock\"] = modifierScrollLock\n    o[\"modifierSuper\"] = modifierSuper\n    o[\"modifierSymbol\"] = modifierSymbol\n    o[\"modifierSymbolLock\"] = modifierSymbolLock\n    o[\"view\"] =
view\n    o[\"detail\"] = detail\n    o[\"bubbles\"] = bubbles\n    o[\"cancelable\"] = cancelable\n    o[\"composed\"] =
composed\n    return o\n}\n\n/**\n * Exposes the JavaScript
[WheelEvent](https://developer.mozilla.org/en/docs/Web/API/WheelEvent) to Kotlin\n */\npublic external open
class WheelEvent(type: String, eventInitDict: WheelEventInit = definedExternally) : MouseEvent {\n    open val
deltaX: Double\n    open val deltaY: Double\n    open val deltaZ: Double\n    open val deltaMode: Int\n\n    companion object {\n        val DOM_DELTA_PIXEL: Int\n        val DOM_DELTA_LINE: Int\n        val
DOM_DELTA_PAGE: Int\n        val NONE: Short\n        val CAPTURING_PHASE: Short\n        val
AT_TARGET: Short\n        val BUBBLING_PHASE: Short\n    }\n\n    public external interface WheelEventInit :
MouseEventInit {\n        var deltaX: Double? /* = 0.0 */\n            get() = definedExternally\n            set(value) =
definedExternally\n        var deltaY: Double? /* = 0.0 */\n            get() = definedExternally\n            set(value) =
definedExternally\n        var deltaZ: Double? /* = 0.0 */\n            get() = definedExternally\n            set(value) =
definedExternally\n        var deltaMode: Int? /* = 0 */\n            get() = definedExternally\n            set(value) =
definedExternally\n    }\n\n    @Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n    @kotlin.internal.InlineOnly\n    public inline fun WheelEventInit(deltaX: Double? = 0.0,
deltaY: Double? = 0.0, deltaZ: Double? = 0.0, deltaMode: Int? = 0, screenX: Int? = 0, screenY: Int? = 0, clientX:
Int? = 0, clientY: Int? = 0, button: Short? = 0, buttons: Short? = 0, relatedTarget: EventTarget? = null, region:
String? = null, ctrlKey: Boolean? = false, shiftKey: Boolean? = false, altKey: Boolean? = false, metaKey: Boolean?
= false, modifierAltGraph: Boolean? = false, modifierCapsLock: Boolean? = false, modifierFn: Boolean? = false,
modifierFnLock: Boolean? = false, modifierHyper: Boolean? = false, modifierNumLock: Boolean? = false,
modifierScrollLock: Boolean? = false, modifierSuper: Boolean? = false, modifierSymbol: Boolean? = false,
modifierSymbolLock: Boolean? = false, view: Window? = null, detail: Int? = 0, bubbles: Boolean? = false,
cancelable: Boolean? = false, composed: Boolean? = false): WheelEventInit {\n        val o = js(\"({})\")\n        o[\"deltaX\"] = deltaX\n        o[\"deltaY\"] = deltaY\n        o[\"deltaZ\"] = deltaZ\n        o[\"deltaMode\"] = deltaMode\n        o[\"screenX\"] = screenX\n        o[\"screenY\"] = screenY\n        o[\"clientX\"] = clientX\n        o[\"clientY\"] = clientY\n        o[\"button\"] = button\n        o[\"buttons\"] = buttons\n        o[\"relatedTarget\"] = relatedTarget\n        o[\"region\"] =
region\n        o[\"ctrlKey\"] = ctrlKey\n        o[\"shiftKey\"] = shiftKey\n        o[\"altKey\"] = altKey\n        o[\"metaKey\"] =
metaKey\n        o[\"modifierAltGraph\"] = modifierAltGraph\n        o[\"modifierCapsLock\"] = modifierCapsLock\n        o[\"modifierFn\"] = modifierFn\n        o[\"modifierFnLock\"] = modifierFnLock\n        o[\"modifierHyper\"] =
modifierHyper\n        o[\"modifierNumLock\"] = modifierNumLock\n        o[\"modifierScrollLock\"] =
modifierScrollLock\n        o[\"modifierSuper\"] = modifierSuper\n        o[\"modifierSymbol\"] = modifierSymbol\n        o[\"modifierSymbolLock\"] = modifierSymbolLock\n        o[\"view\"] = view\n        o[\"detail\"] = detail\n        o[\"bubbles\"] = bubbles\n        o[\"cancelable\"] = cancelable\n        o[\"composed\"] = composed\n        return
o\n    }\n\n    /**\n     * Exposes the JavaScript [InputEvent](https://developer.mozilla.org/en/docs/Web/API/InputEvent) to
Kotlin\n     */\n    public external open class InputEvent(type: String, eventInitDict: InputEventInit = definedExternally) :
UIEvent {\n        open val data: String\n        open val isComposing: Boolean\n\n        companion object {\n            val NONE:

```

```

Short\n    val CAPTURING_PHASE: Short\n    val AT_TARGET: Short\n    val BUBBLING_PHASE:
Short\n    }\n}\n\npublic external interface InputEventInit : UIEventInit {\n    var data: String? /* = \"\" */\n    get() = definedExternally\n    set(value) = definedExternally\n    var isComposing: Boolean? /* = false */\n    get() = definedExternally\n    set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\n\npublic inline fun InputEventInit(data: String? = \"\",
isComposing: Boolean? = false, view: Window? = null, detail: Int? = 0, bubbles: Boolean? = false, cancelable:
Boolean? = false, composed: Boolean? = false): InputEventInit {\n    val o = js(\"({})\")\n    o[\"data\"] = data\n    o[\"isComposing\"] = isComposing\n    o[\"view\"] = view\n    o[\"detail\"] = detail\n    o[\"bubbles\"] = bubbles\n    o[\"cancelable\"] = cancelable\n    o[\"composed\"] = composed\n    return o\n}\n\n/**\n * Exposes the JavaScript
[KeyboardEvent](https://developer.mozilla.org/en/docs/Web/API/KeyboardEvent) to Kotlin\n */\n\npublic external
open class KeyboardEvent(type: String, eventInitDict: KeyboardEventInit = definedExternally) : UIEvent {\n
    open val key: String\n    open val code: String\n    open val location: Int\n    open val ctrlKey: Boolean\n    open val
    shiftKey: Boolean\n    open val altKey: Boolean\n    open val metaKey: Boolean\n    open val repeat: Boolean\n
    open val isComposing: Boolean\n    open val charCode: Int\n    open val keyCode: Int\n    open val which: Int\n
    fun getModifierState(keyArg: String): Boolean\n\n    companion object {\n        val
        DOM_KEY_LOCATION_STANDARD: Int\n        val DOM_KEY_LOCATION_LEFT: Int\n        val
        DOM_KEY_LOCATION_RIGHT: Int\n        val DOM_KEY_LOCATION_NUMPAD: Int\n        val NONE:
        Short\n        val CAPTURING_PHASE: Short\n        val AT_TARGET: Short\n        val BUBBLING_PHASE:
        Short\n        }\n}\n\npublic external interface KeyboardEventInit : EventModifierInit {\n    var key: String? /* = \"\"
*/\n    get() = definedExternally\n    set(value) = definedExternally\n    var code: String? /* = \"\" */\n    get()
= definedExternally\n    set(value) = definedExternally\n    var location: Int? /* = 0 */\n    get() =
definedExternally\n    set(value) = definedExternally\n    var repeat: Boolean? /* = false */\n    get() =
definedExternally\n    set(value) = definedExternally\n    var isComposing: Boolean? /* = false */\n    get() =
definedExternally\n    set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\n\npublic inline fun KeyboardEventInit(key: String? = \"\",
code: String? = \"\", location: Int? = 0, repeat: Boolean? = false, isComposing: Boolean? = false, ctrlKey: Boolean?
= false, shiftKey: Boolean? = false, altKey: Boolean? = false, metaKey: Boolean? = false, modifierAltGraph:
Boolean? = false, modifierCapsLock: Boolean? = false, modifierFn: Boolean? = false, modifierFnLock: Boolean? =
false, modifierHyper: Boolean? = false, modifierNumLock: Boolean? = false, modifierScrollLock: Boolean? = false,
modifierSuper: Boolean? = false, modifierSymbol: Boolean? = false, modifierSymbolLock: Boolean? = false, view:
Window? = null, detail: Int? = 0, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? =
false): KeyboardEventInit {\n    val o = js(\"({})\")\n    o[\"key\"] = key\n    o[\"code\"] = code\n    o[\"location\"] =
location\n    o[\"repeat\"] = repeat\n    o[\"isComposing\"] = isComposing\n    o[\"ctrlKey\"] = ctrlKey\n
    o[\"shiftKey\"] = shiftKey\n    o[\"altKey\"] = altKey\n    o[\"metaKey\"] = metaKey\n    o[\"modifierAltGraph\"] =
modifierAltGraph\n    o[\"modifierCapsLock\"] = modifierCapsLock\n    o[\"modifierFn\"] = modifierFn\n
    o[\"modifierFnLock\"] = modifierFnLock\n    o[\"modifierHyper\"] = modifierHyper\n    o[\"modifierNumLock\"] =
modifierNumLock\n    o[\"modifierScrollLock\"] = modifierScrollLock\n    o[\"modifierSuper\"] = modifierSuper\n
    o[\"modifierSymbol\"] = modifierSymbol\n    o[\"modifierSymbolLock\"] = modifierSymbolLock\n    o[\"view\"] =
view\n    o[\"detail\"] = detail\n    o[\"bubbles\"] = bubbles\n    o[\"cancelable\"] = cancelable\n    o[\"composed\"] =
composed\n    return o\n}\n\n/**\n * Exposes the JavaScript
[CompositionEvent](https://developer.mozilla.org/en/docs/Web/API/CompositionEvent) to Kotlin\n */\n\npublic
external open class CompositionEvent(type: String, eventInitDict: CompositionEventInit = definedExternally) :
UIEvent {\n    open val data: String\n\n    companion object {\n        val NONE: Short\n        val
        CAPTURING_PHASE: Short\n        val AT_TARGET: Short\n        val BUBBLING_PHASE: Short\n
        }\n}\n\npublic external interface CompositionEventInit : UIEventInit {\n    var data: String? /* = \"\" */\n    get() =
definedExternally\n    set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\n\npublic inline fun CompositionEventInit(data: String? =
\", view: Window? = null, detail: Int? = 0, bubbles: Boolean? = false, cancelable: Boolean? = false, composed:

```

```

Boolean? = false): CompositionEventInit {\n  val o = js("{}")\n  o["data"] = data\n  o["view"] = view\n  o["detail"] = detail\n  o["bubbles"] = bubbles\n  o["cancelable"] = cancelable\n  o["composed"] =
composed\n  return o\n}\n\n/**\n * Exposes the JavaScript
[Event](https://developer.mozilla.org/en/docs/Web/API/Event) to Kotlin\n *\npublic external open class
Event(type: String, eventInitDict: EventInit = definedExternally) {\n  open val type: String\n  open val target:
EventTarget?\n  open val currentTarget: EventTarget?\n  open val eventPhase: Short\n  open val bubbles:
Boolean\n  open val cancelable: Boolean\n  open val defaultPrevented: Boolean\n  open val composed:
Boolean\n  open val isTrusted: Boolean\n  open val timeStamp: Number\n  fun composedPath():
Array<EventTarget>\n  fun stopPropagation()\n  fun stopImmediatePropagation()\n  fun preventDefault()\n
fun initEvent(type: String, bubbles: Boolean, cancelable: Boolean)\n\n  companion object {\n    val NONE:
Short\n    val CAPTURING_PHASE: Short\n    val AT_TARGET: Short\n    val BUBBLING_PHASE:
Short\n  }\n}\n\n/**\n * Exposes the JavaScript
[EventTarget](https://developer.mozilla.org/en/docs/Web/API/EventTarget) to Kotlin\n *\npublic external abstract
class EventTarget {\n  fun addEventListener(type: String, callback: EventListener?, options: dynamic =
definedExternally)\n  fun addEventListener(type: String, callback: ((Event) -> Unit)?, options: dynamic =
definedExternally)\n  fun removeEventListener(type: String, callback: EventListener?, options: dynamic =
definedExternally)\n  fun removeEventListener(type: String, callback: ((Event) -> Unit)?, options: dynamic =
definedExternally)\n  fun dispatchEvent(event: Event): Boolean\n}\n\n/**\n * Exposes the JavaScript
[EventListener](https://developer.mozilla.org/en/docs/Web/API/EventListener) to Kotlin\n *\npublic external
interface EventListener {\n  fun handleEvent(event: Event)\n}\n\n/* Copyright 2010-2021 JetBrains s.r.o. and
Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file.\n *\n\n// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT
EDIT!\n// See github.com/kotlin/dukat for details\n\npackage org.w3c.dom\n\nimport kotlin.js.*\nimport
org.khronos.webgl.*\nimport org.w3c.dom.clipboard.*\nimport org.w3c.dom.css.*\nimport
org.w3c.dom.encryptedmedia.*\nimport org.w3c.dom.events.*\nimport org.w3c.dom.mediacapture.*\nimport
org.w3c.dom.mediasource.*\nimport org.w3c.dom.pointerevents.*\nimport org.w3c.dom.svg.*\nimport
org.w3c.fetch.*\nimport org.w3c.files.*\nimport org.w3c.performance.*\nimport org.w3c.workers.*\nimport
org.w3c.xhr.*\n\npublic external abstract class HTMLAllCollection {\n  open val length: Int\n  fun
item(nameOrIndex: String = definedExternally): UnionElementOrHTMLCollection?\n  fun namedItem(name:
String): UnionElementOrHTMLCollection?\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun
HTMLAllCollection.get(index: Int): Element? =
asDynamic()[index]\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun
HTMLAllCollection.get(name: String): UnionElementOrHTMLCollection? = asDynamic()[name]\n\n/**\n *
Exposes the JavaScript
[HTMLFormControlsCollection](https://developer.mozilla.org/en/docs/Web/API/HTMLFormControlsCollection)
to Kotlin\n *\npublic external abstract class HTMLFormControlsCollection : HTMLCollection\n\n/**\n * Exposes
the JavaScript [RadioNodeList](https://developer.mozilla.org/en/docs/Web/API/RadioNodeList) to Kotlin\n
*\npublic external abstract class RadioNodeList : NodeList, UnionElementOrRadioNodeList {\n  open var value:
String\n}\n\n/**\n * Exposes the JavaScript
[HTMLOptionsCollection](https://developer.mozilla.org/en/docs/Web/API/HTMLOptionsCollection) to Kotlin\n
*\npublic external abstract class HTMLOptionsCollection : HTMLCollection {\n  override var length: Int\n  open
var selectedIndex: Int\n  fun add(element: UnionHTMLOptGroupElementOrHTMLOptionElement, before:
dynamic = definedExternally)\n  fun remove(index: Int)\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun
HTMLOptionsCollection.set(index: Int, option: HTMLOptionElement?) { asDynamic()[index] = option }\n\n/**\n *
Exposes the JavaScript [HTMLInputElement](https://developer.mozilla.org/en/docs/Web/API/HTMLInputElement) to

```

```

Kotlin\n *\npublic external abstract class HTMLInputElement : Element, GlobalEventHandlers,
DocumentAndElementEventHandlers, ElementContentEditable, ElementCSSInlineStyle {\n  open var title:
String\n  open var lang: String\n  open var translate: Boolean\n  open var dir: String\n  open val dataset:
DOMStringMap\n  open var hidden: Boolean\n  open var tabIndex: Int\n  open var accessKey: String\n  open
val accessKeyLabel: String\n  open var draggable: Boolean\n  open val dropzone: DOMTokenList\n  open var
contextMenu: HTMLMenuElement?\n  open var spellcheck: Boolean\n  open var innerText: String\n  open val
offsetParent: Element?\n  open val offsetTop: Int\n  open val offsetLeft: Int\n  open val offsetWidth: Int\n  open
val offsetHeight: Int\n  fun click()\n  fun focus()\n  fun blur()\n  fun forceSpellCheck()\n\n  companion object
{\n    val ELEMENT_NODE: Short\n    val ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n
val CDATA_SECTION_NODE: Short\n    val ENTITY_REFERENCE_NODE: Short\n    val
ENTITY_NODE: Short\n    val PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE:
Short\n    val DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n  }\n}\n\n/**\n * Exposes the JavaScript
[HTMLUnknownElement](https://developer.mozilla.org/en/docs/Web/API/HTMLUnknownElement) to Kotlin\n
*\npublic external abstract class HTMLUnknownElement : HTMLInputElement {\n  companion object {\n    val
ELEMENT_NODE: Short\n    val ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n    val
CDATA_SECTION_NODE: Short\n    val ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE:
Short\n    val PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val
DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n  }\n}\n\n/**\n * Exposes the JavaScript
[DOMStringMap](https://developer.mozilla.org/en/docs/Web/API/DOMStringMap) to Kotlin\n
*\npublic external
abstract class DOMStringMap\n\n@Suppress(\n  "INVISIBLE_REFERENCE",
  "INVISIBLE_MEMBER")\n\n@kotlin.internal.InlineOnly\n\npublic inline operator fun DOMStringMap.get(name:
String): String? = asDynamic()[name]\n\n@Suppress(\n  "INVISIBLE_REFERENCE",
  "INVISIBLE_MEMBER")\n\n@kotlin.internal.InlineOnly\n\npublic inline operator fun DOMStringMap.set(name:
String, value: String) { asDynamic()[name] = value }\n\n}\n\n/**\n * Exposes the JavaScript
[HTMLHtmlElement](https://developer.mozilla.org/en/docs/Web/API/HTMLHtmlElement) to Kotlin\n
*\npublic
external abstract class HTMLHtmlElement : HTMLInputElement {\n  open var version: String\n\n  companion object
{\n    val ELEMENT_NODE: Short\n    val ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n
val CDATA_SECTION_NODE: Short\n    val ENTITY_REFERENCE_NODE: Short\n    val
ENTITY_NODE: Short\n    val PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE:
Short\n    val DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n  }\n}\n\n/**\n * Exposes the JavaScript
[HTMLHeadElement](https://developer.mozilla.org/en/docs/Web/API/HTMLHeadElement) to Kotlin\n
*\npublic
external abstract class HTMLHeadElement : HTMLInputElement {\n  companion object {\n    val
ELEMENT_NODE: Short\n    val ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n    val

```

```

CDATA_SECTION_NODE: Short\n    val ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE:
Short\n    val PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val
DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[HTMLTitleElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTitleElement) to Kotlin\n *\npublic
external abstract class HTMLTitleElement : HTMLElement {\n    open var text: String\n\n    companion object {\n
        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val
CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE:
Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[HTMLBaseElement](https://developer.mozilla.org/en/docs/Web/API/HTMLBaseElement) to Kotlin\n *\npublic
external abstract class HTMLBaseElement : HTMLElement {\n    open var href: String\n    open var target:
String\n\n    companion object {\n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n
        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE:
Short\n        val ENTITY_NODE: Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val
COMMENT_NODE: Short\n        val DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n
        val DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[HTMLLinkElement](https://developer.mozilla.org/en/docs/Web/API/HTMLLinkElement) to Kotlin\n *\npublic
external abstract class HTMLLinkElement : HTMLElement, LinkStyle {\n    open var href: String\n    open var
crossOrigin: String?\n    open var rel: String\n    open var `as`: RequestDestination\n    open val relList:
DOMTokenList\n    open var media: String\n    open var nonce: String\n    open var hreflang: String\n    open var
type: String\n    open val sizes: DOMTokenList\n    open var referrerPolicy: String\n    open var charset: String\n
    open var rev: String\n    open var target: String\n    open var scope: String\n    open var workerType:
WorkerType\n\n    companion object {\n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE:
Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val
ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val
PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[HTMLMetaElement](https://developer.mozilla.org/en/docs/Web/API/HTMLMetaElement) to Kotlin\n *\npublic
external abstract class HTMLMetaElement : HTMLElement {\n    open var name: String\n    open var httpEquiv:

```

```

String\n open var content: String\n open var scheme: String\n\n companion object {\n val
ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLStyleElement](https://developer.mozilla.org/en/docs/Web/API/HTMLStyleElement) to Kotlin\n *\npublic
external abstract class HTMLStyleElement : HTMLElement, LinkStyle {\n open var media: String\n open var
nonce: String\n open var type: String\n\n companion object {\n val ELEMENT_NODE: Short\n val
ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLBodyElement](https://developer.mozilla.org/en/docs/Web/API/HTMLBodyElement) to Kotlin\n *\npublic
external abstract class HTMLBodyElement : HTMLElement, WindowEventHandlers {\n open var text: String\n
open var link: String\n open var vLink: String\n open var aLink: String\n open var bgColor: String\n open
var background: String\n\n companion object {\n val ELEMENT_NODE: Short\n val
ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLHeadingElement](https://developer.mozilla.org/en/docs/Web/API/HTMLHeadingElement) to Kotlin\n
*\npublic external abstract class HTMLHeadingElement : HTMLElement {\n open var align: String\n\n
companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val
TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE:
Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val
COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n
val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLParagraphElement](https://developer.mozilla.org/en/docs/Web/API/HTMLParagraphElement) to Kotlin\n
*\npublic external abstract class HTMLParagraphElement : HTMLElement {\n open var align: String\n\n

```

```

companion object {
    val ELEMENT_NODE: Short
    val ATTRIBUTE_NODE: Short
    val TEXT_NODE: Short
    val CDATA_SECTION_NODE: Short
    val ENTITY_REFERENCE_NODE: Short
    val ENTITY_NODE: Short
    val PROCESSING_INSTRUCTION_NODE: Short
    val COMMENT_NODE: Short
    val DOCUMENT_NODE: Short
    val DOCUMENT_TYPE_NODE: Short
    val DOCUMENT_FRAGMENT_NODE: Short
    val NOTATION_NODE: Short
    val DOCUMENT_POSITION_DISCONNECTED: Short
    val DOCUMENT_POSITION_PRECEDING: Short
    val DOCUMENT_POSITION_FOLLOWING: Short
    val DOCUMENT_POSITION_CONTAINS: Short
    val DOCUMENT_POSITION_CONTAINED_BY: Short
}

DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short } } \n\n/**
 * Exposes the JavaScript [HTMLHRElement](https://developer.mozilla.org/en/docs/Web/API/HTMLHRElement) to Kotlin
 * \npublic external abstract class HTMLHRElement : HTMLElement {
    open var align: String
    open var color: String
    open var noShade: Boolean
    open var size: String
    open var width: String
}

companion object {
    val ELEMENT_NODE: Short
    val ATTRIBUTE_NODE: Short
    val TEXT_NODE: Short
    val CDATA_SECTION_NODE: Short
    val ENTITY_REFERENCE_NODE: Short
    val ENTITY_NODE: Short
    val PROCESSING_INSTRUCTION_NODE: Short
    val COMMENT_NODE: Short
    val DOCUMENT_NODE: Short
    val DOCUMENT_TYPE_NODE: Short
    val DOCUMENT_FRAGMENT_NODE: Short
    val NOTATION_NODE: Short
    val DOCUMENT_POSITION_DISCONNECTED: Short
    val DOCUMENT_POSITION_PRECEDING: Short
    val DOCUMENT_POSITION_FOLLOWING: Short
    val DOCUMENT_POSITION_CONTAINS: Short
    val DOCUMENT_POSITION_CONTAINED_BY: Short
}

DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short } } \n\n/**
 * Exposes the JavaScript [HTMLPreElement](https://developer.mozilla.org/en/docs/Web/API/HTMLPreElement) to Kotlin
 * \npublic external abstract class HTMLPreElement : HTMLElement {
    open var width: Int
}

companion object {
    val ELEMENT_NODE: Short
    val ATTRIBUTE_NODE: Short
    val TEXT_NODE: Short
    val CDATA_SECTION_NODE: Short
    val ENTITY_REFERENCE_NODE: Short
    val ENTITY_NODE: Short
    val PROCESSING_INSTRUCTION_NODE: Short
    val COMMENT_NODE: Short
    val DOCUMENT_NODE: Short
    val DOCUMENT_TYPE_NODE: Short
    val DOCUMENT_FRAGMENT_NODE: Short
    val NOTATION_NODE: Short
    val DOCUMENT_POSITION_DISCONNECTED: Short
    val DOCUMENT_POSITION_PRECEDING: Short
    val DOCUMENT_POSITION_FOLLOWING: Short
    val DOCUMENT_POSITION_CONTAINS: Short
    val DOCUMENT_POSITION_CONTAINED_BY: Short
}

DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short } } \n\n/**
 * Exposes the JavaScript [HTMLQuoteElement](https://developer.mozilla.org/en/docs/Web/API/HTMLQuoteElement) to Kotlin
 * \npublic external abstract class HTMLQuoteElement : HTMLElement {
    open var cite: String
}

companion object {
    val ELEMENT_NODE: Short
    val ATTRIBUTE_NODE: Short
    val TEXT_NODE: Short
    val CDATA_SECTION_NODE: Short
    val ENTITY_REFERENCE_NODE: Short
    val ENTITY_NODE: Short
    val PROCESSING_INSTRUCTION_NODE: Short
    val COMMENT_NODE: Short
    val DOCUMENT_NODE: Short
    val DOCUMENT_TYPE_NODE: Short
    val DOCUMENT_FRAGMENT_NODE: Short
    val NOTATION_NODE: Short
    val DOCUMENT_POSITION_DISCONNECTED: Short
    val DOCUMENT_POSITION_PRECEDING: Short
    val DOCUMENT_POSITION_FOLLOWING: Short
    val DOCUMENT_POSITION_CONTAINS: Short
    val DOCUMENT_POSITION_CONTAINED_BY: Short
}

DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short } } \n\n/**
 * Exposes the JavaScript [HTMLLOListElement](https://developer.mozilla.org/en/docs/Web/API/HTMLLOListElement) to Kotlin
 * \npublic external abstract class HTMLLOListElement : HTMLElement {
    open var reversed: Boolean
    open var start: Int
    open var type: String
    open var compact: Boolean
}

companion object {
    val ELEMENT_NODE: Short
    val ATTRIBUTE_NODE: Short
    val TEXT_NODE: Short
    val CDATA_SECTION_NODE: Short
    val ENTITY_REFERENCE_NODE: Short
    val ENTITY_NODE: Short
}

```

```

Short\n    val PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val
DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[HTMLUListElement](https://developer.mozilla.org/en/docs/Web/API/HTMLUListElement) to Kotlin\n */\npublic
external abstract class HTMLUListElement : HTMLElement {\n    open var compact: Boolean\n    open var type:
String\n    companion object {\n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n
        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE:
Short\n        val ENTITY_NODE: Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val
COMMENT_NODE: Short\n        val DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n
        val DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[HTMLLIElement](https://developer.mozilla.org/en/docs/Web/API/HTMLLIElement) to Kotlin\n */\npublic
external abstract class HTMLLIElement : HTMLElement {\n    open var value: Int\n    open var type: String\n
companion object {\n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val
TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE:
Short\n        val ENTITY_NODE: Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val
COMMENT_NODE: Short\n        val DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n
        val DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[HTMLDListElement](https://developer.mozilla.org/en/docs/Web/API/HTMLDListElement) to Kotlin\n */\npublic
external abstract class HTMLDListElement : HTMLElement {\n    open var compact: Boolean\n\n    companion
object {\n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE:
Short\n        val CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val
ENTITY_NODE: Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE:
Short\n        val DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[HTMLDivElement](https://developer.mozilla.org/en/docs/Web/API/HTMLDivElement) to Kotlin\n */\npublic
external abstract class HTMLDivElement : HTMLElement {\n    open var align: String\n\n    companion object {\n
        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val
CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE:
Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n

```

```

    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[HTMLAnchorElement](https://developer.mozilla.org/en/docs/Web/API/HTMLAnchorElement) to Kotlin\n
*/\npublic external abstract class HTMLAnchorElement : HTMLInputElement, HTMLHyperlinkElementUtils {\n    open
var target: String\n    open var download: String\n    open var ping: String\n    open var rel: String\n    open val
relList: DOMTokenList\n    open var hreflang: String\n    open var type: String\n    open var text: String\n    open
var referrerPolicy: String\n    open var coords: String\n    open var charset: String\n    open var name: String\n
open var rev: String\n    open var shape: String\n\n    companion object {\n        val ELEMENT_NODE: Short\n
val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n
val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val
PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[HTMLDataElement](https://developer.mozilla.org/en/docs/Web/API/HTMLDataElement) to Kotlin\n
*/\npublic external abstract class HTMLDataElement : HTMLInputElement {\n    open var value: String\n\n    companion object {\n
        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val
CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE:
Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[HTMLTimeElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTimeElement) to Kotlin\n
*/\npublic external abstract class HTMLTimeElement : HTMLInputElement {\n    open var dateTime: String\n\n    companion
object {\n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE:
Short\n        val CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val
ENTITY_NODE: Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE:
Short\n        val DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[HTMLSpanElement](https://developer.mozilla.org/en/docs/Web/API/HTMLSpanElement) to Kotlin\n
*/\npublic external abstract class HTMLSpanElement : HTMLInputElement {\n    companion object {\n        val
ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val
CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE:
Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n

```

```

    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[HTMLBRElement](https://developer.mozilla.org/en/docs/Web/API/HTMLBRElement) to Kotlin\n
*\npublic external abstract class HTMLBRElement : HTMLElement {\n    open var clear: String\n
    companion object {\n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n
        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n
        val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n
        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n
        val DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n
        val DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n
        val DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[HTMLHyperlinkElementUtils](https://developer.mozilla.org/en/docs/Web/API/HTMLHyperlinkElementUtils) to
Kotlin\n
*\npublic external interface HTMLHyperlinkElementUtils {\n    var href: String\n    val origin: String\n
    var protocol: String\n    var username: String\n    var password: String\n    var host: String\n
    var hostname: String\n    var port: String\n    var pathname: String\n    var search: String\n
    var hash: String\n}\n\n/**\n * Exposes the JavaScript [HTMLModElement](https://developer.mozilla.org/en/docs/Web/API/HTMLModElement) to
Kotlin\n
*\npublic external abstract class HTMLModElement : HTMLElement {\n    open var cite: String\n
    open var dateTime: String\n
    companion object {\n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n
        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n
        val ENTITY_NODE: Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n
        val COMMENT_NODE: Short\n        val DOCUMENT_NODE: Short\n
        val DOCUMENT_TYPE_NODE: Short\n        val DOCUMENT_FRAGMENT_NODE: Short\n
        val NOTATION_NODE: Short\n        val DOCUMENT_POSITION_DISCONNECTED: Short\n
        val DOCUMENT_POSITION_PRECEDING: Short\n        val DOCUMENT_POSITION_FOLLOWING: Short\n
        val DOCUMENT_POSITION_CONTAINS: Short\n        val DOCUMENT_POSITION_CONTAINED_BY: Short\n
        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[HTMLPictureElement](https://developer.mozilla.org/en/docs/Web/API/HTMLPictureElement) to Kotlin\n
*\npublic external abstract class HTMLPictureElement : HTMLElement {\n    companion object {\n
        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n
        val CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n
        val ENTITY_NODE: Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n
        val COMMENT_NODE: Short\n        val DOCUMENT_NODE: Short\n
        val DOCUMENT_TYPE_NODE: Short\n        val DOCUMENT_FRAGMENT_NODE: Short\n
        val NOTATION_NODE: Short\n        val DOCUMENT_POSITION_DISCONNECTED: Short\n
        val DOCUMENT_POSITION_PRECEDING: Short\n        val DOCUMENT_POSITION_FOLLOWING: Short\n
        val DOCUMENT_POSITION_CONTAINS: Short\n        val DOCUMENT_POSITION_CONTAINED_BY: Short\n
        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[HTMLSourceElement](https://developer.mozilla.org/en/docs/Web/API/HTMLSourceElement) to Kotlin\n
*\npublic external abstract class HTMLSourceElement : HTMLElement {\n    open var src: String\n
    open var type: String\n    open var srcset: String\n    open var sizes: String\n
    open var media: String\n
    companion object {\n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n
        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n
        val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n
        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n
        val DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n
        val

```



```

    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    } \n} \n \n /** \n * Exposes the JavaScript
[HTMLElement](https://developer.mozilla.org/en/docs/Web/API/HTMLElement) to Kotlin \n
*\npublic external abstract class HTMLElement : HTMLElement {\n    open var data: String\n    open var
type: String\n    open var typeMustMatch: Boolean\n    open var name: String\n    open var useMap: String\n    open
val form: HTMLFormElement?\n    open var width: String\n    open var height: String\n    open val
contentDocument: Document?\n    open val contentWindow: Window?\n    open val willValidate: Boolean\n    open
val validity: ValidityState\n    open val validationMessage: String\n    open var align: String\n    open var archive:
String\n    open var code: String\n    open var declare: Boolean\n    open var hspace: Int\n    open var standby:
String\n    open var vspace: Int\n    open var codeBase: String\n    open var codeType: String\n    open var border:
String\n    fun getSVGDocument(): Document?\n    fun checkValidity(): Boolean\n    fun reportValidity():
Boolean\n    fun setCustomValidity(error: String)\n\n    companion object {\n        val ELEMENT_NODE: Short\n
        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n
        val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val
PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    } \n} \n \n /** \n * Exposes the JavaScript
[HTMLParamElement](https://developer.mozilla.org/en/docs/Web/API/HTMLParamElement) to Kotlin \n
*\npublic external abstract class HTMLParamElement : HTMLElement {\n    open var name: String\n    open var
value: String\n    open var type: String\n    open var valueType: String\n\n    companion object {\n        val
ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val
CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE:
Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    } \n} \n \n /** \n * Exposes the JavaScript
[HTMLVideoElement](https://developer.mozilla.org/en/docs/Web/API/HTMLVideoElement) to Kotlin \n
*\npublic external abstract class HTMLVideoElement : HTMLMediaElement, CanvasImageSource, TexImageSource {\n
    open var width: Int\n    open var height: Int\n    open val videoWidth: Int\n    open val videoHeight: Int\n    open var
poster: String\n    open var playsInline: Boolean\n\n    companion object {\n        val NETWORK_EMPTY: Short\n
        val NETWORK_IDLE: Short\n        val NETWORK_LOADING: Short\n        val NETWORK_NO_SOURCE:
Short\n        val HAVE_NOTHING: Short\n        val HAVE_METADATA: Short\n        val
HAVE_CURRENT_DATA: Short\n        val HAVE_FUTURE_DATA: Short\n        val HAVE_ENOUGH_DATA:
Short\n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n
        val CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val
ENTITY_NODE: Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE:
Short\n        val DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n

```

```

    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[HTMLAudioElement](https://developer.mozilla.org/en/docs/Web/API/HTMLAudioElement) to Kotlin\n
*\n\npublic external abstract class HTMLAudioElement : HTMLMediaElement {\n    companion object {\n        val
NETWORK_EMPTY: Short\n        val NETWORK_IDLE: Short\n        val NETWORK_LOADING: Short\n
val NETWORK_NO_SOURCE: Short\n        val HAVE_NOTHING: Short\n        val HAVE_METADATA:
Short\n        val HAVE_CURRENT_DATA: Short\n        val HAVE_FUTURE_DATA: Short\n        val
HAVE_ENOUGH_DATA: Short\n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n
val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE:
Short\n        val ENTITY_NODE: Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val
COMMENT_NODE: Short\n        val DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n
        val DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[HTMLTrackElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTrackElement) to Kotlin\n
*\n\npublic external abstract class HTMLTrackElement : HTMLMediaElement {\n    open var kind: String\n    open var src: String\n
open var srclang: String\n    open var label: String\n    open var default: Boolean\n    open val readyState: Short\n
open val track: TextTrack\n\n    companion object {\n        val NONE: Short\n        val LOADING: Short\n        val
LOADED: Short\n        val ERROR: Short\n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE:
Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val
ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val
PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[HTMLMediaElement](https://developer.mozilla.org/en/docs/Web/API/HTMLMediaElement) to Kotlin\n
*\n\npublic external abstract class HTMLMediaElement : HTMLMediaElement {\n    open val error: MediaError?\n    open
var src: String\n    open var srcObject: MediaProvider?\n    open val currentSrc: String\n    open var crossOrigin:
String?\n    open val networkState: Short\n    open var preload: String\n    open val buffered: TimeRanges\n    open
val readyState: Short\n    open val seeking: Boolean\n    open var currentTime: Double\n    open val duration:
Double\n    open val paused: Boolean\n    open var defaultPlaybackRate: Double\n    open var playbackRate:
Double\n    open val played: TimeRanges\n    open val seekable: TimeRanges\n    open val ended: Boolean\n    open
var autoplay: Boolean\n    open var loop: Boolean\n    open var controls: Boolean\n    open var volume: Double\n
open var muted: Boolean\n    open var defaultMuted: Boolean\n    open val audioTracks: AudioTrackList\n    open
val videoTracks: VideoTrackList\n    open val textTracks: TextTrackList\n    open val mediaKeys: MediaKeys?\n
open var onencrypted: ((Event) -> dynamic)?\n    open var onwaitingforkey: ((Event) -> dynamic)?\n    fun load()\n
fun canPlayType(type: String): CanPlayTypeResult\n    fun fastSeek(time: Double)\n    fun getStartDate():
dynamic\n    fun play(): Promise<Unit>\n    fun pause()\n    fun addTextTrack(kind: TextTrackKind, label: String =
definedExternally, language: String = definedExternally): TextTrack\n    fun setMediaKeys(mediaKeys:
MediaKeys?): Promise<Unit>\n\n    companion object {\n        val NETWORK_EMPTY: Short\n        val
NETWORK_IDLE: Short\n        val NETWORK_LOADING: Short\n        val NETWORK_NO_SOURCE: Short\n
        val HAVE_NOTHING: Short\n        val HAVE_METADATA: Short\n        val HAVE_CURRENT_DATA:
Short\n        val HAVE_FUTURE_DATA: Short\n        val HAVE_ENOUGH_DATA: Short\n        val

```

ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[MediaError](https://developer.mozilla.org/en/docs/Web/API/MediaError) to Kotlin\n */\npublic external abstract
class MediaError {\n open val code: Short\n\n companion object {\n val MEDIA_ERR_ABORTED: Short\n
val MEDIA_ERR_NETWORK: Short\n val MEDIA_ERR_DECODE: Short\n val
MEDIA_ERR_SRC_NOT_SUPPORTED: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[AudioTrackList](https://developer.mozilla.org/en/docs/Web/API/AudioTrackList) to Kotlin\n */\npublic external
abstract class AudioTrackList : EventTarget {\n open val length: Int\n open var onchange: ((Event) ->
dynamic)?\n open var onaddtrack: ((TrackEvent) -> dynamic)?\n open var onremovetrack: ((TrackEvent) ->
dynamic)?\n fun getTrackById(id: String): AudioTrack?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun AudioTrackList.get(index:
Int): AudioTrack? = asDynamic()[index]\n\n/**\n * Exposes the JavaScript
[AudioTrack](https://developer.mozilla.org/en/docs/Web/API/AudioTrack) to Kotlin\n */\npublic external abstract
class AudioTrack : UnionAudioTrackOrTextTrackOrVideoTrack {\n open val id: String\n open val kind:
String\n open val label: String\n open val language: String\n open var enabled: Boolean\n open val
sourceBuffer: SourceBuffer?\n}\n\n/**\n * Exposes the JavaScript
[VideoTrackList](https://developer.mozilla.org/en/docs/Web/API/VideoTrackList) to Kotlin\n */\npublic external
abstract class VideoTrackList : EventTarget {\n open val length: Int\n open val selectedIndex: Int\n open var
onchange: ((Event) -> dynamic)?\n open var onaddtrack: ((TrackEvent) -> dynamic)?\n open var
onremovetrack: ((TrackEvent) -> dynamic)?\n fun getTrackById(id: String):
VideoTrack?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun VideoTrackList.get(index:
Int): VideoTrack? = asDynamic()[index]\n\n/**\n * Exposes the JavaScript
[VideoTrack](https://developer.mozilla.org/en/docs/Web/API/VideoTrack) to Kotlin\n */\npublic external abstract
class VideoTrack : UnionAudioTrackOrTextTrackOrVideoTrack {\n open val id: String\n open val kind:
String\n open val label: String\n open val language: String\n open var selected: Boolean\n open val
sourceBuffer: SourceBuffer?\n}\n\npublic external abstract class TextTrackList : EventTarget {\n open val length:
Int\n open var onchange: ((Event) -> dynamic)?\n open var onaddtrack: ((TrackEvent) -> dynamic)?\n open
var onremovetrack: ((TrackEvent) -> dynamic)?\n fun getTrackById(id: String):
TextTrack?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun TextTrackList.get(index: Int):
TextTrack? = asDynamic()[index]\n\n/**\n * Exposes the JavaScript
[TextTrack](https://developer.mozilla.org/en/docs/Web/API/TextTrack) to Kotlin\n */\npublic external abstract
class TextTrack : EventTarget, UnionAudioTrackOrTextTrackOrVideoTrack {\n open val kind: TextTrackKind\n
open val label: String\n open val language: String\n open val id: String\n open val
inBandMetadataTrackDispatchType: String\n open var mode: TextTrackMode\n open val cues:
TextTrackCueList?\n open val activeCues: TextTrackCueList?\n open var oncuechange: ((Event) ->
dynamic)?\n open val sourceBuffer: SourceBuffer?\n fun addCue(cue: TextTrackCue)\n fun removeCue(cue:
TextTrackCue)\n}\n\npublic external abstract class TextTrackCueList {\n open val length: Int\n fun
getCueById(id: String): TextTrackCue?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun TextTrackCueList.get(index:

Int): `TextTrackCue?` = `asDynamic()[index]`\n\n/**\n * Exposes the JavaScript `[TextTrackCue]`(<https://developer.mozilla.org/en/docs/Web/API/TextTrackCue>) to Kotlin\n *
 \npublic external abstract class `TextTrackCue` : `EventTarget` {\n open val `track`: `TextTrack?`\n open var `id`: `String`\n open var `startTime`: `Double`\n open var `endTime`: `Double`\n open var `pauseOnExit`: `Boolean`\n open var `onenter`: ((`Event`) -> `dynamic`)?\n open var `onexit`: ((`Event`) -> `dynamic`)?\n}\n\n/**\n * Exposes the JavaScript `[TimeRanges]`(<https://developer.mozilla.org/en/docs/Web/API/TimeRanges>) to Kotlin\n *
 \npublic external abstract class `TimeRanges` {\n open val `length`: `Int`\n fun `start(index: Int)`: `Double`\n fun `end(index: Int)`: `Double`\n}\n\n/**\n * Exposes the JavaScript `[TrackEvent]`(<https://developer.mozilla.org/en/docs/Web/API/TrackEvent>) to Kotlin\n *
 \npublic external open class `TrackEvent`(`type`: `String`, `eventInitDict`: `TrackEventInit` = `definedExternally`): `Event` {\n open val `track`: `UnionAudioTrackOrTextTrackOrVideoTrack?`\n\n companion object {\n val `NONE`: `Short`\n val `CAPTURING_PHASE`: `Short`\n val `AT_TARGET`: `Short`\n val `BUBBLING_PHASE`: `Short`\n}\n}\n\npublic external interface `TrackEventInit` : `EventInit` {\n var `track`: `UnionAudioTrackOrTextTrackOrVideoTrack?` /* = null */\n get() = `definedExternally`\n set(value) = `definedExternally`\n}\n\n@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun `TrackEventInit`(`track`: `UnionAudioTrackOrTextTrackOrVideoTrack?` = null, `bubbles`: `Boolean?` = false, `cancelable`: `Boolean?` = false, `composed`: `Boolean?` = false): `TrackEventInit` {\n val o = js("{}")\n o["track"] = track\n o["bubbles"] = bubbles\n o["cancelable"] = cancelable\n o["composed"] = composed\n return o\n}\n\n/**\n * Exposes the JavaScript `[HTMLMapElement]`(<https://developer.mozilla.org/en/docs/Web/API/HTMLMapElement>) to Kotlin\n *
 \npublic external abstract class `HTMLMapElement` : `HTMLElement` {\n open var `name`: `String`\n open val `areas`: `HTMLCollection`\n\n companion object {\n val `ELEMENT_NODE`: `Short`\n val `ATTRIBUTE_NODE`: `Short`\n val `TEXT_NODE`: `Short`\n val `CDATA_SECTION_NODE`: `Short`\n val `ENTITY_REFERENCE_NODE`: `Short`\n val `ENTITY_NODE`: `Short`\n val `PROCESSING_INSTRUCTION_NODE`: `Short`\n val `COMMENT_NODE`: `Short`\n val `DOCUMENT_NODE`: `Short`\n val `DOCUMENT_TYPE_NODE`: `Short`\n val `DOCUMENT_FRAGMENT_NODE`: `Short`\n val `NOTATION_NODE`: `Short`\n val `DOCUMENT_POSITION_DISCONNECTED`: `Short`\n val `DOCUMENT_POSITION_PRECEDING`: `Short`\n val `DOCUMENT_POSITION_FOLLOWING`: `Short`\n val `DOCUMENT_POSITION_CONTAINS`: `Short`\n val `DOCUMENT_POSITION_CONTAINED_BY`: `Short`\n val `DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC`: `Short`\n }\n}\n\n/**\n * Exposes the JavaScript `[HTMLAreaElement]`(<https://developer.mozilla.org/en/docs/Web/API/HTMLAreaElement>) to Kotlin\n *
 \npublic external abstract class `HTMLAreaElement` : `HTMLElement`, `HTMLHyperlinkElementUtils` {\n open var `alt`: `String`\n open var `coords`: `String`\n open var `shape`: `String`\n open var `target`: `String`\n open var `download`: `String`\n open var `ping`: `String`\n open var `rel`: `String`\n open val `relList`: `DOMTokenList`\n open var `referrerPolicy`: `String`\n open var `noHref`: `Boolean`\n\n companion object {\n val `ELEMENT_NODE`: `Short`\n val `ATTRIBUTE_NODE`: `Short`\n val `TEXT_NODE`: `Short`\n val `CDATA_SECTION_NODE`: `Short`\n val `ENTITY_REFERENCE_NODE`: `Short`\n val `ENTITY_NODE`: `Short`\n val `PROCESSING_INSTRUCTION_NODE`: `Short`\n val `COMMENT_NODE`: `Short`\n val `DOCUMENT_NODE`: `Short`\n val `DOCUMENT_TYPE_NODE`: `Short`\n val `DOCUMENT_FRAGMENT_NODE`: `Short`\n val `NOTATION_NODE`: `Short`\n val `DOCUMENT_POSITION_DISCONNECTED`: `Short`\n val `DOCUMENT_POSITION_PRECEDING`: `Short`\n val `DOCUMENT_POSITION_FOLLOWING`: `Short`\n val `DOCUMENT_POSITION_CONTAINS`: `Short`\n val `DOCUMENT_POSITION_CONTAINED_BY`: `Short`\n val `DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC`: `Short`\n }\n}\n\n/**\n * Exposes the JavaScript `[HTMLTableElement]`(<https://developer.mozilla.org/en/docs/Web/API/HTMLTableElement>) to Kotlin\n *
 \npublic external abstract class `HTMLTableElement` : `HTMLElement` {\n open var `caption`: `HTMLTableCaptionElement?`\n open var `tHead`: `HTMLTableSectionElement?`\n open var `tFoot`:


```

DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }n}\n\n/**\n * Exposes the JavaScript
[HTMLTableRowElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTableRowElement) to Kotlin\n
*\npublic external abstract class HTMLTableRowElement : HTMLInputElement {\n    open val rowIndex: Int\n    open
val sectionRowIndex: Int\n    open val cells: HTMLCollection\n    open var align: String\n    open var ch: String\n
open var chOff: String\n    open var vAlign: String\n    open var bgColor: String\n    fun insertCell(index: Int =
definedExternally): HTMLInputElement\n    fun deleteCell(index: Int)\n\n    companion object {\n        val
ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val
CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE:
Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }n}\n\n/**\n * Exposes the JavaScript
[HTMLTableCellElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTableCellElement) to Kotlin\n
*\npublic external abstract class HTMLTableCellElement : HTMLInputElement {\n    open var colSpan: Int\n    open var
rowSpan: Int\n    open var headers: String\n    open val cellIndex: Int\n    open var scope: String\n    open var abbr:
String\n    open var align: String\n    open var axis: String\n    open var height: String\n    open var width: String\n
open var ch: String\n    open var chOff: String\n    open var noWrap: Boolean\n    open var vAlign: String\n    open
var bgColor: String\n\n    companion object {\n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE:
Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val
ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val
PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }n}\n\n/**\n * Exposes the JavaScript
[HTMLFormElement](https://developer.mozilla.org/en/docs/Web/API/HTMLFormElement) to Kotlin\n
*\npublic external abstract class HTMLFormElement : HTMLInputElement {\n    open var acceptCharset: String\n    open var
action: String\n    open var autocomplete: String\n    open var enctype: String\n    open var encoding: String\n    open
var method: String\n    open var name: String\n    open var noValidate: Boolean\n    open var target: String\n    open
val elements: HTMLFormControlsCollection\n    open val length: Int\n    fun submit()\n    fun reset()\n    fun
checkValidity(): Boolean\n    fun reportValidity(): Boolean\n\n    companion object {\n        val ELEMENT_NODE:
Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE:
Short\n        val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val
PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n

```

```

}\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun
HTMLFormElement.get(index: Int): Element? =
asDynamic()[index]\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun
HTMLFormElement.get(name: String): UnionElementOrRadioNodeList? = asDynamic()[name]\n\n/**\n * Exposes
the JavaScript [HTMLLabelElement](https://developer.mozilla.org/en/docs/Web/API/HTMLLabelElement) to
Kotlin\n */\npublic external abstract class HTMLLabelElement : HTMLInputElement {\n    open val form:
HTMLFormElement?\n    open var htmlFor: String\n    open val control: HTMLFormElement?\n\n    companion object
{\n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n
val CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val
ENTITY_NODE: Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE:
Short\n        val DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[HTMLInputElement](https://developer.mozilla.org/en/docs/Web/API/HTMLInputElement) to Kotlin\n */\npublic
external abstract class HTMLInputElement : HTMLFormElement {\n    open var accept: String\n    open var alt: String\n
open var autocomplete: String\n    open var autofocus: Boolean\n    open var defaultChecked: Boolean\n    open var
checked: Boolean\n    open var dirName: String\n    open var disabled: Boolean\n    open val form:
HTMLFormElement?\n    open val files: FileList?\n    open var formAction: String\n    open var formEnctype:
String\n    open var formMethod: String\n    open var formNoValidate: Boolean\n    open var formTarget: String\n
open var height: Int\n    open var indeterminate: Boolean\n    open var inputMode: String\n    open val list:
HTMLFormElement?\n    open var max: String\n    open var maxLength: Int\n    open var min: String\n    open var
minLength: Int\n    open var multiple: Boolean\n    open var name: String\n    open var pattern: String\n    open var
placeholder: String\n    open var readOnly: Boolean\n    open var required: Boolean\n    open var size: Int\n    open
var src: String\n    open var step: String\n    open var type: String\n    open var defaultValue: String\n    open var
value: String\n    open var valueAsDate: dynamic\n    open var valueAsNumber: Double\n    open var width: Int\n
open val willValidate: Boolean\n    open val validity: ValidityState\n    open val validationMessage: String\n    open
val labels: NodeList\n    open var selectionStart: Int?\n    open var selectionEnd: Int?\n    open var
selectionDirection: String?\n    open var align: String\n    open var useMap: String\n    fun stepUp(n: Int =
definedExternally)\n    fun stepDown(n: Int = definedExternally)\n    fun checkValidity(): Boolean\n    fun
reportValidity(): Boolean\n    fun setCustomValidity(error: String)\n    fun select()\n    fun
setRangeText(replacement: String)\n    fun setRangeText(replacement: String, start: Int, end: Int, selectionMode:
SelectionMode = definedExternally)\n    fun setSelectionRange(start: Int, end: Int, direction: String =
definedExternally)\n\n    companion object {\n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE:
Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val
ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val
PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n}\n\n/**\n * Exposes the JavaScript
[HTMLButtonElement](https://developer.mozilla.org/en/docs/Web/API/HTMLButtonElement) to Kotlin\n

```

```

*\npublic external abstract class HTMLButtonElement : HTMLInputElement {\n  open var autofocus: Boolean\n  open var disabled: Boolean\n  open val form: HTMLFormElement?\n  open var formAction: String\n  open var formEnctype: String\n  open var formMethod: String\n  open var formNoValidate: Boolean\n  open var formTarget: String\n  open var name: String\n  open var type: String\n  open var value: String\n  open var menu: HTMLMenuElement?\n  open val willValidate: Boolean\n  open val validity: ValidityState\n  open val validationMessage: String\n  open val labels: NodeList\n  fun checkValidity(): Boolean\n  fun reportValidity(): Boolean\n  fun setCustomValidity(error: String)\n\n  companion object {\n    val ELEMENT_NODE: Short\n    val ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n    val CDATA_SECTION_NODE: Short\n    val ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE: Short\n    val PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n  }\n}\n\n/**\n * Exposes the JavaScript [HTMLSelectElement](https://developer.mozilla.org/en/docs/Web/API/HTMLSelectElement) to Kotlin\n */\n\npublic external abstract class HTMLSelectElement : HTMLInputElement, ItemArrayLike<Element> {\n  open var autocomplete: String\n  open var autofocus: Boolean\n  open var disabled: Boolean\n  open val form: HTMLFormElement?\n  open var multiple: Boolean\n  open var name: String\n  open var required: Boolean\n  open var size: Int\n  open val type: String\n  open val options: HTMLOptionsCollection\n  override var length: Int\n  open val selectedOptions: HTMLCollection\n  open var selectedIndex: Int\n  open var value: String\n  open val willValidate: Boolean\n  open val validity: ValidityState\n  open val validationMessage: String\n  open val labels: NodeList\n  fun namedItem(name: String): HTMLOptionElement?\n  fun add(element: UnionHTMLOptGroupElementOrHTMLOptionElement, before: dynamic = definedExternally)\n  fun remove(index: Int)\n  fun checkValidity(): Boolean\n  fun reportValidity(): Boolean\n  fun setCustomValidity(error: String)\n  override fun item(index: Int): Element?\n\n  companion object {\n    val ELEMENT_NODE: Short\n    val ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n    val CDATA_SECTION_NODE: Short\n    val ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE: Short\n    val PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n  }\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun HTMLSelectElement.get(index: Int): Element? = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun HTMLSelectElement.set(index: Int, option: HTMLOptionElement?) { asDynamic()[index] = option }\n\n/**\n * Exposes the JavaScript [HTMLDataListElement](https://developer.mozilla.org/en/docs/Web/API/HTMLDataListElement) to Kotlin\n */\n\npublic external abstract class HTMLDataListElement : HTMLInputElement {\n  open val options: HTMLCollection\n\n  companion object {\n    val ELEMENT_NODE: Short\n    val ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n    val CDATA_SECTION_NODE: Short\n    val ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE: Short\n  }\n}

```

```

PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val
DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[HTMLOptGroupElement](https://developer.mozilla.org/en/docs/Web/API/HTMLOptGroupElement) to Kotlin\n
*\npublic external abstract class HTMLOptGroupElement : HTMLElement,
UnionHTMLOptGroupElementOrHTMLOptionElement {\n    open var disabled: Boolean\n    open var label:
String\n\n    companion object {\n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n
val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE:
Short\n        val ENTITY_NODE: Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val
COMMENT_NODE: Short\n        val DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n
        val DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n        }\n}\n\n/**\n * Exposes the JavaScript
[HTMLOptionElement](https://developer.mozilla.org/en/docs/Web/API/HTMLOptionElement) to Kotlin\n
*\npublic external abstract class HTMLOptionElement : HTMLElement,
UnionHTMLOptGroupElementOrHTMLOptionElement {\n    open var disabled: Boolean\n    open var form:
HTMLFormElement?\n    open var label: String\n    open var defaultSelected: Boolean\n    open var selected:
Boolean\n    open var value: String\n    open var text: String\n    open val index: Int\n\n    companion object {\n
val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val
CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE:
Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n        }\n}\n\n/**\n * Exposes the JavaScript
[HTMLTextAreaElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTextAreaElement) to Kotlin\n
*\npublic external abstract class HTMLTextAreaElement : HTMLElement {\n    open var autocomplete: String\n
open var autofocus: Boolean\n    open var cols: Int\n    open var dirName: String\n    open var disabled: Boolean\n
open val form: HTMLFormElement?\n    open var inputMode: String\n    open var maxLength: Int\n    open var
minLength: Int\n    open var name: String\n    open var placeholder: String\n    open var readOnly: Boolean\n    open
var required: Boolean\n    open var rows: Int\n    open var wrap: String\n    open val type: String\n    open var
defaultValue: String\n    open var value: String\n    open val textLength: Int\n    open val willValidate: Boolean\n
open val validity: ValidityState\n    open val validationMessage: String\n    open val labels: NodeList\n    open var
selectionStart: Int?\n    open var selectionEnd: Int?\n    open var selectionDirection: String?\n    fun checkValidity():
Boolean\n    fun reportValidity(): Boolean\n    fun setCustomValidity(error: String)\n    fun select()\n    fun
setRangeText(replacement: String)\n    fun setRangeText(replacement: String, start: Int, end: Int, selectionMode:
SelectionMode = definedExternally)\n    fun setSelectionRange(start: Int, end: Int, direction: String =
definedExternally)\n\n    companion object {\n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE:
Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val
ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val

```

PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
 DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
 DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
 DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
 DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n } \n} \n \n /** \n * Exposes the JavaScript
 [HTMLKeygenElement](https://developer.mozilla.org/en/docs/Web/API/HTMLKeygenElement) to Kotlin \n
 * \n public external abstract class HTMLKeygenElement : HTMLInputElement { \n open var autofocus: Boolean \n
 open var challenge: String \n open var disabled: Boolean \n open val form: HTMLFormElement? \n open var
 keytype: String \n open var name: String \n open val type: String \n open val willValidate: Boolean \n open val
 validity: ValidityState \n open val validationMessage: String \n open val labels: NodeList \n fun checkValidity():
 Boolean \n fun reportValidity(): Boolean \n fun setCustomValidity(error: String) \n \n companion object { \n
 val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
 CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
 Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
 DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
 DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
 DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
 DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n } \n} \n \n /** \n * Exposes the JavaScript
 [HTMLOutputElement](https://developer.mozilla.org/en/docs/Web/API/HTMLOutputElement) to Kotlin \n
 * \n public external abstract class HTMLOutputElement : HTMLInputElement { \n open val htmlFor: DOMTokenList \n
 open val form: HTMLFormElement? \n open var name: String \n open val type: String \n open var
 defaultValue: String \n open var value: String \n open val willValidate: Boolean \n open val validity:
 ValidityState \n open val validationMessage: String \n open val labels: NodeList \n fun checkValidity():
 Boolean \n fun reportValidity(): Boolean \n fun setCustomValidity(error: String) \n \n companion object { \n
 val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
 CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
 Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
 DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
 DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
 DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
 DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n } \n} \n \n /** \n * Exposes the JavaScript
 [HTMLProgressElement](https://developer.mozilla.org/en/docs/Web/API/HTMLProgressElement) to Kotlin \n
 * \n public external abstract class HTMLProgressElement : HTMLInputElement { \n open var value: Double \n open
 var max: Double \n open val position: Double \n open val labels: NodeList \n \n companion object { \n val
 ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
 CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
 Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
 DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
 DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
 DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
 val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
 val DOCUMENT_POSITION_CONTAINED_BY: Short\n val


```

companion object {
    val ELEMENT_NODE: Short
    val ATTRIBUTE_NODE: Short
    val TEXT_NODE: Short
    val CDATA_SECTION_NODE: Short
    val ENTITY_REFERENCE_NODE: Short
    val ENTITY_NODE: Short
    val PROCESSING_INSTRUCTION_NODE: Short
    val COMMENT_NODE: Short
    val DOCUMENT_NODE: Short
    val DOCUMENT_TYPE_NODE: Short
    val DOCUMENT_FRAGMENT_NODE: Short
    val NOTATION_NODE: Short
    val DOCUMENT_POSITION_DISCONNECTED: Short
    val DOCUMENT_POSITION_PRECEDING: Short
    val DOCUMENT_POSITION_FOLLOWING: Short
    val DOCUMENT_POSITION_CONTAINS: Short
    val DOCUMENT_POSITION_CONTAINED_BY: Short
}

DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short }
}

public external abstract class HTMLMenuElement : HTMLElement {
    open var type: String
    open var label: String
    open var compact: Boolean

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
    }

    DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short }
}

public external abstract class HTMLMenuItemElement : HTMLElement {
    open var type: String
    open var label: String
    open var icon: String
    open var disabled: Boolean
    open var checked: Boolean
    open var radiogroup: String
    open var default: Boolean

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
    }

    DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short }
}

public external open class RelatedEvent(
    type: String, eventInitDict: RelatedEventInit = definedExternally
) : Event {
    open val relatedTarget: EventTarget?

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }

    public external interface RelatedEventInit : EventInit {
        var relatedTarget: EventTarget? /* = null */
        get() = definedExternally
        set(value) = definedExternally
    }

    @Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")
    @kotlin.internal.InlineOnly
    public inline fun RelatedEventInit(
        relatedTarget: EventTarget? = null, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false
    ): RelatedEventInit {
        val o = js("{}")
        o["relatedTarget"] = relatedTarget
        o["bubbles"] = bubbles
        o["cancelable"] = cancelable
        o["composed"] = composed
        return o
    }

    /** Exposes the JavaScript [HTMLDialogElement](https://developer.mozilla.org/en/docs/Web/API/HTMLDialogElement) to Kotlin
    */
    public external abstract class HTMLDialogElement : HTMLElement {
        open var open: Boolean
        open var returnValue: String

        fun show(anchor: UnionElementOrMouseEvent = definedExternally)
        fun showModal(anchor: UnionElementOrMouseEvent = definedExternally)
        fun close(returnValue: String = definedExternally)

        companion object {
            val ELEMENT_NODE: Short
            val ATTRIBUTE_NODE: Short
            val TEXT_NODE: Short
            val CDATA_SECTION_NODE: Short
            val ENTITY_REFERENCE_NODE: Short
            val ENTITY_NODE: Short
            val PROCESSING_INSTRUCTION_NODE: Short
            val COMMENT_NODE: Short
        }
    }
}

```

```

DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[HTMLScriptElement](https://developer.mozilla.org/en/docs/Web/API/HTMLScriptElement) to Kotlin\n *\npublic
external abstract class HTMLScriptElement : HTMLElement, HTMLOrSVGScriptElement {\n    open var src:
String\n    open var type: String\n    open var charset: String\n    open var async: Boolean\n    open var defer:
Boolean\n    open var crossOrigin: String?\n    open var text: String\n    open var nonce: String\n    open var event:
String\n    open var htmlFor: String\n\n    companion object {\n        val ELEMENT_NODE: Short\n        val
ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val
ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val
PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[HTMLTemplateElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTemplateElement) to Kotlin\n
*\npublic external abstract class HTMLTemplateElement : HTMLElement {\n    open val content:
DocumentFragment\n\n    companion object {\n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE:
Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val
ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val
PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[HTMLSlotElement](https://developer.mozilla.org/en/docs/Web/API/HTMLSlotElement) to Kotlin\n *\npublic
external abstract class HTMLSlotElement : HTMLElement {\n    open var name: String\n    fun
assignedNodes(options: AssignedNodesOptions = definedExternally): Array<Node>\n\n    companion object {\n
        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val
CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE:
Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\npublic external interface
AssignedNodesOptions {\n    var flatten: Boolean? /* = false */\n    get() = definedExternally\n    set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun AssignedNodesOptions(flatten:
Boolean? = false): AssignedNodesOptions {\n    val o = js(\"({})\")\n    o[\"flatten\"] = flatten\n    return

```

```

o\n}\n\n/**\n * Exposes the JavaScript
[HTMLCanvasElement](https://developer.mozilla.org/en/docs/Web/API/HTMLCanvasElement) to Kotlin\n
*\npublic external abstract class HTMLCanvasElement : HTMLDivElement, CanvasImageSource, TexImageSource
{\n  open var width: Int\n  open var height: Int\n  fun getContext(contextId: String, vararg arguments: Any?):
RenderingContext?\n  fun toDataURL(type: String = definedExternally, quality: Any? = definedExternally):
String\n  fun toBlob(_callback: (Blob?) -> Unit, type: String = definedExternally, quality: Any? =
definedExternally)\n\n  companion object {\n    val ELEMENT_NODE: Short\n    val ATTRIBUTE_NODE:
Short\n    val TEXT_NODE: Short\n    val CDATA_SECTION_NODE: Short\n    val
ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE: Short\n    val
PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val
DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n  }\n}\n\npublic external interface
CanvasRenderingContext2DSettings {\n  var alpha: Boolean? /* = true */\n  get() = definedExternally\n
set(value) = definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun
CanvasRenderingContext2DSettings(alpha: Boolean? = true): CanvasRenderingContext2DSettings {\n  val o =
js("{}")\n  o["alpha"] = alpha\n  return o\n}\n\n/**\n * Exposes the JavaScript
[CanvasRenderingContext2D](https://developer.mozilla.org/en/docs/Web/API/CanvasRenderingContext2D) to
Kotlin\n *\npublic external abstract class CanvasRenderingContext2D : CanvasState, CanvasTransform,
CanvasCompositing, CanvasImageSmoothing, CanvasFillStrokeStyles, CanvasShadowStyles, CanvasFilters,
CanvasRect, CanvasDrawPath, CanvasUserInterface, CanvasText, CanvasDrawImage, CanvasHitRegion,
CanvasImageData, CanvasPathDrawingStyles, CanvasTextDrawingStyles, CanvasPath, RenderingContext {\n
open val canvas: HTMLCanvasElement\n}\n\npublic external interface CanvasState {\n  fun save()\n  fun
restore()\n}\n\npublic external interface CanvasTransform {\n  fun scale(x: Double, y: Double)\n  fun
rotate(angle: Double)\n  fun translate(x: Double, y: Double)\n  fun transform(a: Double, b: Double, c: Double, d:
Double, e: Double, f: Double)\n  fun getTransform(): DOMMatrix\n  fun setTransform(a: Double, b: Double, c:
Double, d: Double, e: Double, f: Double)\n  fun setTransform(transform: dynamic = definedExternally)\n  fun
resetTransform()\n}\n\npublic external interface CanvasCompositing {\n  var globalAlpha: Double\n  var
globalCompositeOperation: String\n}\n\npublic external interface CanvasImageSmoothing {\n  var
imageSmoothingEnabled: Boolean\n  var imageSmoothingQuality: ImageSmoothingQuality\n}\n\npublic external
interface CanvasFillStrokeStyles {\n  var strokeStyle: dynamic\n  get() = definedExternally\n  set(value) =
definedExternally\n  var fillStyle: dynamic\n  get() = definedExternally\n  set(value) = definedExternally\n
fun createLinearGradient(x0: Double, y0: Double, x1: Double, y1: Double): CanvasGradient\n  fun
createRadialGradient(x0: Double, y0: Double, r0: Double, x1: Double, y1: Double, r1: Double): CanvasGradient\n
fun createPattern(image: CanvasImageSource, repetition: String): CanvasPattern?\n}\n\npublic external interface
CanvasShadowStyles {\n  var shadowOffsetX: Double\n  var shadowOffsetY: Double\n  var shadowBlur:
Double\n  var shadowColor: String\n}\n\npublic external interface CanvasFilters {\n  var filter:
String\n}\n\npublic external interface CanvasRect {\n  fun clearRect(x: Double, y: Double, w: Double, h:
Double)\n  fun fillRect(x: Double, y: Double, w: Double, h: Double)\n  fun strokeRect(x: Double, y: Double, w:
Double, h: Double)\n}\n\npublic external interface CanvasDrawPath {\n  fun beginPath()\n  fun fill(fillRule:
CanvasFillRule = definedExternally)\n  fun fill(path: Path2D, fillRule: CanvasFillRule = definedExternally)\n
fun stroke()\n  fun stroke(path: Path2D)\n  fun clip(fillRule: CanvasFillRule = definedExternally)\n  fun
clip(path: Path2D, fillRule: CanvasFillRule = definedExternally)\n  fun resetClip()\n  fun isPointInPath(x:
Double, y: Double, fillRule: CanvasFillRule = definedExternally): Boolean\n  fun isPointInPath(path: Path2D, x:

```

```

Double, y: Double, fillRule: CanvasFillRule = definedExternally): Boolean\n fun isPointInStroke(x: Double, y:
Double): Boolean\n fun isPointInStroke(path: Path2D, x: Double, y: Double): Boolean\n}\n\npublic external
interface CanvasUserInterface {\n fun drawFocusIfNeeded(element: Element)\n fun drawFocusIfNeeded(path:
Path2D, element: Element)\n fun scrollPathIntoView()\n fun scrollPathIntoView(path: Path2D)\n}\n\npublic
external interface CanvasText {\n fun fillText(text: String, x: Double, y: Double, maxWidth: Double =
definedExternally)\n fun strokeText(text: String, x: Double, y: Double, maxWidth: Double = definedExternally)\n
fun measureText(text: String): TextMetrics\n}\n\npublic external interface CanvasDrawImage {\n fun
drawImage(image: CanvasImageSource, dx: Double, dy: Double)\n fun drawImage(image: CanvasImageSource,
dx: Double, dy: Double, dw: Double, dh: Double)\n fun drawImage(image: CanvasImageSource, sx: Double, sy:
Double, sw: Double, sh: Double, dx: Double, dy: Double, dw: Double, dh: Double)\n}\n\npublic external interface
CanvasHitRegion {\n fun addHitRegion(options: HitRegionOptions = definedExternally)\n fun
removeHitRegion(id: String)\n fun clearHitRegions()\n}\n\npublic external interface CanvasImageData {\n fun
createImageData(sw: Double, sh: Double): ImageData\n fun createImageData(imagedata: ImageData):
ImageData\n fun getImageData(sx: Double, sy: Double, sw: Double, sh: Double): ImageData\n fun
putImageData(imagedata: ImageData, dx: Double, dy: Double)\n fun putImageData(imagedata: ImageData, dx:
Double, dy: Double, dirtyX: Double, dirtyY: Double, dirtyWidth: Double, dirtyHeight: Double)\n}\n\npublic
external interface CanvasPathDrawingStyles {\n var lineWidth: Double\n var lineCap: CanvasLineCap\n var
lineJoin: CanvasLineJoin\n var miterLimit: Double\n var lineDashOffset: Double\n fun setLineDash(segments:
Array<Double>)\n fun getLineDash(): Array<Double>\n}\n\npublic external interface CanvasTextDrawingStyles
{\n var font: String\n var textAlign: CanvasTextAlign\n var textBaseline: CanvasTextBaseline\n var
direction: CanvasDirection\n}\n\npublic external interface CanvasPath {\n fun closePath()\n fun moveTo(x:
Double, y: Double)\n fun lineTo(x: Double, y: Double)\n fun quadraticCurveTo(cpx: Double, cpy: Double, x:
Double, y: Double)\n fun bezierCurveTo(cp1x: Double, cp1y: Double, cp2x: Double, cp2y: Double, x: Double, y:
Double)\n fun arcTo(x1: Double, y1: Double, x2: Double, y2: Double, radius: Double)\n fun arcTo(x1: Double,
y1: Double, x2: Double, y2: Double, radiusX: Double, radiusY: Double, rotation: Double)\n fun rect(x: Double, y:
Double, w: Double, h: Double)\n fun arc(x: Double, y: Double, radius: Double, startAngle: Double, endAngle:
Double, anticlockwise: Boolean = definedExternally)\n fun ellipse(x: Double, y: Double, radiusX: Double,
radiusY: Double, rotation: Double, startAngle: Double, endAngle: Double, anticlockwise: Boolean =
definedExternally)\n}\n\n/**\n * Exposes the JavaScript
[CanvasGradient](https://developer.mozilla.org/en/docs/Web/API/CanvasGradient) to Kotlin\n */\n\npublic external
abstract class CanvasGradient {\n fun addColorStop(offset: Double, color: String)\n}\n\n/**\n * Exposes the
JavaScript [CanvasPattern](https://developer.mozilla.org/en/docs/Web/API/CanvasPattern) to Kotlin\n */\n\npublic
external abstract class CanvasPattern {\n fun setTransform(transform: dynamic = definedExternally)\n}\n\n/**\n *
Exposes the JavaScript [TextMetrics](https://developer.mozilla.org/en/docs/Web/API/TextMetrics) to Kotlin\n
*/\n\npublic external abstract class TextMetrics {\n open val width: Double\n open val actualBoundingBoxLeft:
Double\n open val actualBoundingBoxRight: Double\n open val fontBoundingBoxAscent: Double\n open val
fontBoundingBoxDescent: Double\n open val actualBoundingBoxAscent: Double\n open val
actualBoundingBoxDescent: Double\n open val emHeightAscent: Double\n open val emHeightDescent:
Double\n open val hangingBaseline: Double\n open val alphabeticBaseline: Double\n open val
ideographicBaseline: Double\n}\n\npublic external interface HitRegionOptions {\n var path: Path2D? /* = null
*/\n fun get() = definedExternally\n fun set(value) = definedExternally\n var fillRule: CanvasFillRule? /* =
CanvasFillRule.NONZERO */\n fun get() = definedExternally\n fun set(value) = definedExternally\n var id:
String? /* = \"\" */\n fun get() = definedExternally\n fun set(value) = definedExternally\n var parentID: String? /*
= null */\n fun get() = definedExternally\n fun set(value) = definedExternally\n var cursor: String? /* = \"inherit\"
*/\n fun get() = definedExternally\n fun set(value) = definedExternally\n var control: Element? /* = null */\n
fun get() = definedExternally\n fun set(value) = definedExternally\n var label: String? /* = null */\n fun get() =
definedExternally\n fun set(value) = definedExternally\n var role: String? /* = null */\n fun get() =
definedExternally\n fun set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",

```

```

\ "INVISIBLE_MEMBER" )\n@kotlin.internal.InlineOnly\npublic inline fun HitRegionOptions(path: Path2D? =
null, fillRule: CanvasFillRule? = CanvasFillRule.NONZERO, id: String? = "", parentID: String? = null, cursor:
String? = "inherit", control: Element? = null, label: String? = null, role: String? = null): HitRegionOptions {
    val o = js("{}")\n    o["path"] = path\n    o["fillRule"] = fillRule\n    o["id"] = id\n    o["parentID"] = parentID\n
    o["cursor"] = cursor\n    o["control"] = control\n    o["label"] = label\n    o["role"] = role\n    return
o}\n\n/**\n * Exposes the JavaScript [ImageData](https://developer.mozilla.org/en/docs/Web/API/ImageData) to
Kotlin\n *\npublic external open class ImageData : ImageBitmapSource, TexImageSource {\n    constructor(sw:
Int, sh: Int)\n    constructor(data: Uint8ClampedArray, sw: Int, sh: Int = definedExternally)\n    open val width: Int\n
open val height: Int\n    open val data: Uint8ClampedArray\n}\n\n/**\n * Exposes the JavaScript
[Path2D](https://developer.mozilla.org/en/docs/Web/API/Path2D) to Kotlin\n *\npublic external open class
Path2D() : CanvasPath {\n    constructor(path: Path2D)\n    constructor(paths: Array<Path2D>, fillRule:
CanvasFillRule = definedExternally)\n    constructor(d: String)\n    fun addPath(path: Path2D, transform: dynamic =
definedExternally)\n    override fun closePath()\n    override fun moveTo(x: Double, y: Double)\n    override fun
lineTo(x: Double, y: Double)\n    override fun quadraticCurveTo(cpx: Double, cpy: Double, x: Double, y: Double)\n
    override fun bezierCurveTo(cp1x: Double, cp1y: Double, cp2x: Double, cp2y: Double, x: Double, y: Double)\n
    override fun arcTo(x1: Double, y1: Double, x2: Double, y2: Double, radius: Double)\n    override fun arcTo(x1:
Double, y1: Double, x2: Double, y2: Double, radiusX: Double, radiusY: Double, rotation: Double)\n    override fun
rect(x: Double, y: Double, w: Double, h: Double)\n    override fun arc(x: Double, y: Double, radius: Double,
startAngle: Double, endAngle: Double, anticlockwise: Boolean /* = definedExternally */) \n    override fun ellipse(x:
Double, y: Double, radiusX: Double, radiusY: Double, rotation: Double, startAngle: Double, endAngle: Double,
anticlockwise: Boolean /* = definedExternally */) \n}\n\n/**\n * Exposes the JavaScript
[ImageBitmapRenderingContext](https://developer.mozilla.org/en/docs/Web/API/ImageBitmapRenderingContext)
to Kotlin\n *\npublic external abstract class ImageBitmapRenderingContext {\n    open val canvas:
HTMLCanvasElement\n    fun transferFromImageBitmap(bitmap: ImageBitmap?)\n}\n\npublic external interface
ImageBitmapRenderingContextSettings {\n    var alpha: Boolean? /* = true */\n    get() = definedExternally\n
set(value) = definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun
ImageBitmapRenderingContextSettings(alpha: Boolean? = true): ImageBitmapRenderingContextSettings {\n    val o
= js("{}")\n    o["alpha"] = alpha\n    return o}\n\n/**\n * Exposes the JavaScript
[CustomElementRegistry](https://developer.mozilla.org/en/docs/Web/API/CustomElementRegistry) to Kotlin\n
*\npublic external abstract class CustomElementRegistry {\n    fun define(name: String, constructor: () -> dynamic,
options: ElementDefinitionOptions = definedExternally)\n    fun get(name: String): Any?\n    fun
whenDefined(name: String): Promise<Unit>\n}\n\npublic external interface ElementDefinitionOptions {\n    var
extends: String?\n    get() = definedExternally\n    set(value) =
definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun ElementDefinitionOptions(extends:
String? = undefined): ElementDefinitionOptions {\n    val o = js("{}")\n    o["extends"] = extends\n    return
o}\n\npublic external interface ElementContentEditable {\n    var contentEditable: String\n    val
isContentEditable: Boolean\n}\n\n/**\n * Exposes the JavaScript
[DataTransfer](https://developer.mozilla.org/en/docs/Web/API/DataTransfer) to Kotlin\n *\npublic external
abstract class DataTransfer {\n    open var dropEffect: String\n    open var effectAllowed: String\n    open val items:
DataTransferItemList\n    open val types: Array<out String>\n    open val files: FileList\n    fun
setDragImage(image: Element, x: Int, y: Int)\n    fun getData(format: String): String\n    fun setData(format: String,
data: String)\n    fun clearData(format: String = definedExternally)\n}\n\n/**\n * Exposes the JavaScript
[DataTransferItemList](https://developer.mozilla.org/en/docs/Web/API/DataTransferItemList) to Kotlin\n *\npublic
external abstract class DataTransferItemList {\n    open val length: Int\n    fun add(data: String, type: String):
DataTransferItem?\n    fun add(data: File): DataTransferItem?\n    fun remove(index: Int)\n    fun
clear()\n}\n\n@Suppress("INVISIBLE_REFERENCE",

```

```

\ "INVISIBLE_MEMBER" ) \n @kotlin.internal.InlineOnly \n public inline operator fun
DataTransferItemList.get(index: Int): DataTransferItem? = asDynamic()[index] \n \n /** \n * Exposes the JavaScript
[DataTransferItem](https://developer.mozilla.org/en/docs/Web/API/DataTransferItem) to Kotlin \n */ \n public
external abstract class DataTransferItem { \n     open val kind: String \n     open val type: String \n     fun
getAsString(_callback: ((String) -> Unit)?) \n     fun getAsFile(): File? \n } \n \n /** \n * Exposes the JavaScript
[DragEvent](https://developer.mozilla.org/en/docs/Web/API/DragEvent) to Kotlin \n */ \n public external open class
DragEvent(type: String, eventInitDict: DragEventInit = definedExternally) : MouseEvent { \n     open val
dataTransfer: DataTransfer? \n \n     companion object { \n         val NONE: Short \n         val CAPTURING_PHASE:
Short \n         val AT_TARGET: Short \n         val BUBBLING_PHASE: Short \n     } \n } \n \n public external interface
DragEventInit : MouseEventInit { \n     var dataTransfer: DataTransfer? /* = null */ \n     get() = definedExternally \n
     set(value) = definedExternally \n } \n \n @Suppress("INVISIBLE_REFERENCE",
\ "INVISIBLE_MEMBER" ) \n @kotlin.internal.InlineOnly \n public inline fun DragEventInit(dataTransfer:
DataTransfer? = null, screenX: Int? = 0, screenY: Int? = 0, clientX: Int? = 0, clientY: Int? = 0, button: Short? = 0,
buttons: Short? = 0, relatedTarget: EventTarget? = null, region: String? = null, ctrlKey: Boolean? = false, shiftKey:
Boolean? = false, altKey: Boolean? = false, metaKey: Boolean? = false, modifierAltGraph: Boolean? = false,
modifierCapsLock: Boolean? = false, modifierFn: Boolean? = false, modifierFnLock: Boolean? = false,
modifierHyper: Boolean? = false, modifierNumLock: Boolean? = false, modifierScrollLock: Boolean? = false,
modifierSuper: Boolean? = false, modifierSymbol: Boolean? = false, modifierSymbolLock: Boolean? = false, view:
Window? = null, detail: Int? = 0, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? =
false): DragEventInit { \n     val o = js("{}") \n     o["dataTransfer"] = dataTransfer \n     o["screenX"] = screenX \n
     o["screenY"] = screenY \n     o["clientX"] = clientX \n     o["clientY"] = clientY \n     o["button"] = button \n
     o["buttons"] = buttons \n     o["relatedTarget"] = relatedTarget \n     o["region"] = region \n     o["ctrlKey"] =
ctrlKey \n     o["shiftKey"] = shiftKey \n     o["altKey"] = altKey \n     o["metaKey"] = metaKey \n
     o["modifierAltGraph"] = modifierAltGraph \n     o["modifierCapsLock"] = modifierCapsLock \n
     o["modifierFn"] = modifierFn \n     o["modifierFnLock"] = modifierFnLock \n     o["modifierHyper"] =
modifierHyper \n     o["modifierNumLock"] = modifierNumLock \n     o["modifierScrollLock"] =
modifierScrollLock \n     o["modifierSuper"] = modifierSuper \n     o["modifierSymbol"] = modifierSymbol \n
     o["modifierSymbolLock"] = modifierSymbolLock \n     o["view"] = view \n     o["detail"] = detail \n
     o["bubbles"] = bubbles \n     o["cancelable"] = cancelable \n     o["composed"] = composed \n     return
o \n } \n \n /** \n * Exposes the JavaScript [Window](https://developer.mozilla.org/en/docs/Web/API/Window) to
Kotlin \n */ \n public external abstract class Window : EventTarget, GlobalEventHandlers, WindowEventHandlers,
WindowOrWorkerGlobalScope, WindowSessionStorage, WindowLocalStorage, GlobalPerformance,
UnionMessagePortOrWindowProxy { \n     open val window: Window \n     open val self: Window \n     open val
document: Document \n     open var name: String \n     open val location: Location \n     open val history: History \n
     open val customElements: CustomElementRegistry \n     open val locationbar: BarProp \n     open val menubar:
BarProp \n     open val personalbar: BarProp \n     open val scrollbars: BarProp \n     open val statusbar: BarProp \n
     open val toolbar: BarProp \n     open var status: String \n     open val closed: Boolean \n     open val frames: Window \n
     open val length: Int \n     open val top: Window \n     open var opener: Any? \n     open val parent: Window \n
     open val frameElement: Element? \n     open val navigator: Navigator \n     open val applicationCache: ApplicationCache \n
     open val external: External \n     open val screen: Screen \n     open val innerWidth: Int \n     open val innerHeight: Int \n
     open val scrollX: Double \n     open val pageXOffset: Double \n     open val scrollY: Double \n     open val
pageYOffset: Double \n     open val screenX: Int \n     open val screenY: Int \n     open val outerWidth: Int \n     open val
outerHeight: Int \n     open val devicePixelRatio: Double \n     fun close() \n     fun stop() \n     fun focus() \n     fun blur() \n
     fun open(url: String = definedExternally, target: String = definedExternally, features: String = definedExternally):
Window? \n     fun alert() \n     fun alert(message: String) \n     fun confirm(message: String = definedExternally):
Boolean \n     fun prompt(message: String = definedExternally, default: String = definedExternally): String? \n     fun
print() \n     fun requestAnimationFrame(callback: (Double) -> Unit): Int \n     fun cancelAnimationFrame(handle:
Int) \n     fun postMessage(message: Any?, targetOrigin: String, transfer: Array<dynamic> = definedExternally) \n

```

```

fun captureEvents()\n fun releaseEvents()\n fun matchMedia(query: String): MediaQueryList\n fun moveTo(x:
Int, y: Int)\n fun moveBy(x: Int, y: Int)\n fun resizeTo(x: Int, y: Int)\n fun resizeBy(x: Int, y: Int)\n fun
scroll(options: ScrollToOptions = definedExternally)\n fun scroll(x: Double, y: Double)\n fun scrollTo(options:
ScrollToOptions = definedExternally)\n fun scrollTo(x: Double, y: Double)\n fun scrollBy(options:
ScrollToOptions = definedExternally)\n fun scrollBy(x: Double, y: Double)\n fun getComputedStyle(elt:
Element, pseudoElt: String? = definedExternally):
CSSStyleDeclaration\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Window.get(name: String):
dynamic = asDynamic()[name]\n\npublic external abstract class BarProp {\n open val visible: Boolean\n}\n\n/**\n * Exposes the JavaScript [History](https://developer.mozilla.org/en/docs/Web/API/History) to Kotlin\n */\n\npublic
external abstract class History {\n open val length: Int\n open var scrollRestoration: ScrollRestoration\n open
val state: Any?\n fun go(delta: Int = definedExternally)\n fun back()\n fun forward()\n fun pushState(data:
Any?, title: String, url: String? = definedExternally)\n fun replaceState(data: Any?, title: String, url: String? =
definedExternally)\n}\n\n/**\n * Exposes the JavaScript
[Location](https://developer.mozilla.org/en/docs/Web/API/Location) to Kotlin\n */\n\npublic external abstract class
Location {\n open var href: String\n open val origin: String\n open var protocol: String\n open var host:
String\n open var hostname: String\n open var port: String\n open var pathname: String\n open var search:
String\n open var hash: String\n open val ancestorOrigins: Array<out String>\n fun assign(url: String)\n fun
replace(url: String)\n fun reload()\n}\n\n/**\n * Exposes the JavaScript
[PopStateEvent](https://developer.mozilla.org/en/docs/Web/API/PopStateEvent) to Kotlin\n */\n\npublic external
open class PopStateEvent(type: String, eventInitDict: PopStateEventInit = definedExternally) : Event {\n open val
state: Any?\n\n companion object {\n val NONE: Short\n val CAPTURING_PHASE: Short\n val
AT_TARGET: Short\n val BUBBLING_PHASE: Short\n }\n}\n\npublic external interface PopStateEventInit
: EventInit {\n var state: Any? /* = null */\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun PopStateEventInit(state: Any? = null,
bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): PopStateEventInit {\n val o
= js(\"({})\")\n o[\"state\"] = state\n o[\"bubbles\"] = bubbles\n o[\"cancelable\"] = cancelable\n
o[\"composed\"] = composed\n return o\n}\n\n/**\n * Exposes the JavaScript
[HashChangeEvent](https://developer.mozilla.org/en/docs/Web/API/HashChangeEvent) to Kotlin\n */\n\npublic
external open class HashChangeEvent(type: String, eventInitDict: HashChangeEventInit = definedExternally) :
Event {\n open val oldURL: String\n open val newURL: String\n\n companion object {\n val NONE:
Short\n val CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val BUBBLING_PHASE:
Short\n }\n}\n\npublic external interface HashChangeEventInit : EventInit {\n var oldURL: String? /* = \"\" */\n
get() = definedExternally\n set(value) = definedExternally\n var newURL: String? /* = \"\" */\n get() =
definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun HashChangeEventInit(oldURL:
String? = \"\", newURL: String? = \"\", bubbles: Boolean? = false, cancelable: Boolean? = false, composed:
Boolean? = false): HashChangeEventInit {\n val o = js(\"({})\")\n o[\"oldURL\"] = oldURL\n o[\"newURL\"]
= newURL\n o[\"bubbles\"] = bubbles\n o[\"cancelable\"] = cancelable\n o[\"composed\"] = composed\n
return o\n}\n\n/**\n * Exposes the JavaScript
[PageTransitionEvent](https://developer.mozilla.org/en/docs/Web/API/PageTransitionEvent) to Kotlin\n */\n\npublic
external open class PageTransitionEvent(type: String, eventInitDict: PageTransitionEventInit = definedExternally) :
Event {\n open val persisted: Boolean\n\n companion object {\n val NONE: Short\n val
CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val BUBBLING_PHASE: Short\n
}\n}\n\npublic external interface PageTransitionEventInit : EventInit {\n var persisted: Boolean? /* = false */\n
get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun PageTransitionEventInit(persisted:

```

Boolean? = false, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false):

PageTransitionEventInit {
 val o = js("{}")
 o["persisted"] = persisted
 o["bubbles"] = bubbles
 o["cancelable"] = cancelable
 o["composed"] = composed
 return o
} // Exposes the JavaScript [BeforeUnloadEvent](https://developer.mozilla.org/en/docs/Web/API/BeforeUnloadEvent) to Kotlin
*^public external open class BeforeUnloadEvent : Event {
 var returnValue: String
 companion object {
 val NONE: Short
 val CAPTURING_PHASE: Short
 val AT_TARGET: Short
 val BUBBLING_PHASE: Short
 }
} // public external abstract class ApplicationCache : EventTarget {
 open val status: Short
 open var onchecking: ((Event) -> dynamic)?
 open var onerror: ((Event) -> dynamic)?
 open var onnoupdate: ((Event) -> dynamic)?
 open var ondownloading: ((Event) -> dynamic)?
 open var onprogress: ((ProgressEvent) -> dynamic)?
 open var onupdateready: ((Event) -> dynamic)?
 open var oncached: ((Event) -> dynamic)?
 open var onobsolete: ((Event) -> dynamic)?
 fun update()
 fun abort()
 fun swapCache()
 companion object {
 val UNCACHED: Short
 val IDLE: Short
 val CHECKING: Short
 val DOWNLOADING: Short
 val UPDATEREADY: Short
 val OBSOLETE: Short
 }
} // Exposes the JavaScript [NavigatorOnLine](https://developer.mozilla.org/en/docs/Web/API/NavigatorOnLine) to Kotlin
*^public external interface NavigatorOnLine {
 val onLine: Boolean
} // Exposes the JavaScript [ErrorEvent](https://developer.mozilla.org/en/docs/Web/API/ErrorEvent) to Kotlin
*^public external open class ErrorEvent(type: String, eventInitDict: ErrorEventInit = definedExternally) : Event {
 open val message: String
 open val filename: String
 open val lineno: Int
 open val colno: Int
 open val error: Any?
 companion object {
 val NONE: Short
 val CAPTURING_PHASE: Short
 val AT_TARGET: Short
 val BUBBLING_PHASE: Short
 }
} // public external interface ErrorEventInit : EventInit {
 var message: String? /* = "" */
 get() = definedExternally
 set(value) = definedExternally
 var filename: String? /* = "" */
 get() = definedExternally
 set(value) = definedExternally
 var lineno: Int? /* = 0 */
 get() = definedExternally
 set(value) = definedExternally
 var colno: Int? /* = 0 */
 get() = definedExternally
 set(value) = definedExternally
 var error: Any? /* = null */
 get() = definedExternally
 set(value) = definedExternally
} // @Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline fun ErrorEventInit(message: String? = "", filename: String? = "", lineno: Int? = 0, colno: Int? = 0, error: Any? = null, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): ErrorEventInit {
 val o = js("{}")
 o["message"] = message
 o["filename"] = filename
 o["lineno"] = lineno
 o["colno"] = colno
 o["error"] = error
 o["bubbles"] = bubbles
 o["cancelable"] = cancelable
 o["composed"] = composed
 return o
} // Exposes the JavaScript [PromiseRejectionEvent](https://developer.mozilla.org/en/docs/Web/API/PromiseRejectionEvent) to Kotlin
*^public external open class PromiseRejectionEvent(type: String, eventInitDict: PromiseRejectionEventInit) : Event {
 open val promise: Promise<Any?>
 open val reason: Any?
 companion object {
 val NONE: Short
 val CAPTURING_PHASE: Short
 val AT_TARGET: Short
 val BUBBLING_PHASE: Short
 }
} // public external interface PromiseRejectionEventInit : EventInit {
 var promise: Promise<Any?>
 var reason: Any?
 get() = definedExternally
 set(value) = definedExternally
} // @Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline fun PromiseRejectionEventInit(promise: Promise<Any?>, reason: Any? = undefined, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): PromiseRejectionEventInit {
 val o = js("{}")
 o["promise"] = promise
 o["reason"] = reason
 o["bubbles"] = bubbles
 o["cancelable"] = cancelable
 o["composed"] = composed
 return o
} // Exposes the JavaScript [GlobalEventHandlers](https://developer.mozilla.org/en/docs/Web/API/GlobalEventHandlers) to Kotlin
*^public external interface GlobalEventHandlers {
 var onabort: ((Event) -> dynamic)?
 get() = definedExternally
 set(value) = definedExternally
 var onblur: ((FocusEvent) -> dynamic)?
 get() = definedExternally
 set(value) = definedExternally
 var oncancel: ((Event) -> dynamic)?
 get() =

```

definedExternally\n      set(value) = definedExternally\n      var oncanplay: ((Event) -> dynamic)?\n      get() =
definedExternally\n      set(value) = definedExternally\n      var oncanplaythrough: ((Event) -> dynamic)?\n      get()
= definedExternally\n      set(value) = definedExternally\n      var onchange: ((Event) -> dynamic)?\n      get() =
definedExternally\n      set(value) = definedExternally\n      var onclick: ((MouseEvent) -> dynamic)?\n      get() =
definedExternally\n      set(value) = definedExternally\n      var onclose: ((Event) -> dynamic)?\n      get() =
definedExternally\n      set(value) = definedExternally\n      var oncontextmenu: ((MouseEvent) -> dynamic)?\n
get() = definedExternally\n      set(value) = definedExternally\n      var oncuechange: ((Event) -> dynamic)?\n
get() = definedExternally\n      set(value) = definedExternally\n      var ondblclick: ((MouseEvent) -> dynamic)?\n
get() = definedExternally\n      set(value) = definedExternally\n      var ondrag: ((DragEvent) -> dynamic)?\n
get() = definedExternally\n      set(value) = definedExternally\n      var ondragend: ((DragEvent) -> dynamic)?\n
get() = definedExternally\n      set(value) = definedExternally\n      var ondragenter: ((DragEvent) -> dynamic)?\n
get() = definedExternally\n      set(value) = definedExternally\n      var ondragexit: ((DragEvent) -> dynamic)?\n
get() = definedExternally\n      set(value) = definedExternally\n      var ondragleave: ((DragEvent) -> dynamic)?\n
get() = definedExternally\n      set(value) = definedExternally\n      var ondragover: ((DragEvent) -> dynamic)?\n
get() = definedExternally\n      set(value) = definedExternally\n      var ondragstart: ((DragEvent) -> dynamic)?\n
get() = definedExternally\n      set(value) = definedExternally\n      var ondrop: ((DragEvent) -> dynamic)?\n
get() = definedExternally\n      set(value) = definedExternally\n      var ondurationchange: ((Event) -> dynamic)?\n
get() = definedExternally\n      set(value) = definedExternally\n      var onemptied: ((Event) -> dynamic)?\n
get() = definedExternally\n      set(value) = definedExternally\n      var onended: ((Event) -> dynamic)?\n      get() =
definedExternally\n      set(value) = definedExternally\n      var onerror: ((dynamic, String, Int, Int, Any?) ->
dynamic)?\n      get() = definedExternally\n      set(value) = definedExternally\n      var onfocus: ((FocusEvent) ->
dynamic)?\n      get() = definedExternally\n      set(value) = definedExternally\n      var oninput: ((InputEvent) ->
dynamic)?\n      get() = definedExternally\n      set(value) = definedExternally\n      var oninvalid: ((Event) ->
dynamic)?\n      get() = definedExternally\n      set(value) = definedExternally\n      var onkeydown:
((KeyboardEvent) -> dynamic)?\n      get() = definedExternally\n      set(value) = definedExternally\n      var
onkeypress: ((KeyboardEvent) -> dynamic)?\n      get() = definedExternally\n      set(value) = definedExternally\n
var onkeyup: ((KeyboardEvent) -> dynamic)?\n      get() = definedExternally\n      set(value) =
definedExternally\n      var onload: ((Event) -> dynamic)?\n      get() = definedExternally\n      set(value) =
definedExternally\n      var onloadeddata: ((Event) -> dynamic)?\n      get() = definedExternally\n      set(value) =
definedExternally\n      var onloadedmetadata: ((Event) -> dynamic)?\n      get() = definedExternally\n
set(value) = definedExternally\n      var onloadend: ((Event) -> dynamic)?\n      get() = definedExternally\n
set(value) = definedExternally\n      var onloadstart: ((ProgressEvent) -> dynamic)?\n      get() = definedExternally\n
set(value) = definedExternally\n      var onmousedown: ((MouseEvent) -> dynamic)?\n      get() =
definedExternally\n      set(value) = definedExternally\n      var onmouseenter: ((MouseEvent) -> dynamic)?\n
get() = definedExternally\n      set(value) = definedExternally\n      var onmouseleave: ((MouseEvent) ->
dynamic)?\n      get() = definedExternally\n      set(value) = definedExternally\n      var onmousemove:
((MouseEvent) -> dynamic)?\n      get() = definedExternally\n      set(value) = definedExternally\n      var
onmouseout: ((MouseEvent) -> dynamic)?\n      get() = definedExternally\n      set(value) = definedExternally\n
var onmouseover: ((MouseEvent) -> dynamic)?\n      get() = definedExternally\n      set(value) =
definedExternally\n      var onmouseup: ((MouseEvent) -> dynamic)?\n      get() = definedExternally\n
set(value) = definedExternally\n      var onwheel: ((WheelEvent) -> dynamic)?\n      get() = definedExternally\n
set(value) = definedExternally\n      var onpause: ((Event) -> dynamic)?\n      get() = definedExternally\n
set(value) = definedExternally\n      var onplay: ((Event) -> dynamic)?\n      get() = definedExternally\n
set(value) = definedExternally\n      var onplaying: ((Event) -> dynamic)?\n      get() = definedExternally\n
set(value) = definedExternally\n      var onprogress: ((ProgressEvent) -> dynamic)?\n      get() = definedExternally\n
set(value) = definedExternally\n      var onratechange: ((Event) -> dynamic)?\n      get() = definedExternally\n
set(value) = definedExternally\n      var onreset: ((Event) -> dynamic)?\n      get() = definedExternally\n
set(value) = definedExternally\n      var onresize: ((Event) -> dynamic)?\n      get() = definedExternally\n

```

```

set(value) = definedExternally\n    var onscroll: ((Event) -> dynamic)?\n    get() = definedExternally\n
set(value) = definedExternally\n    var onseeked: ((Event) -> dynamic)?\n    get() = definedExternally\n
set(value) = definedExternally\n    var onseeking: ((Event) -> dynamic)?\n    get() = definedExternally\n
set(value) = definedExternally\n    var onselect: ((Event) -> dynamic)?\n    get() = definedExternally\n
set(value) = definedExternally\n    var onshow: ((Event) -> dynamic)?\n    get() = definedExternally\n
set(value) = definedExternally\n    var onstalled: ((Event) -> dynamic)?\n    get() = definedExternally\n
set(value) = definedExternally\n    var onsubmit: ((Event) -> dynamic)?\n    get() = definedExternally\n
set(value) = definedExternally\n    var onsuspend: ((Event) -> dynamic)?\n    get() = definedExternally\n
set(value) = definedExternally\n    var ontimeupdate: ((Event) -> dynamic)?\n    get() = definedExternally\n
set(value) = definedExternally\n    var ontoggle: ((Event) -> dynamic)?\n    get() = definedExternally\n
set(value) = definedExternally\n    var onvolumechange: ((Event) -> dynamic)?\n    get() = definedExternally\n
set(value) = definedExternally\n    var onwaiting: ((Event) -> dynamic)?\n    get() = definedExternally\n
set(value) = definedExternally\n    var ongotpointercapture: ((PointerEvent) -> dynamic)?\n    get() =
definedExternally\n    set(value) = definedExternally\n    var onlostpointercapture: ((PointerEvent) -> dynamic)?\n
    get() = definedExternally\n    set(value) = definedExternally\n    var onpointerdown: ((PointerEvent) ->
dynamic)?\n    get() = definedExternally\n    set(value) = definedExternally\n    var onpointermove:
((PointerEvent) -> dynamic)?\n    get() = definedExternally\n    set(value) = definedExternally\n    var
onpointerup: ((PointerEvent) -> dynamic)?\n    get() = definedExternally\n    set(value) = definedExternally\n
var onpointercancel: ((PointerEvent) -> dynamic)?\n    get() = definedExternally\n    set(value) =
definedExternally\n    var onpointerover: ((PointerEvent) -> dynamic)?\n    get() = definedExternally\n
set(value) = definedExternally\n    var onpointerout: ((PointerEvent) -> dynamic)?\n    get() = definedExternally\n
    set(value) = definedExternally\n    var onpointerenter: ((PointerEvent) -> dynamic)?\n    get() =
definedExternally\n    set(value) = definedExternally\n    var onpointerleave: ((PointerEvent) -> dynamic)?\n
get() = definedExternally\n    set(value) = definedExternally\n}\n\n**\n * Exposes the JavaScript
[WindowEventHandlers](https://developer.mozilla.org/en/docs/Web/API/WindowEventHandlers) to Kotlin\n
*\n\npublic external interface WindowEventHandlers {\n    var onafterprint: ((Event) -> dynamic)?\n    get() =
definedExternally\n    set(value) = definedExternally\n    var onbeforeprint: ((Event) -> dynamic)?\n    get() =
definedExternally\n    set(value) = definedExternally\n    var onbeforeunload: ((BeforeUnloadEvent) ->
String)?\n    get() = definedExternally\n    set(value) = definedExternally\n    var onhashchange:
((HashChangeEvent) -> dynamic)?\n    get() = definedExternally\n    set(value) = definedExternally\n    var
onlanguagechange: ((Event) -> dynamic)?\n    get() = definedExternally\n    set(value) = definedExternally\n
var onmessage: ((MessageEvent) -> dynamic)?\n    get() = definedExternally\n    set(value) =
definedExternally\n    var onoffline: ((Event) -> dynamic)?\n    get() = definedExternally\n    set(value) =
definedExternally\n    var ononline: ((Event) -> dynamic)?\n    get() = definedExternally\n    set(value) =
definedExternally\n    var onpagehide: ((PageTransitionEvent) -> dynamic)?\n    get() = definedExternally\n
set(value) = definedExternally\n    var onpageshow: ((PageTransitionEvent) -> dynamic)?\n    get() =
definedExternally\n    set(value) = definedExternally\n    var onpopstate: ((PopStateEvent) -> dynamic)?\n
get() = definedExternally\n    set(value) = definedExternally\n    var onrejectionhandled: ((Event) -> dynamic)?\n
    get() = definedExternally\n    set(value) = definedExternally\n    var onstorage: ((StorageEvent) -> dynamic)?\n
    get() = definedExternally\n    set(value) = definedExternally\n    var onunhandledrejection:
((PromiseRejectionEvent) -> dynamic)?\n    get() = definedExternally\n    set(value) = definedExternally\n
var onunload: ((Event) -> dynamic)?\n    get() = definedExternally\n    set(value) =
definedExternally\n}\n\npublic external interface DocumentAndElementEventHandlers {\n    var oncopy:
((ClipboardEvent) -> dynamic)?\n    get() = definedExternally\n    set(value) = definedExternally\n    var oncut:
((ClipboardEvent) -> dynamic)?\n    get() = definedExternally\n    set(value) = definedExternally\n    var
onpaste: ((ClipboardEvent) -> dynamic)?\n    get() = definedExternally\n    set(value) =
definedExternally\n}\n\n**\n * Exposes the JavaScript
[WindowOrWorkerGlobalScope](https://developer.mozilla.org/en/docs/Web/API/WindowOrWorkerGlobalScope)

```

to Kotlin\n */\npublic external interface WindowOrWorkerGlobalScope {\n val origin: String\n val caches: CacheStorage\n fun btoa(data: String): String\n fun atob(data: String): String\n fun setTimeout(handler: dynamic, timeout: Int = definedExternally, vararg arguments: Any?): Int\n fun clearTimeout(handle: Int = definedExternally)\n fun setInterval(handler: dynamic, timeout: Int = definedExternally, vararg arguments: Any?): Int\n fun clearInterval(handle: Int = definedExternally)\n fun createImageBitmap(image: ImageBitmapSource, options: ImageBitmapOptions = definedExternally): Promise<ImageBitmap>\n fun createImageBitmap(image: ImageBitmapSource, sx: Int, sy: Int, sw: Int, sh: Int, options: ImageBitmapOptions = definedExternally): Promise<ImageBitmap>\n fun fetch(input: dynamic, init: RequestInit = definedExternally): Promise<Response>\n}\n\n**\n * Exposes the JavaScript

[Navigator](https://developer.mozilla.org/en/docs/Web/API/Navigator) to Kotlin\n */\npublic external abstract class Navigator : NavigatorID, NavigatorLanguage, NavigatorOnLine, NavigatorContentUtils, NavigatorCookies, NavigatorPlugins, NavigatorConcurrentHardware {\n open val clipboard: Clipboard\n open val mediaDevices: MediaDevices\n open val maxTouchPoints: Int\n open val serviceWorker: ServiceWorkerContainer\n fun requestMediaKeySystemAccess(keySystem: String, supportedConfigurations: Array<MediaKeySystemConfiguration>): Promise<MediaKeySystemAccess>\n fun getUserMedia(constraints: MediaStreamConstraints, successCallback: (MediaStream) -> Unit, errorCallback: (dynamic) -> Unit)\n fun vibrate(pattern: dynamic): Boolean\n}\n\n**\n * Exposes the JavaScript

[NavigatorID](https://developer.mozilla.org/en/docs/Web/API/NavigatorID) to Kotlin\n */\npublic external interface NavigatorID {\n val appCodeName: String\n val appName: String\n val appVersion: String\n val platform: String\n val product: String\n val productSub: String\n val userAgent: String\n val vendor: String\n val vendorSub: String\n val oscpu: String\n fun taintEnabled(): Boolean\n}\n\n**\n * Exposes the JavaScript

[NavigatorLanguage](https://developer.mozilla.org/en/docs/Web/API/NavigatorLanguage) to Kotlin\n */\npublic external interface NavigatorLanguage {\n val language: String\n val languages: Array<out String>\n}\n\npublic external interface NavigatorContentUtils {\n fun registerProtocolHandler(scheme: String, url: String, title: String)\n fun registerContentHandler(mimeType: String, url: String, title: String)\n fun isProtocolHandlerRegistered(scheme: String, url: String): String\n fun isContentHandlerRegistered(mimeType: String, url: String): String\n fun unregisterProtocolHandler(scheme: String, url: String)\n fun unregisterContentHandler(mimeType: String, url: String)\n}\n\npublic external interface NavigatorCookies {\n val cookieEnabled: Boolean\n}\n\n**\n * Exposes the JavaScript

[NavigatorPlugins](https://developer.mozilla.org/en/docs/Web/API/NavigatorPlugins) to Kotlin\n */\npublic external interface NavigatorPlugins {\n val plugins: PluginArray\n val mimeTypes: MimeTypeArray\n fun javaEnabled(): Boolean\n}\n\n**\n * Exposes the JavaScript

[PluginArray](https://developer.mozilla.org/en/docs/Web/API/PluginArray) to Kotlin\n */\npublic external abstract class PluginArray : ItemArrayLike<Plugin> {\n fun refresh(reload: Boolean = definedExternally)\n override fun item(index: Int): Plugin?\n fun namedItem(name: String): Plugin?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun PluginArray.get(index: Int): Plugin? = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun PluginArray.get(name: String): Plugin? = asDynamic()[name]\n\n**\n * Exposes the JavaScript

[MimeTypeArray](https://developer.mozilla.org/en/docs/Web/API/MimeTypeArray) to Kotlin\n */\npublic external abstract class MimeTypeArray : ItemArrayLike<MimeType> {\n override fun item(index: Int): MimeType?\n fun namedItem(name: String): MimeType?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun MimeTypeArray.get(index: Int): MimeType? = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun MimeTypeArray.get(name: String): MimeType? = asDynamic()[name]\n\n**\n * Exposes the JavaScript

[Plugin](https://developer.mozilla.org/en/docs/Web/API/Plugin) to Kotlin\n */\npublic external abstract class Plugin

```

: ItemArrayLike<MimeType> {
    open val name: String
    open val description: String
    open val filename: String
    override fun item(index: Int): MimeType?
    fun namedItem(name: String): MimeType?
}
@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline operator fun Plugin.get(index: Int): MimeType? = asDynamic()[index]
@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline operator fun Plugin.get(name: String): MimeType? = asDynamic()[name]
/** Exposes the JavaScript [MimeType](https://developer.mozilla.org/en/docs/Web/API/MimeType) to Kotlin
 */
public external abstract class MimeType {
    open val type: String
    open val description: String
    open val suffixes: String
    open val enabledPlugin: Plugin
}
/** Exposes the JavaScript [ImageBitmap](https://developer.mozilla.org/en/docs/Web/API/ImageBitmap) to Kotlin
 */
public external abstract class ImageBitmap : CanvasImageSource, TexImageSource {
    open val width: Int
    open val height: Int
    fun close()
}
public external interface ImageBitmapOptions {
    var imageOrientation: ImageOrientation? /* = ImageOrientation.NONE */
    get() = definedExternally
    set(value) = definedExternally
    var premultiplyAlpha: PremultiplyAlpha? /* = PremultiplyAlpha.DEFAULT */
    get() = definedExternally
    set(value) = definedExternally
    var colorSpaceConversion: ColorSpaceConversion? /* = ColorSpaceConversion.DEFAULT */
    get() = definedExternally
    set(value) = definedExternally
    var resizeMode: Int?
    get() = definedExternally
    set(value) = definedExternally
    var resizeMode: Int?
    get() = definedExternally
    set(value) = definedExternally
    var resizeMode: Int?
    get() = definedExternally
    set(value) = definedExternally
    var resizeMode: Int?
    get() = definedExternally
    set(value) = definedExternally
}
@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline fun ImageBitmapOptions(imageOrientation: ImageOrientation? = ImageOrientation.NONE, premultiplyAlpha: PremultiplyAlpha? = PremultiplyAlpha.DEFAULT, colorSpaceConversion: ColorSpaceConversion? = ColorSpaceConversion.DEFAULT, resizeMode: Int? = undefined, resizeMode: Int? = undefined, resizeMode: Int? = undefined, resizeMode: Int? = undefined): ImageBitmapOptions {
    val o = js("{}")
    o["imageOrientation"] = imageOrientation
    o["premultiplyAlpha"] = premultiplyAlpha
    o["colorSpaceConversion"] = colorSpaceConversion
    o["resizeWidth"] = resizeMode
    o["resizeHeight"] = resizeMode
    o["resizeQuality"] = resizeMode
    return o
}
/** Exposes the JavaScript [MessageEvent](https://developer.mozilla.org/en/docs/Web/API/MessageEvent) to Kotlin
 */
public external open class MessageEvent(type: String, eventInitDict: MessageEventInit = definedExternally) : Event {
    open val data: Any?
    open val origin: String
    open val lastEventId: String
    open val source: UnionMessagePortOrWindowProxy?
    open val ports: Array<out MessagePort>
    fun initMessageEvent(type: String, bubbles: Boolean, cancelable: Boolean, data: Any?, origin: String, lastEventId: String, source: UnionMessagePortOrWindowProxy?, ports: Array<MessagePort>)
    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}
public external interface MessageEventInit : EventInit {
    var data: Any? /* = null */
    get() = definedExternally
    set(value) = definedExternally
    var origin: String? /* = "" */
    get() = definedExternally
    set(value) = definedExternally
    var lastEventId: String? /* = "" */
    get() = definedExternally
    set(value) = definedExternally
    var source: UnionMessagePortOrWindowProxy? /* = null */
    get() = definedExternally
    set(value) = definedExternally
    var ports: Array<MessagePort>? /* = arrayOf() */
    get() = definedExternally
    set(value) = definedExternally
}
@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline fun MessageEventInit(data: Any? = null, origin: String? = "", lastEventId: String? = "", source: UnionMessagePortOrWindowProxy? = null, ports: Array<MessagePort>? = arrayOf(), bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): MessageEventInit {
    val o = js("{}")
    o["data"] = data
    o["origin"] = origin
    o["lastEventId"] = lastEventId
    o["source"] = source
    o["ports"] = ports
    o["bubbles"] = bubbles
}

```

```

o["cancelable"] = cancelable\n  o["composed"] = composed\n  return o\n}\n\n/**\n * Exposes the JavaScript
[EventSource](https://developer.mozilla.org/en/docs/Web/API/EventSource) to Kotlin\n *\npublic external open
class EventSource(url: String, eventSourceInitDict: EventSourceInit = definedExternally) : EventTarget {\n  open
val url: String\n  open val withCredentials: Boolean\n  open val readyState: Short\n  var onopen: ((Event) ->
dynamic)?\n  var onmessage: ((MessageEvent) -> dynamic)?\n  var onerror: ((Event) -> dynamic)?\n  fun
close()\n\n  companion object {\n    val CONNECTING: Short\n    val OPEN: Short\n    val CLOSED:
Short\n  }\n}\n\npublic external interface EventSourceInit {\n  var withCredentials: Boolean? /* = false *\n  get() = definedExternally\n  set(value) = definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun EventSourceInit(withCredentials:
Boolean? = false): EventSourceInit {\n  val o = js("{}")\n  o["withCredentials"] = withCredentials\n  return
o\n}\n\n/**\n * Exposes the JavaScript [WebSocket](https://developer.mozilla.org/en/docs/Web/API/WebSocket) to
Kotlin\n *\npublic external open class WebSocket(url: String, protocols: dynamic = definedExternally) :
EventTarget {\n  open val url: String\n  open val readyState: Short\n  open val bufferedAmount: Number\n  var
onopen: ((Event) -> dynamic)?\n  var onerror: ((Event) -> dynamic)?\n  var onclose: ((Event) -> dynamic)?\n
open val extensions: String\n  open val protocol: String\n  var onmessage: ((MessageEvent) -> dynamic)?\n  var
binaryType: BinaryType\n  fun close(code: Short = definedExternally, reason: String = definedExternally)\n  fun
send(data: String)\n  fun send(data: Blob)\n  fun send(data: ArrayBuffer)\n  fun send(data:
ArrayBufferView)\n\n  companion object {\n    val CONNECTING: Short\n    val OPEN: Short\n    val
CLOSING: Short\n    val CLOSED: Short\n  }\n}\n\n/**\n * Exposes the JavaScript
[CloseEvent](https://developer.mozilla.org/en/docs/Web/API/CloseEvent) to Kotlin\n *\npublic external open class
CloseEvent(type: String, eventInitDict: CloseEventInit = definedExternally) : Event {\n  open val wasClean:
Boolean\n  open val code: Short\n  open val reason: String\n\n  companion object {\n    val NONE: Short\n    val
CAPTURING_PHASE: Short\n    val AT_TARGET: Short\n    val BUBBLING_PHASE: Short\n  }\n}\n\npublic external interface CloseEventInit : EventInit {\n  var wasClean: Boolean? /* = false *\n  get() =
definedExternally\n  set(value) = definedExternally\n  var code: Short? /* = 0 *\n  get() =
definedExternally\n  set(value) = definedExternally\n  var reason: String? /* = "" *\n  get() =
definedExternally\n  set(value) = definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun CloseEventInit(wasClean: Boolean? =
false, code: Short? = 0, reason: String? = "", bubbles: Boolean? = false, cancelable: Boolean? = false, composed:
Boolean? = false): CloseEventInit {\n  val o = js("{}")\n  o["wasClean"] = wasClean\n  o["code"] = code\n  o["reason"] = reason\n  o["bubbles"] = bubbles\n  o["cancelable"] = cancelable\n  o["composed"] =
composed\n  return o\n}\n\n/**\n * Exposes the JavaScript
[MessageChannel](https://developer.mozilla.org/en/docs/Web/API/MessageChannel) to Kotlin\n *\npublic external
open class MessageChannel {\n  open val port1: MessagePort\n  open val port2: MessagePort\n}\n\n/**\n *
Exposes the JavaScript [MessagePort](https://developer.mozilla.org/en/docs/Web/API/MessagePort) to Kotlin\n
*\npublic external abstract class MessagePort : EventTarget, UnionMessagePortOrWindowProxy,
UnionMessagePortOrServiceWorker, UnionClientOrMessagePortOrServiceWorker {\n  open var onmessage:
((MessageEvent) -> dynamic)?\n  fun postMessage(message: Any?, transfer: Array<dynamic> =
definedExternally)\n  fun start()\n  fun close()\n}\n\n/**\n * Exposes the JavaScript
[BroadcastChannel](https://developer.mozilla.org/en/docs/Web/API/BroadcastChannel) to Kotlin\n *\npublic
external open class BroadcastChannel(name: String) : EventTarget {\n  open val name: String\n  var onmessage:
((MessageEvent) -> dynamic)?\n  fun postMessage(message: Any?)\n  fun close()\n}\n\n/**\n * Exposes the
JavaScript [WorkerGlobalScope](https://developer.mozilla.org/en/docs/Web/API/WorkerGlobalScope) to Kotlin\n
*\npublic external abstract class WorkerGlobalScope : EventTarget, WindowOrWorkerGlobalScope,
GlobalPerformance {\n  open val self: WorkerGlobalScope\n  open val location: WorkerLocation\n  open val
navigator: WorkerNavigator\n  open var onerror: ((dynamic, String, Int, Int, Any?) -> dynamic)?\n  open var
onlanguagechange: ((Event) -> dynamic)?\n  open var onoffline: ((Event) -> dynamic)?\n  open var ononline:
((Event) -> dynamic)?\n  open var onrejectionhandled: ((Event) -> dynamic)?\n  open var onunhandledrejection:

```

```

(PromiseRejectionEvent) -> dynamic)?\n fun importScripts(vararg urls: String)\n}\n\n/**\n * Exposes the
JavaScript
[DedicatedWorkerGlobalScope](https://developer.mozilla.org/en/docs/Web/API/DedicatedWorkerGlobalScope) to
Kotlin\n *\npublic external abstract class DedicatedWorkerGlobalScope : WorkerGlobalScope {\n open var
onmessage: ((MessageEvent) -> dynamic)?\n fun postMessage(message: Any?, transfer: Array<dynamic> =
definedExternally)\n fun close()\n}\n\n/**\n * Exposes the JavaScript
[SharedWorkerGlobalScope](https://developer.mozilla.org/en/docs/Web/API/SharedWorkerGlobalScope) to
Kotlin\n *\npublic external abstract class SharedWorkerGlobalScope : WorkerGlobalScope {\n open val name:
String\n open val applicationCache: ApplicationCache\n open var onconnect: ((Event) -> dynamic)?\n fun
close()\n}\n\n/**\n * Exposes the JavaScript
[AbstractWorker](https://developer.mozilla.org/en/docs/Web/API/AbstractWorker) to Kotlin\n *\npublic external
interface AbstractWorker {\n var onerror: ((Event) -> dynamic)?\n get() = definedExternally\n set(value)
= definedExternally\n}\n\n/**\n * Exposes the JavaScript
[Worker](https://developer.mozilla.org/en/docs/Web/API/Worker) to Kotlin\n *\npublic external open class
Worker(scriptURL: String, options: WorkerOptions = definedExternally) : EventTarget, AbstractWorker {\n var
onmessage: ((MessageEvent) -> dynamic)?\n override var onerror: ((Event) -> dynamic)?\n fun terminate()\n
fun postMessage(message: Any?, transfer: Array<dynamic> = definedExternally)\n}\n\npublic external interface
WorkerOptions {\n var type: WorkerType? /* = WorkerType.CLASSIC */\n get() = definedExternally\n
set(value) = definedExternally\n var credentials: RequestCredentials? /* = RequestCredentials.OMIT */\n
get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun WorkerOptions(type: WorkerType? =
WorkerType.CLASSIC, credentials: RequestCredentials? = RequestCredentials.OMIT): WorkerOptions {\n val o
= js("{}")\n o["type"] = type\n o["credentials"] = credentials\n return o\n}\n\n/**\n * Exposes the
JavaScript [SharedWorker](https://developer.mozilla.org/en/docs/Web/API/SharedWorker) to Kotlin\n *\npublic
external open class SharedWorker(scriptURL: String, name: String = definedExternally, options: WorkerOptions =
definedExternally) : EventTarget, AbstractWorker {\n open val port: MessagePort\n override var onerror:
((Event) -> dynamic)?\n}\n\n/**\n * Exposes the JavaScript
[NavigatorConcurrentHardware](https://developer.mozilla.org/en/docs/Web/API/NavigatorConcurrentHardware) to
Kotlin\n *\npublic external interface NavigatorConcurrentHardware {\n val hardwareConcurrency:
Number\n}\n\n/**\n * Exposes the JavaScript
[WorkerNavigator](https://developer.mozilla.org/en/docs/Web/API/WorkerNavigator) to Kotlin\n *\npublic
external abstract class WorkerNavigator : NavigatorID, NavigatorLanguage, NavigatorOnLine,
NavigatorConcurrentHardware {\n open val serviceWorker: ServiceWorkerContainer\n}\n\n/**\n * Exposes the
JavaScript [WorkerLocation](https://developer.mozilla.org/en/docs/Web/API/WorkerLocation) to Kotlin\n
*\npublic external abstract class WorkerLocation {\n open val href: String\n open val origin: String\n open val
protocol: String\n open val host: String\n open val hostname: String\n open val port: String\n open val
pathname: String\n open val search: String\n open val hash: String\n}\n\n/**\n * Exposes the JavaScript
[Storage](https://developer.mozilla.org/en/docs/Web/API/Storage) to Kotlin\n *\npublic external abstract class
Storage {\n open val length: Int\n fun key(index: Int): String?\n fun removeItem(key: String)\n fun clear()\n
fun getItem(key: String): String?\n fun setItem(key: String, value:
String)\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun Storage.get(key: String):
String? = asDynamic()[key]\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun Storage.set(key: String, value:
String) { asDynamic()[key] = value }\n\n/**\n * Exposes the JavaScript
[WindowSessionStorage](https://developer.mozilla.org/en/docs/Web/API/WindowSessionStorage) to Kotlin\n
*\npublic external interface WindowSessionStorage {\n val sessionStorage: Storage\n}\n\n/**\n * Exposes the
JavaScript [WindowLocalStorage](https://developer.mozilla.org/en/docs/Web/API/WindowLocalStorage) to

```

```

Kotlin\n *\/\npublic external interface WindowLocalStorage {\n    val localStorage: Storage\n}\n\n/**\n * Exposes the JavaScript [StorageEvent](https://developer.mozilla.org/en/docs/Web/API/StorageEvent) to Kotlin\n *\/\npublic external open class StorageEvent(type: String, eventInitDict: StorageEventInit = definedExternally) : Event {\n    open val key: String?\n    open val oldValue: String?\n    open val newValue: String?\n    open val url: String\n    open val storageArea: Storage?\n\n    companion object {\n        val NONE: Short\n        val CAPTURING_PHASE: Short\n        val AT_TARGET: Short\n        val BUBBLING_PHASE: Short\n    }\n}\n\npublic external interface StorageEventInit : EventInit {\n    var key: String? /* = null */\n    get() = definedExternally\n    set(value) = definedExternally\n    var oldValue: String? /* = null */\n    get() = definedExternally\n    set(value) = definedExternally\n    var newValue: String? /* = null */\n    get() = definedExternally\n    set(value) = definedExternally\n    var url: String? /* = \"\" */\n    get() = definedExternally\n    set(value) = definedExternally\n    var storageArea: Storage? /* = null */\n    get() = definedExternally\n    set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun StorageEventInit(key: String? = null, oldValue: String? = null, newValue: String? = null, url: String? = \"\", storageArea: Storage? = null, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): StorageEventInit {\n    val o = js(\"({})\")\n    o[\"key\"] = key\n    o[\"oldValue\"] = oldValue\n    o[\"newValue\"] = newValue\n    o[\"url\"] = url\n    o[\"storageArea\"] = storageArea\n    o[\"bubbles\"] = bubbles\n    o[\"cancelable\"] = cancelable\n    o[\"composed\"] = composed\n    return o\n}\n\npublic external abstract class HTMLAppletElement : HTMLElement {\n    open var align: String\n    open var alt: String\n    open var archive: String\n    open var code: String\n    open var codeBase: String\n    open var height: String\n    open var hspace: Int\n    open var name: String\n    open var _object: String\n    open var vspace: Int\n    open var width: String\n\n    companion object {\n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript [HTMLMarqueeElement](https://developer.mozilla.org/en/docs/Web/API/HTMLMarqueeElement) to Kotlin\n *\/\npublic external abstract class HTMLMarqueeElement : HTMLElement {\n    open var behavior: String\n    open var bgColor: String\n    open var direction: String\n    open var height: String\n    open var hspace: Int\n    open var loop: Int\n    open var scrollAmount: Int\n    open var scrollDelay: Int\n    open var trueSpeed: Boolean\n    open var vspace: Int\n    open var width: String\n    open var onbounce: ((Event) -> dynamic)?\n    open var onfinish: ((Event) -> dynamic)?\n    open var onstart: ((Event) -> dynamic)?\n    fun start()\n    fun stop()\n\n    companion object {\n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript [HTMLFrameSetElement](https://developer.mozilla.org/en/docs/Web/API/HTMLFrameSetElement) to Kotlin\n *\/\npublic external abstract class HTMLFrameSetElement : HTMLElement, WindowEventHandlers {\n    open var cols: String\n    open var rows: String\n\n    companion object {\n        val ELEMENT_NODE: Short\n        val

```

```

ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n    val CDATA_SECTION_NODE: Short\n    val
ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE: Short\n    val
PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val
DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\npublic external abstract class
HTMLFrameElement : HTMLElement {\n    open var name: String\n    open var scrolling: String\n    open var src:
String\n    open var frameBorder: String\n    open var longDesc: String\n    open var noResize: Boolean\n    open val
contentDocument: Document?\n    open val contentWindow: Window?\n    open var marginHeight: String\n    open
var marginWidth: String\n\n    companion object {\n        val ELEMENT_NODE: Short\n        val
ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val
ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val
PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\npublic external abstract class
HTMLDirectoryElement : HTMLElement {\n    open var compact: Boolean\n\n    companion object {\n        val
ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val
CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE:
Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[HTMLFontElement](https://developer.mozilla.org/en/docs/Web/API/HTMLFontElement) to Kotlin\n *\npublic
external abstract class HTMLFontElement : HTMLElement {\n    open var color: String\n    open var face: String\n
open var size: String\n\n    companion object {\n        val ELEMENT_NODE: Short\n        val
ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val
ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val
PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\npublic external interface External
{\n    fun AddSearchProvider()\n    fun IsSearchProviderInstalled()\n}\n\npublic external interface EventInit {\n
var bubbles: Boolean? /* = false */\n    get() = definedExternally\n    set(value) = definedExternally\n
var cancelable: Boolean? /* = false */\n    get() = definedExternally\n    set(value) = definedExternally\n
var composed: Boolean? /* = false */\n    get() = definedExternally\n    set(value) =

```


[HTMLCollection](https://developer.mozilla.org/en/docs/Web/API/HTMLCollection) to Kotlin\n *\npublic external abstract class HTMLCollection : ItemArrayLike<Element>, UnionElementOrHTMLCollection {\n override fun item(index: Int): Element?\n fun namedItem(name: String): Element?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun HTMLCollection.get(index: Int): Element? = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun HTMLCollection.get(name: String): Element? = asDynamic()[name]\n\n/**\n * Exposes the JavaScript

[MutationObserver](https://developer.mozilla.org/en/docs/Web/API/MutationObserver) to Kotlin\n *\npublic external open class MutationObserver(callback: (Array<MutationRecord>, MutationObserver) -> Unit) {\n fun observe(target: Node, options: MutationObserverInit = definedExternally)\n fun disconnect()\n fun takeRecords(): Array<MutationRecord>\n}\n\n/**\n * Exposes the JavaScript

[MutationObserverInit](https://developer.mozilla.org/en/docs/Web/API/MutationObserverInit) to Kotlin\n *\npublic external interface MutationObserverInit {\n var childList: Boolean? /* = false */\n get() = definedExternally\n set(value) = definedExternally\n var attributes: Boolean?\n get() = definedExternally\n set(value) = definedExternally\n var characterData: Boolean?\n get() = definedExternally\n set(value) = definedExternally\n var subtree: Boolean? /* = false */\n get() = definedExternally\n set(value) = definedExternally\n var attributeOldValue: Boolean?\n get() = definedExternally\n set(value) = definedExternally\n var characterDataOldValue: Boolean?\n get() = definedExternally\n set(value) = definedExternally\n var attributeFilter: Array<String>?\n get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun MutationObserverInit(childList: Boolean? = false, attributes: Boolean? = undefined, characterData: Boolean? = undefined, subtree: Boolean? = false, attributeOldValue: Boolean? = undefined, characterDataOldValue: Boolean? = undefined, attributeFilter: Array<String>? = undefined): MutationObserverInit {\n val o = js(\"({})\")\n o[\"childList\"] = childList\n o[\"attributes\"] = attributes\n o[\"characterData\"] = characterData\n o[\"subtree\"] = subtree\n o[\"attributeOldValue\"] = attributeOldValue\n o[\"characterDataOldValue\"] = characterDataOldValue\n o[\"attributeFilter\"] = attributeFilter\n return o\n}\n\n/**\n * Exposes the JavaScript

[MutationRecord](https://developer.mozilla.org/en/docs/Web/API/MutationRecord) to Kotlin\n *\npublic external abstract class MutationRecord {\n open val type: String\n open val target: Node\n open val addedNodes: NodeList\n open val removedNodes: NodeList\n open val previousSibling: Node?\n open val nextSibling: Node?\n open val attributeName: String?\n open val attributeNamespace: String?\n open val oldValue: String?\n}\n\n/**\n * Exposes the JavaScript

[Node](https://developer.mozilla.org/en/docs/Web/API/Node) to Kotlin\n *\npublic external abstract class Node : EventTarget {\n open val nodeType: Short\n open val nodeName: String\n open val baseURI: String\n open val isConnected: Boolean\n open val ownerDocument: Document?\n open val parentNode: Node?\n open val parentElement: Element?\n open val childNodes: NodeList\n open val firstChild: Node?\n open val lastChild: Node?\n open val previousSibling: Node?\n open val nextSibling: Node?\n open var nodeValue: String?\n open var textContent: String?\n fun getRootNode(options: GetRootNodeOptions = definedExternally): Node\n fun hasChildNodes(): Boolean\n fun normalize()\n fun cloneNode(deep: Boolean = definedExternally): Node\n fun isEqualNode(otherNode: Node?): Boolean\n fun isSameNode(otherNode: Node?): Boolean\n fun compareDocumentPosition(other: Node): Short\n fun contains(other: Node?): Boolean\n fun lookupPrefix(namespace: String?): String?\n fun lookupNamespaceURI(prefix: String?): String?\n fun isDefaultNamespace(namespace: String?): Boolean\n fun insertBefore(node: Node, child: Node?): Node\n fun appendChild(node: Node): Node\n fun replaceChild(node: Node, child: Node): Node\n fun removeChild(child: Node): Node\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val

```

DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\npublic external interface
GetRootNodeOptions {\n    var composed: Boolean? /* = false */\n    get() = definedExternally\n    set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n\n@kotlin.internal.InlineOnly\n\npublic inline fun GetRootNodeOptions(composed:
Boolean? = false): GetRootNodeOptions {\n    val o = js(\"({})\")\n    o[\"composed\"] = composed\n    return
o\n}\n\n/**\n * Exposes the JavaScript [Document](https://developer.mozilla.org/en/docs/Web/API/Document) to
Kotlin\n */\n\npublic external open class Document : Node, GlobalEventHandlers,
DocumentAndElementEventHandlers, NonElementParentNode, DocumentOrShadowRoot, ParentNode,
GeometryUtils {\n    open val implementation: DOMImplementation\n    open val URL: String\n    open val
documentURI: String\n    open val origin: String\n    open val compatMode: String\n    open val characterSet:
String\n    open val charset: String\n    open val inputEncoding: String\n    open val contentType: String\n    open val
doctype: DocumentType?\n    open val documentElement: Element?\n    open val location: Location?\n    var
domain: String\n    open val referrer: String\n    var cookie: String\n    open val lastModified: String\n    open val
readyState: DocumentReadyState\n    var title: String\n    var dir: String\n    var body: HTMLElement?\n    open val
head: HTMLHeadElement?\n    open val images: HTMLCollection\n    open val embeds: HTMLCollection\n    open
val plugins: HTMLCollection\n    open val links: HTMLCollection\n    open val forms: HTMLCollection\n    open
val scripts: HTMLCollection\n    open val currentScript: HTMLScriptElement?\n    open val defaultView:
Window?\n    open val activeElement: Element?\n    var designMode: String\n    var onreadystatechange: ((Event) ->
dynamic)?\n    var fgColor: String\n    var linkColor: String\n    var vlinkColor: String\n    var alinkColor: String\n
var bgColor: String\n    open val anchors: HTMLCollection\n    open val applets: HTMLCollection\n    open val all:
HTMLAllCollection\n    open val scrollingElement: Element?\n    open val styleSheets: StyleSheetList\n    open val
rootElement: SVGSVGElement?\n    open val fullscreenEnabled: Boolean\n    open val fullscreen: Boolean\n    var
onfullscreenchange: ((Event) -> dynamic)?\n    var onfullscreenerror: ((Event) -> dynamic)?\n    override var
onabort: ((Event) -> dynamic)?\n    override var onblur: ((FocusEvent) -> dynamic)?\n    override var oncancel:
((Event) -> dynamic)?\n    override var oncanplay: ((Event) -> dynamic)?\n    override var oncanplaythrough:
((Event) -> dynamic)?\n    override var onchange: ((Event) -> dynamic)?\n    override var onclick: ((MouseEvent) ->
dynamic)?\n    override var onclose: ((Event) -> dynamic)?\n    override var oncontextmenu: ((MouseEvent) ->
dynamic)?\n    override var oncuechange: ((Event) -> dynamic)?\n    override var ondblclick: ((MouseEvent) ->
dynamic)?\n    override var ondrag: ((DragEvent) -> dynamic)?\n    override var ondragend: ((DragEvent) ->
dynamic)?\n    override var ondragenter: ((DragEvent) -> dynamic)?\n    override var ondragexit: ((DragEvent) ->
dynamic)?\n    override var ondragleave: ((DragEvent) -> dynamic)?\n    override var ondragover: ((DragEvent) ->
dynamic)?\n    override var ondragstart: ((DragEvent) -> dynamic)?\n    override var ondrop: ((DragEvent) ->
dynamic)?\n    override var ondurationchange: ((Event) -> dynamic)?\n    override var onemptied: ((Event) ->
dynamic)?\n    override var onended: ((Event) -> dynamic)?\n    override var onerror: ((dynamic, String, Int, Int,
Any?) -> dynamic)?\n    override var onfocus: ((FocusEvent) -> dynamic)?\n    override var oninput: ((InputEvent) ->
dynamic)?\n    override var oninvalid: ((Event) -> dynamic)?\n    override var onkeydown: ((KeyboardEvent) ->
dynamic)?\n    override var onkeypress: ((KeyboardEvent) -> dynamic)?\n    override var onkeyup:
((KeyboardEvent) -> dynamic)?\n    override var onload: ((Event) -> dynamic)?\n    override var onloadeddata:
((Event) -> dynamic)?\n    override var onloadedmetadata: ((Event) -> dynamic)?\n    override var onloadend:
((Event) -> dynamic)?\n    override var onloadstart: ((ProgressEvent) -> dynamic)?\n    override var onmousedown:
((MouseEvent) -> dynamic)?\n    override var onmouseenter: ((MouseEvent) -> dynamic)?\n    override var
onmouseleave: ((MouseEvent) -> dynamic)?\n    override var onmousemove: ((MouseEvent) -> dynamic)?\n    override
var onmouseout: ((MouseEvent) -> dynamic)?\n    override var onmouseover: ((MouseEvent) ->

```

```

dynamic)?\n  override var onmouseup: ((MouseEvent) -> dynamic)?\n  override var onwheel: ((WheelEvent) ->
dynamic)?\n  override var onpause: ((Event) -> dynamic)?\n  override var onplay: ((Event) -> dynamic)?\n
override var onplaying: ((Event) -> dynamic)?\n  override var onprogress: ((ProgressEvent) -> dynamic)?\n
override var onratechange: ((Event) -> dynamic)?\n  override var onreset: ((Event) -> dynamic)?\n  override var
onresize: ((Event) -> dynamic)?\n  override var onscroll: ((Event) -> dynamic)?\n  override var onseeked:
((Event) -> dynamic)?\n  override var onseeking: ((Event) -> dynamic)?\n  override var onselect: ((Event) ->
dynamic)?\n  override var onshow: ((Event) -> dynamic)?\n  override var onstalled: ((Event) -> dynamic)?\n
override var onsubmit: ((Event) -> dynamic)?\n  override var onsuspend: ((Event) -> dynamic)?\n  override var
ontimeupdate: ((Event) -> dynamic)?\n  override var ontoggle: ((Event) -> dynamic)?\n  override var
onvolumechange: ((Event) -> dynamic)?\n  override var onwaiting: ((Event) -> dynamic)?\n  override var
ongotpointercapture: ((PointerEvent) -> dynamic)?\n  override var onlostpointercapture: ((PointerEvent) ->
dynamic)?\n  override var onpointerdown: ((PointerEvent) -> dynamic)?\n  override var onpointermove:
((PointerEvent) -> dynamic)?\n  override var onpointerup: ((PointerEvent) -> dynamic)?\n  override var
onpointercancel: ((PointerEvent) -> dynamic)?\n  override var onpointerover: ((PointerEvent) -> dynamic)?\n
override var onpointerout: ((PointerEvent) -> dynamic)?\n  override var onpointerenter: ((PointerEvent) ->
dynamic)?\n  override var onpointerleave: ((PointerEvent) -> dynamic)?\n  override var oncopy:
((ClipboardEvent) -> dynamic)?\n  override var oncut: ((ClipboardEvent) -> dynamic)?\n  override var onpaste:
((ClipboardEvent) -> dynamic)?\n  override val fullscreenElement: Element?\n  override val children:
HTMLCollection\n  override val firstElementChild: Element?\n  override val lastElementChild: Element?\n
override val childElementCount: Int\n  fun getElementsByTagName(qualifiedName: String): HTMLCollection\n  fun
getElementsByTagNameNS(namespace: String?, localName: String): HTMLCollection\n  fun
getElementsByTagName(className: String): HTMLCollection\n  fun createElement(localName: String,
options: ElementCreationOptions = definedExternally): Element\n  fun createElementNS(namespace: String?,
qualifiedName: String, options: ElementCreationOptions = definedExternally): Element\n  fun
createDocumentFragment(): DocumentFragment\n  fun createTextNode(data: String): Text\n  fun
createCDATASection(data: String): CDATASection\n  fun createComment(data: String): Comment\n  fun
createProcessingInstruction(target: String, data: String): ProcessingInstruction\n  fun importNode(node: Node,
deep: Boolean = definedExternally): Node\n  fun adoptNode(node: Node): Node\n  fun
createAttribute(localName: String): Attr\n  fun createAttributeNS(namespace: String?, qualifiedName: String):
Attr\n  fun createEvent(`interface`: String): Event\n  fun createRange(): Range\n  fun createNodeIterator(root:
Node, whatToShow: Int = definedExternally, filter: NodeFilter? = definedExternally): NodeIterator\n  fun
createNodeIterator(root: Node, whatToShow: Int = definedExternally, filter: ((Node) -> Short)? =
definedExternally): NodeIterator\n  fun createTreeWalker(root: Node, whatToShow: Int = definedExternally, filter:
NodeFilter? = definedExternally): TreeWalker\n  fun createTreeWalker(root: Node, whatToShow: Int =
definedExternally, filter: ((Node) -> Short)? = definedExternally): TreeWalker\n  fun
getElementsByTagName(elementName: String): NodeList\n  fun open(type: String = definedExternally, replace:
String = definedExternally): Document\n  fun open(url: String, name: String, features: String): Window\n  fun
close()\n  fun write(vararg text: String)\n  fun writeln(vararg text: String)\n  fun hasFocus(): Boolean\n  fun
execCommand(commandId: String, showUI: Boolean = definedExternally, value: String = definedExternally):
Boolean\n  fun queryCommandEnabled(commandId: String): Boolean\n  fun
queryCommandIndeterm(commandId: String): Boolean\n  fun queryCommandState(commandId: String):
Boolean\n  fun queryCommandSupported(commandId: String): Boolean\n  fun
queryCommandValue(commandId: String): String\n  fun clear()\n  fun captureEvents()\n  fun releaseEvents()\n
fun elementFromPoint(x: Double, y: Double): Element?\n  fun elementsFromPoint(x: Double, y: Double):
Array<Element>\n  fun caretPositionFromPoint(x: Double, y: Double): CaretPosition?\n  fun createTouch(view:
Window, target: EventTarget, identifier: Int, pageX: Int, pageY: Int, screenX: Int, screenY: Int): Touch\n  fun
createTouchList(vararg touches: Touch): TouchList\n  fun exitFullscreen(): Promise<Unit>\n  override fun
getElementById(elementId: String): Element?\n  override fun prepend(vararg nodes: dynamic)\n  override fun

```

```

append(vararg nodes: dynamic)\n  override fun querySelector(selectors: String): Element?\n  override fun
querySelectorAll(selectors: String): NodeList\n  override fun getBoxQuads(options: BoxQuadOptions /* =
definedExternally */): Array<DOMQuad>\n  override fun convertQuadFromNode(quad: dynamic, from: dynamic,
options: ConvertCoordinateOptions /* = definedExternally */): DOMQuad\n  override fun
convertRectFromNode(rect: DOMRectReadOnly, from: dynamic, options: ConvertCoordinateOptions /* =
definedExternally */): DOMQuad\n  override fun convertPointFromNode(point: DOMPointInit, from: dynamic,
options: ConvertCoordinateOptions /* = definedExternally */): DOMPoint\n\n  companion object {\n    val
ELEMENT_NODE: Short\n    val ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n    val
CDATA_SECTION_NODE: Short\n    val ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE:
Short\n    val PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val
DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n
  }\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Document.get(name: String):
dynamic = asDynamic()[name]\n\n/**\n * Exposes the JavaScript
[XMLDocument](https://developer.mozilla.org/en/docs/Web/API/XMLDocument) to Kotlin\n */\npublic external
open class XMLDocument : Document {\n  companion object {\n    val ELEMENT_NODE: Short\n    val
ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n    val CDATA_SECTION_NODE: Short\n    val
ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE: Short\n    val
PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val
DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n  }\n}\n\npublic external interface
ElementCreationOptions {\n  var `is`: String?\n  get() = definedExternally\n  set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ElementCreationOptions(`is`: String?
= undefined): ElementCreationOptions {\n  val o = js(\"{\}\")\n  o[\"is\"] = `is`\n  return o\n}\n\n/**\n *
Exposes the JavaScript
[DOMImplementation](https://developer.mozilla.org/en/docs/Web/API/DOMImplementation) to Kotlin\n */\npublic
external abstract class DOMImplementation {\n  fun createDocumentType(qualifiedName: String,
publicId: String, systemId: String): DocumentType\n  fun createDocument(namespace: String?, qualifiedName:
String, doctype: DocumentType? = definedExternally): XMLDocument\n  fun createHTMLDocument(title: String
= definedExternally): Document\n  fun hasFeature(): Boolean\n}\n\n/**\n * Exposes the JavaScript
[DocumentType](https://developer.mozilla.org/en/docs/Web/API/DocumentType) to Kotlin\n */\npublic external
abstract class DocumentType : Node, ChildNode {\n  open val name: String\n  open val publicId: String\n  open
val systemId: String\n\n  companion object {\n    val ELEMENT_NODE: Short\n    val ATTRIBUTE_NODE:
Short\n    val TEXT_NODE: Short\n    val CDATA_SECTION_NODE: Short\n    val
ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE: Short\n    val
PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val
DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val

```

DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val

DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }n}\n\n/**\n * Exposes the JavaScript [DocumentFragment](https://developer.mozilla.org/en/docs/Web/API/DocumentFragment) to Kotlin\n *\npublic external open class DocumentFragment : Node, NonElementParentNode, ParentNode {\n override val children: HTMLCollection\n override val firstElementChild: Element?\n override val lastElementChild: Element?\n override val childElementCount: Int\n override fun getElementById(elementId: String): Element?\n override fun prepend(vararg nodes: dynamic)\n override fun append(vararg nodes: dynamic)\n override fun querySelector(selectors: String): Element?\n override fun querySelectorAll(selectors: String): NodeList\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val

DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val

DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }n}\n\n/**\n * Exposes the JavaScript [ShadowRoot](https://developer.mozilla.org/en/docs/Web/API/ShadowRoot) to Kotlin\n *\npublic external open class ShadowRoot : DocumentFragment, DocumentOrShadowRoot {\n open val mode: ShadowRootMode\n open val host: Element\n override val fullscreenElement: Element?\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val

DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val

DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }n}\n\n/**\n * Exposes the JavaScript [Element](https://developer.mozilla.org/en/docs/Web/API/Element) to Kotlin\n *\npublic external abstract class Element : Node, ParentNode, NonDocumentTypeChildNode, ChildNode, Slotable, GeometryUtils, UnionElementOrHTMLCollection, UnionElementOrRadioNodeList, UnionElementOrMouseEvent, UnionElementOrProcessingInstruction {\n open val namespaceURI: String?\n open val prefix: String?\n open val localName: String\n open val tagName: String\n open var id: String\n open var className: String\n open val classList: DOMTokenList\n open var slot: String\n open val attributes: NamedNodeMap\n open val shadowRoot: ShadowRoot?\n open var scrollTop: Double\n open var scrollLeft: Double\n open val scrollWidth: Int\n open val scrollHeight: Int\n open val clientTop: Int\n open val clientLeft: Int\n open val clientWidth: Int\n open val clientHeight: Int\n open var innerHTML: String\n open var outerHTML: String\n\n fun hasAttributes(): Boolean\n fun getAttributeNames(): Array<String>\n fun getAttribute(qualifiedName: String): String?\n fun getAttributeNS(namespace: String?, localName: String): String?\n fun setAttribute(qualifiedName: String, value: String)\n fun setAttributeNS(namespace: String?, qualifiedName: String, value: String)\n fun removeAttribute(qualifiedName: String)\n fun removeAttributeNS(namespace: String?, localName: String)\n fun hasAttribute(qualifiedName: String): Boolean\n fun hasAttributeNS(namespace: String?, localName: String): Boolean\n fun getAttributeNode(qualifiedName: String): Attr?\n fun getAttributeNodeNS(namespace: String?, localName: String): Attr?\n fun setAttributeNode(attr: Attr): Attr?\n fun setAttributeNodeNS(attr: Attr): Attr?\n fun removeAttributeNode(attr: Attr): Attr\n fun

```

attachShadow(init: ShadowRootInit): ShadowRoot\n fun closest(selectors: String): Element?\n fun
matches(selectors: String): Boolean\n fun webkitMatchesSelector(selectors: String): Boolean\n fun
getElementsByTagName(qualifiedName: String): HTMLCollection\n fun
getElementsByTagNameNS(namespace: String?, localName: String): HTMLCollection\n fun
getElementsByClassName(classNames: String): HTMLCollection\n fun insertAdjacentElement(where: String,
element: Element): Element?\n fun insertAdjacentText(where: String, data: String)\n fun getClientRects():
Array<DOMRect>\n fun getBoundingClientRect(): DOMRect\n fun scrollIntoView()\n fun
scrollIntoView(arg: dynamic)\n fun scroll(options: ScrollToOptions = definedExternally)\n fun scroll(x: Double,
y: Double)\n fun scrollTo(options: ScrollToOptions = definedExternally)\n fun scrollTo(x: Double, y: Double)\n
fun scrollBy(options: ScrollToOptions = definedExternally)\n fun scrollBy(x: Double, y: Double)\n fun
insertAdjacentHTML(position: String, text: String)\n fun setPointerCapture(pointerId: Int)\n fun
releasePointerCapture(pointerId: Int)\n fun hasPointerCapture(pointerId: Int): Boolean\n fun requestFullscreen():
Promise<Unit>\n\n companion object {\n    val ELEMENT_NODE: Short\n    val ATTRIBUTE_NODE:
Short\n    val TEXT_NODE: Short\n    val CDATA_SECTION_NODE: Short\n    val
ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE: Short\n    val
PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val
DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n } }\n\npublic external interface
ShadowRootInit {\n    var mode: ShadowRootMode?\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun ShadowRootInit(mode:
ShadowRootMode?): ShadowRootInit {\n    val o = js("{}")\n    o["mode"] = mode\n    return o\n}\n\n/**\n *
Exposes the JavaScript [NamedNodeMap](https://developer.mozilla.org/en/docs/Web/API/NamedNodeMap) to
Kotlin\n */\npublic external abstract class NamedNodeMap : ItemArrayLike<Attr> {\n    fun
getNamedItemNS(namespace: String?, localName: String): Attr?\n    fun setNamedItem(attr: Attr): Attr?\n    fun
setNamedItemNS(attr: Attr): Attr?\n    fun removeNamedItem(qualifiedName: String): Attr\n    fun
removeNamedItemNS(namespace: String?, localName: String): Attr\n    override fun item(index: Int): Attr?\n    fun
getNamedItem(qualifiedName: String): Attr?\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun NamedNodeMap.get(index:
Int): Attr? = asDynamic()[index]\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun
NamedNodeMap.get(qualifiedName: String): Attr? = asDynamic()[qualifiedName]\n\n/**\n * Exposes the
JavaScript [Attr](https://developer.mozilla.org/en/docs/Web/API/Attr) to Kotlin\n */\npublic external abstract class
Attr : Node {\n    open val namespaceURI: String?\n    open val prefix: String?\n    open val localName: String\n
open val name: String\n    open var value: String\n    open val ownerElement: Element?\n    open val specified:
Boolean\n\n    companion object {\n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n
        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE:
Short\n        val ENTITY_NODE: Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val
COMMENT_NODE: Short\n        val DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n
        val DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    } }\n\n/**\n * Exposes the JavaScript
[CharacterData](https://developer.mozilla.org/en/docs/Web/API/CharacterData) to Kotlin\n */\npublic external

```

```

abstract class CharacterData : Node, NonDocumentTypeChildNode, ChildNode {\n  open val data: String\n  open
val length: Int\n  fun substringData(offset: Int, count: Int): String\n  fun appendData(data: String)\n  fun
insertData(offset: Int, data: String)\n  fun deleteData(offset: Int, count: Int)\n  fun replaceData(offset: Int, count:
Int, data: String)\n\n  companion object {\n    val ELEMENT_NODE: Short\n    val ATTRIBUTE_NODE:
Short\n    val TEXT_NODE: Short\n    val CDATA_SECTION_NODE: Short\n    val
ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE: Short\n    val
PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val
DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n  }\n\n  /**\n   * Exposes the JavaScript
[Text](https://developer.mozilla.org/en/docs/Web/API/Text) to Kotlin\n   */\n  public external open class Text(data:
String = definedExternally) : CharacterData, Slotable, GeometryUtils {\n    open val wholeText: String\n    override
val assignedSlot: HTMLSlotElement?\n    override val previousElementSibling: Element?\n    override val
nextElementSibling: Element?\n    fun splitText(offset: Int): Text\n    override fun getBoxQuads(options:
BoxQuadOptions /* = definedExternally */): Array<DOMQuad>\n    override fun convertQuadFromNode(quad:
dynamic, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMQuad\n    override
fun convertRectFromNode(rect: DOMRectReadOnly, from: dynamic, options: ConvertCoordinateOptions /* =
definedExternally */): DOMQuad\n    override fun convertPointFromNode(point: DOMPointInit, from: dynamic,
options: ConvertCoordinateOptions /* = definedExternally */): DOMPoint\n    override fun before(vararg nodes:
dynamic)\n    override fun after(vararg nodes: dynamic)\n    override fun replaceWith(vararg nodes: dynamic)\n
    override fun remove()\n\n    companion object {\n      val ELEMENT_NODE: Short\n      val
ATTRIBUTE_NODE: Short\n      val TEXT_NODE: Short\n      val CDATA_SECTION_NODE: Short\n      val
ENTITY_REFERENCE_NODE: Short\n      val ENTITY_NODE: Short\n      val
PROCESSING_INSTRUCTION_NODE: Short\n      val COMMENT_NODE: Short\n      val
DOCUMENT_NODE: Short\n      val DOCUMENT_TYPE_NODE: Short\n      val
DOCUMENT_FRAGMENT_NODE: Short\n      val NOTATION_NODE: Short\n      val
DOCUMENT_POSITION_DISCONNECTED: Short\n      val DOCUMENT_POSITION_PRECEDING: Short\n
      val DOCUMENT_POSITION_FOLLOWING: Short\n      val DOCUMENT_POSITION_CONTAINS: Short\n
      val DOCUMENT_POSITION_CONTAINED_BY: Short\n      val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n\n    /**\n     * Exposes the JavaScript
[CDATASection](https://developer.mozilla.org/en/docs/Web/API/CDATASection) to Kotlin\n     */\n    public external
open class CDATASection : Text {\n      companion object {\n        val ELEMENT_NODE: Short\n        val
ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val
ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val
PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n      }\n\n      /**\n       * Exposes the JavaScript
[ProcessingInstruction](https://developer.mozilla.org/en/docs/Web/API/ProcessingInstruction) to Kotlin\n       */\n      public external abstract class ProcessingInstruction : CharacterData, LinkStyle,
UnionElementOrProcessingInstruction {\n        open val target: String\n\n        companion object {\n          val
ELEMENT_NODE: Short\n          val ATTRIBUTE_NODE: Short\n          val TEXT_NODE: Short\n          val

```

```

CDATA_SECTION_NODE: Short\n    val ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE:
Short\n    val PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val
DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    } \n} \n \n /** \n * Exposes the JavaScript
[Comment](https://developer.mozilla.org/en/docs/Web/API/Comment) to Kotlin \n * \n public external open class
Comment(data: String = definedExternally) : CharacterData { \n    override val previousElementSibling: Element? \n
    override val nextElementSibling: Element? \n    override fun before(vararg nodes: dynamic) \n    override fun
after(vararg nodes: dynamic) \n    override fun replaceWith(vararg nodes: dynamic) \n    override fun remove() \n \n
companion object { \n    val ELEMENT_NODE: Short\n    val ATTRIBUTE_NODE: Short\n    val
TEXT_NODE: Short\n    val CDATA_SECTION_NODE: Short\n    val ENTITY_REFERENCE_NODE:
Short\n    val ENTITY_NODE: Short\n    val PROCESSING_INSTRUCTION_NODE: Short\n    val
COMMENT_NODE: Short\n    val DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n
    val DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    } \n} \n \n /** \n * Exposes the JavaScript
[Range](https://developer.mozilla.org/en/docs/Web/API/Range) to Kotlin \n * \n public external open class Range { \n
    open val startContainer: Node \n    open val startOffset: Int \n    open val endContainer: Node \n    open val
endOffset: Int \n    open val collapsed: Boolean \n    open val commonAncestorContainer: Node \n    fun setStart(node:
Node, offset: Int) \n    fun setEnd(node: Node, offset: Int) \n    fun setStartBefore(node: Node) \n    fun
setStartAfter(node: Node) \n    fun setEndBefore(node: Node) \n    fun setEndAfter(node: Node) \n    fun
collapse(toStart: Boolean = definedExternally) \n    fun selectNode(node: Node) \n    fun selectNodeContents(node:
Node) \n    fun compareBoundaryPoints(how: Short, sourceRange: Range): Short \n    fun deleteContents() \n    fun
extractContents(): DocumentFragment \n    fun cloneContents(): DocumentFragment \n    fun insertNode(node:
Node) \n    fun surroundContents(newParent: Node) \n    fun cloneRange(): Range \n    fun detach() \n    fun
isPointInRange(node: Node, offset: Int): Boolean \n    fun comparePoint(node: Node, offset: Int): Short \n    fun
intersectsNode(node: Node): Boolean \n    fun getClientRects(): Array<DOMRect> \n    fun
getBoundingClientRect(): DOMRect \n    fun createContextualFragment(fragment: String): DocumentFragment \n \n
companion object { \n    val START_TO_START: Short\n    val START_TO_END: Short\n    val
END_TO_END: Short\n    val END_TO_START: Short\n    } \n} \n \n /** \n * Exposes the JavaScript
[NodeIterator](https://developer.mozilla.org/en/docs/Web/API/NodeIterator) to Kotlin \n * \n public external abstract
class NodeIterator { \n    open val root: Node \n    open val referenceNode: Node \n    open val
pointerBeforeReferenceNode: Boolean \n    open val whatToShow: Int \n    open val filter: NodeFilter? \n    fun
nextNode(): Node? \n    fun previousNode(): Node? \n    fun detach() \n} \n \n /** \n * Exposes the JavaScript
[TreeWalker](https://developer.mozilla.org/en/docs/Web/API/TreeWalker) to Kotlin \n * \n public external abstract
class TreeWalker { \n    open val root: Node \n    open val whatToShow: Int \n    open val filter: NodeFilter? \n    open
var currentNode: Node \n    fun parentNode(): Node? \n    fun firstChild(): Node? \n    fun lastChild(): Node? \n    fun
previousSibling(): Node? \n    fun nextSibling(): Node? \n    fun previousNode(): Node? \n    fun nextNode():
Node? \n} \n \n /** \n * Exposes the JavaScript
[NodeFilter](https://developer.mozilla.org/en/docs/Web/API/NodeFilter) to Kotlin \n
* \n @Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE") \n public external interface NodeFilter { \n
    fun acceptNode(node: Node): Short \n \n    companion object { \n        val FILTER_ACCEPT: Short\n        val
FILTER_REJECT: Short\n        val FILTER_SKIP: Short\n        val SHOW_ALL: Int\n        val

```



```

definedExternally\n    set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ScrollOptions(behavior:
ScrollBehavior? = ScrollBehavior.AUTO): ScrollOptions {\n    val o = js(\"({})\")\n    o[\"behavior\"] = behavior\n    return o\n}\n\n/**\n * Exposes the JavaScript
[ScrollToOptions](https://developer.mozilla.org/en/docs/Web/API/ScrollToOptions) to Kotlin\n */\npublic external
interface ScrollToOptions : ScrollOptions {\n    var left: Double?\n    get() = definedExternally\n    set(value) =
definedExternally\n    var top: Double?\n    get() = definedExternally\n    set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ScrollToOptions(left: Double? =
undefined, top: Double? = undefined, behavior: ScrollBehavior? = ScrollBehavior.AUTO): ScrollToOptions {\n
    val o = js(\"({})\")\n    o[\"left\"] = left\n    o[\"top\"] = top\n    o[\"behavior\"] = behavior\n    return o\n}\n\n/**\n * Exposes the JavaScript [MediaQueryList](https://developer.mozilla.org/en/docs/Web/API/MediaQueryList) to
Kotlin\n */\npublic external abstract class MediaQueryList : EventTarget {\n    open val media: String\n    open val
matches: Boolean\n    open var onchange: ((Event) -> dynamic)?\n    fun addListener(listener: EventListener?)\n
fun addListener(listener: ((Event) -> Unit)?)\n    fun removeListener(listener: EventListener?)\n    fun
removeListener(listener: ((Event) -> Unit)?)\n}\n\n/**\n * Exposes the JavaScript
[MediaQueryListEvent](https://developer.mozilla.org/en/docs/Web/API/MediaQueryListEvent) to Kotlin\n */\npublic
external open class MediaQueryListEvent(type: String, eventInitDict: MediaQueryListEventInit =
definedExternally) : Event {\n    open val media: String\n    open val matches: Boolean\n\n    companion object {\n
        val NONE: Short\n        val CAPTURING_PHASE: Short\n        val AT_TARGET: Short\n        val
BUBBLING_PHASE: Short\n    }\n\n    public external interface MediaQueryListEventInit : EventInit {\n        var
media: String? /* = \"\" */\n        get() = definedExternally\n        set(value) = definedExternally\n        var
matches: Boolean? /* = false */\n        get() = definedExternally\n        set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun MediaQueryListEventInit(media:
String? = \"\", matches: Boolean? = false, bubbles: Boolean? = false, cancelable: Boolean? = false, composed:
Boolean? = false): MediaQueryListEventInit {\n    val o = js(\"({})\")\n    o[\"media\"] = media\n    o[\"matches\"] =
matches\n    o[\"bubbles\"] = bubbles\n    o[\"cancelable\"] = cancelable\n    o[\"composed\"] = composed\n    return
o\n}\n\n/**\n * Exposes the JavaScript [Screen](https://developer.mozilla.org/en/docs/Web/API/Screen) to Kotlin\n */\npublic
external abstract class Screen {\n    open val availWidth: Int\n    open val availHeight: Int\n    open val
width: Int\n    open val height: Int\n    open val colorDepth: Int\n    open val pixelDepth: Int\n}\n\n/**\n * Exposes
the JavaScript [CaretPosition](https://developer.mozilla.org/en/docs/Web/API/CaretPosition) to Kotlin\n */\npublic
external abstract class CaretPosition {\n    open val offsetNode: Node\n    open val offset: Int\n    fun
getClientRect(): DOMRect?\n}\n\npublic external interface ScrollIntoViewOptions : ScrollOptions {\n    var block:
ScrollLogicalPosition? /* = ScrollLogicalPosition.CENTER */\n    get() = definedExternally\n    set(value) =
definedExternally\n    var inline: ScrollLogicalPosition? /* = ScrollLogicalPosition.CENTER */\n    get() =
definedExternally\n    set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ScrollIntoViewOptions(block:
ScrollLogicalPosition? = ScrollLogicalPosition.CENTER, inline: ScrollLogicalPosition? =
ScrollLogicalPosition.CENTER, behavior: ScrollBehavior? = ScrollBehavior.AUTO): ScrollIntoViewOptions {\n
    val o = js(\"({})\")\n    o[\"block\"] = block\n    o[\"inline\"] = inline\n    o[\"behavior\"] = behavior\n    return
o\n}\n\npublic external interface BoxQuadOptions {\n    var box: CSSBoxType? /* = CSSBoxType.BORDER */\n    get() =
definedExternally\n    set(value) = definedExternally\n    var relativeTo: dynamic\n    get() =
definedExternally\n    set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun BoxQuadOptions(box: CSSBoxType?
= CSSBoxType.BORDER, relativeTo: dynamic = undefined): BoxQuadOptions {\n    val o = js(\"({})\")\n    o[\"box\"] =
box\n    o[\"relativeTo\"] = relativeTo\n    return o\n}\n\npublic external interface
ConvertCoordinateOptions {\n    var fromBox: CSSBoxType? /* = CSSBoxType.BORDER */\n    get() =

```

```

definedExternally\n    set(value) = definedExternally\n    var toBox: CSSBoxType? /* = CSSBoxType.BORDER
*/\n    get() = definedExternally\n    set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ConvertCoordinateOptions(fromBox:
CSSBoxType? = CSSBoxType.BORDER, toBox: CSSBoxType? = CSSBoxType.BORDER):
ConvertCoordinateOptions {\n    val o = js(\"({})\")\n    o[\"fromBox\"] = fromBox\n    o[\"toBox\"] = toBox\n    return o\n}\n\n/**\n * Exposes the JavaScript
[GeometryUtils](https://developer.mozilla.org/en/docs/Web/API/GeometryUtils) to Kotlin\n */\npublic external
interface GeometryUtils {\n    fun getBoxQuads(options: BoxQuadOptions = definedExternally):
Array<DOMQuad>\n    fun convertQuadFromNode(quad: dynamic, from: dynamic, options:
ConvertCoordinateOptions = definedExternally): DOMQuad\n    fun convertRectFromNode(rect:
DOMRectReadOnly, from: dynamic, options: ConvertCoordinateOptions = definedExternally): DOMQuad\n    fun
convertPointFromNode(point: DOMPointInit, from: dynamic, options: ConvertCoordinateOptions =
definedExternally): DOMPoint\n}\n\n/**\n * Exposes the JavaScript
[Touch](https://developer.mozilla.org/en/docs/Web/API/Touch) to Kotlin\n */\npublic external abstract class Touch
{\n    open val identifier: Int\n    open val target: EventTarget\n    open val screenX: Int\n    open val screenY: Int\n    open val clientX: Int\n    open val clientY: Int\n    open val pageX: Int\n    open val pageY: Int\n    open val region:
String?\n}\n\npublic external abstract class TouchList : ItemArrayLike<Touch> {\n    override fun item(index: Int):
Touch?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun TouchList.get(index: Int):
Touch? = asDynamic()[index]\n\npublic external open class TouchEvent : UIEvent {\n    open val touches:
TouchList\n    open val targetTouches: TouchList\n    open val changedTouches: TouchList\n    open val altKey:
Boolean\n    open val metaKey: Boolean\n    open val ctrlKey: Boolean\n    open val shiftKey: Boolean\n    companion object {\n        val NONE: Short\n        val CAPTURING_PHASE: Short\n        val AT_TARGET:
Short\n        val BUBBLING_PHASE: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[Image](https://developer.mozilla.org/en/docs/Web/API/Image) to Kotlin\n */\npublic external open class
Image(width: Int = definedExternally, height: Int = definedExternally) : HTMLElement {\n    override var
onabort: ((Event) -> dynamic)?\n    override var onblur: ((FocusEvent) -> dynamic)?\n    override var oncancel:
((Event) -> dynamic)?\n    override var oncanplay: ((Event) -> dynamic)?\n    override var oncanplaythrough:
((Event) -> dynamic)?\n    override var onchange: ((Event) -> dynamic)?\n    override var onclick: ((MouseEvent) ->
dynamic)?\n    override var onclose: ((Event) -> dynamic)?\n    override var oncontextmenu: ((MouseEvent) ->
dynamic)?\n    override var oncuechange: ((Event) -> dynamic)?\n    override var ondblclick: ((MouseEvent) ->
dynamic)?\n    override var ondrag: ((DragEvent) -> dynamic)?\n    override var ondragend: ((DragEvent) ->
dynamic)?\n    override var ondragenter: ((DragEvent) -> dynamic)?\n    override var ondragexit: ((DragEvent) ->
dynamic)?\n    override var ondragleave: ((DragEvent) -> dynamic)?\n    override var ondragover: ((DragEvent) ->
dynamic)?\n    override var ondragstart: ((DragEvent) -> dynamic)?\n    override var ondrop: ((DragEvent) ->
dynamic)?\n    override var ondurationchange: ((Event) -> dynamic)?\n    override var onemptied: ((Event) ->
dynamic)?\n    override var onended: ((Event) -> dynamic)?\n    override var onerror: ((dynamic, String, Int, Int,
Any?) -> dynamic)?\n    override var onfocus: ((FocusEvent) -> dynamic)?\n    override var oninput: ((InputEvent) ->
dynamic)?\n    override var oninvalid: ((Event) -> dynamic)?\n    override var onkeydown: ((KeyboardEvent) ->
dynamic)?\n    override var onkeypress: ((KeyboardEvent) -> dynamic)?\n    override var onkeyup:
((KeyboardEvent) -> dynamic)?\n    override var onload: ((Event) -> dynamic)?\n    override var onloadeddata:
((Event) -> dynamic)?\n    override var onloadedmetadata: ((Event) -> dynamic)?\n    override var onloadend:
((Event) -> dynamic)?\n    override var onloadstart: ((ProgressEvent) -> dynamic)?\n    override var onmousedown:
((MouseEvent) -> dynamic)?\n    override var onmouseenter: ((MouseEvent) -> dynamic)?\n    override var
onmouseleave: ((MouseEvent) -> dynamic)?\n    override var onmousemove: ((MouseEvent) -> dynamic)?\n    override var
onmouseout: ((MouseEvent) -> dynamic)?\n    override var onmouseover: ((MouseEvent) ->
dynamic)?\n    override var onmouseup: ((MouseEvent) -> dynamic)?\n    override var onwheel: ((WheelEvent) ->

```

```

dynamic)?\n  override var onpause: ((Event) -> dynamic)?\n  override var onplay: ((Event) -> dynamic)?\n
override var onplaying: ((Event) -> dynamic)?\n  override var onprogress: ((ProgressEvent) -> dynamic)?\n
override var onratechange: ((Event) -> dynamic)?\n  override var onreset: ((Event) -> dynamic)?\n  override var
onresize: ((Event) -> dynamic)?\n  override var onscroll: ((Event) -> dynamic)?\n  override var onseeked:
((Event) -> dynamic)?\n  override var onseeking: ((Event) -> dynamic)?\n  override var onselect: ((Event) ->
dynamic)?\n  override var onshow: ((Event) -> dynamic)?\n  override var onstalled: ((Event) -> dynamic)?\n
override var onsubmit: ((Event) -> dynamic)?\n  override var onsuspend: ((Event) -> dynamic)?\n  override var
ontimeupdate: ((Event) -> dynamic)?\n  override var ontoggle: ((Event) -> dynamic)?\n  override var
onvolumechange: ((Event) -> dynamic)?\n  override var onwaiting: ((Event) -> dynamic)?\n  override var
ongotpointercapture: ((PointerEvent) -> dynamic)?\n  override var onlostpointercapture: ((PointerEvent) ->
dynamic)?\n  override var onpointerdown: ((PointerEvent) -> dynamic)?\n  override var onpointermove:
((PointerEvent) -> dynamic)?\n  override var onpointerup: ((PointerEvent) -> dynamic)?\n  override var
onpointercancel: ((PointerEvent) -> dynamic)?\n  override var onpointerover: ((PointerEvent) -> dynamic)?\n
override var onpointerout: ((PointerEvent) -> dynamic)?\n  override var onpointerenter: ((PointerEvent) ->
dynamic)?\n  override var onpointerleave: ((PointerEvent) -> dynamic)?\n  override var oncopy:
((ClipboardEvent) -> dynamic)?\n  override var oncut: ((ClipboardEvent) -> dynamic)?\n  override var onpaste:
((ClipboardEvent) -> dynamic)?\n  override var contentEditable: String\n  override val isContentEditable:
Boolean\n  override val style: CSSStyleDeclaration\n  override val children: HTMLCollection\n  override val
firstElementChild: Element?\n  override val lastElementChild: Element?\n  override val childElementCount: Int\n
  override val previousElementSibling: Element?\n  override val nextElementSibling: Element?\n  override val
assignedSlot: HTMLSlotElement?\n  override fun prepend(vararg nodes: dynamic)\n  override fun append(vararg
nodes: dynamic)\n  override fun querySelector(selectors: String): Element?\n  override fun
querySelectorAll(selectors: String): NodeList\n  override fun before(vararg nodes: dynamic)\n  override fun
after(vararg nodes: dynamic)\n  override fun replaceWith(vararg nodes: dynamic)\n  override fun remove()\n
override fun getBoxQuads(options: BoxQuadOptions /* = definedExternally */): Array<DOMQuad>\n  override
fun convertQuadFromNode(quad: dynamic, from: dynamic, options: ConvertCoordinateOptions /* =
definedExternally */): DOMQuad\n  override fun convertRectFromNode(rect: DOMRectReadOnly, from:
dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMQuad\n  override fun
convertPointFromNode(point: DOMPointInit, from: dynamic, options: ConvertCoordinateOptions /* =
definedExternally */): DOMPoint\n\n  companion object {\n    val ELEMENT_NODE: Short\n    val
ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n    val CDATA_SECTION_NODE: Short\n    val
ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE: Short\n    val
PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val
DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n  }\n\n  public external open class
Audio(src: String = definedExternally) : HTMLAudioElement {\n  override var onabort: ((Event) -> dynamic)?\n
override var onblur: ((FocusEvent) -> dynamic)?\n  override var oncancel: ((Event) -> dynamic)?\n  override var
oncanplay: ((Event) -> dynamic)?\n  override var oncanplaythrough: ((Event) -> dynamic)?\n  override var
onchange: ((Event) -> dynamic)?\n  override var onclick: ((MouseEvent) -> dynamic)?\n  override var onclose:
((Event) -> dynamic)?\n  override var oncontextmenu: ((MouseEvent) -> dynamic)?\n  override var oncuechange:
((Event) -> dynamic)?\n  override var ondblclick: ((MouseEvent) -> dynamic)?\n  override var ondrag:
((DragEvent) -> dynamic)?\n  override var ondragend: ((DragEvent) -> dynamic)?\n  override var ondragenter:
((DragEvent) -> dynamic)?\n  override var ondragexit: ((DragEvent) -> dynamic)?\n  override var ondragleave:
((DragEvent) -> dynamic)?\n  override var ondragover: ((DragEvent) -> dynamic)?\n  override var ondragstart:

```

```

((DragEvent) -> dynamic)?\n  override var ondrop: ((DragEvent) -> dynamic)?\n  override var ondurationchange:
((Event) -> dynamic)?\n  override var onemptied: ((Event) -> dynamic)?\n  override var onended: ((Event) ->
dynamic)?\n  override var onerror: ((dynamic, String, Int, Int, Any?) -> dynamic)?\n  override var onfocus:
((FocusEvent) -> dynamic)?\n  override var oninput: ((InputEvent) -> dynamic)?\n  override var oninvalid:
((Event) -> dynamic)?\n  override var onkeydown: ((KeyboardEvent) -> dynamic)?\n  override var onkeypress:
((KeyboardEvent) -> dynamic)?\n  override var onkeyup: ((KeyboardEvent) -> dynamic)?\n  override var onload:
((Event) -> dynamic)?\n  override var onloadeddata: ((Event) -> dynamic)?\n  override var onloadedmetadata:
((Event) -> dynamic)?\n  override var onloadend: ((Event) -> dynamic)?\n  override var onloadstart:
((ProgressEvent) -> dynamic)?\n  override var onmousedown: ((MouseEvent) -> dynamic)?\n  override var
onmouseenter: ((MouseEvent) -> dynamic)?\n  override var onmouseleave: ((MouseEvent) -> dynamic)?\n
override var onmousemove: ((MouseEvent) -> dynamic)?\n  override var onmouseout: ((MouseEvent) ->
dynamic)?\n  override var onmouseover: ((MouseEvent) -> dynamic)?\n  override var onmouseup: ((MouseEvent)
-> dynamic)?\n  override var onwheel: ((WheelEvent) -> dynamic)?\n  override var onpause: ((Event) ->
dynamic)?\n  override var onplay: ((Event) -> dynamic)?\n  override var onplaying: ((Event) -> dynamic)?\n
override var onprogress: ((ProgressEvent) -> dynamic)?\n  override var onratechange: ((Event) -> dynamic)?\n
override var onreset: ((Event) -> dynamic)?\n  override var onresize: ((Event) -> dynamic)?\n  override var
onscroll: ((Event) -> dynamic)?\n  override var onseeked: ((Event) -> dynamic)?\n  override var onseeking:
((Event) -> dynamic)?\n  override var onselect: ((Event) -> dynamic)?\n  override var onshow: ((Event) ->
dynamic)?\n  override var onstalled: ((Event) -> dynamic)?\n  override var onsubmit: ((Event) -> dynamic)?\n
override var onsuspend: ((Event) -> dynamic)?\n  override var ontimeupdate: ((Event) -> dynamic)?\n  override
var ontoggle: ((Event) -> dynamic)?\n  override var onvolumechange: ((Event) -> dynamic)?\n  override var
onwaiting: ((Event) -> dynamic)?\n  override var ongotpointercapture: ((PointerEvent) -> dynamic)?\n  override
var onlostpointercapture: ((PointerEvent) -> dynamic)?\n  override var onpointerdown: ((PointerEvent) ->
dynamic)?\n  override var onpointermove: ((PointerEvent) -> dynamic)?\n  override var onpointerup:
((PointerEvent) -> dynamic)?\n  override var onpointercancel: ((PointerEvent) -> dynamic)?\n  override var
onpointerover: ((PointerEvent) -> dynamic)?\n  override var onpointerout: ((PointerEvent) -> dynamic)?\n
override var onpointerenter: ((PointerEvent) -> dynamic)?\n  override var onpointerleave: ((PointerEvent) ->
dynamic)?\n  override var oncopy: ((ClipboardEvent) -> dynamic)?\n  override var oncut: ((ClipboardEvent) ->
dynamic)?\n  override var onpaste: ((ClipboardEvent) -> dynamic)?\n  override var contentEditable: String\n
override val isContentEditable: Boolean\n  override val style: CSSStyleDeclaration\n  override val children:
HTMLCollection\n  override val firstElementChild: Element?\n  override val lastElementChild: Element?\n
override val childElementCount: Int\n  override val previousElementSibling: Element?\n  override val
nextElementSibling: Element?\n  override val assignedSlot: HTMLSlotElement?\n  override fun prepend(vararg
nodes: dynamic)\n  override fun append(vararg nodes: dynamic)\n  override fun querySelector(selectors: String):
Element?\n  override fun querySelectorAll(selectors: String): NodeList\n  override fun before(vararg nodes:
dynamic)\n  override fun after(vararg nodes: dynamic)\n  override fun replaceWith(vararg nodes: dynamic)\n
override fun remove()\n  override fun getBoxQuads(options: BoxQuadOptions /* = definedExternally */):
Array<DOMQuad>\n  override fun convertQuadFromNode(quad: dynamic, from: dynamic, options:
ConvertCoordinateOptions /* = definedExternally */): DOMQuad\n  override fun convertRectFromNode(rect:
DOMRectReadOnly, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMQuad\n
override fun convertPointFromNode(point: DOMPointInit, from: dynamic, options: ConvertCoordinateOptions /* =
definedExternally */): DOMPoint\n\n  companion object {\n    val NETWORK_EMPTY: Short\n    val
NETWORK_IDLE: Short\n    val NETWORK_LOADING: Short\n    val NETWORK_NO_SOURCE: Short\n
    val HAVE_NOTHING: Short\n    val HAVE_METADATA: Short\n    val HAVE_CURRENT_DATA:
Short\n    val HAVE_FUTURE_DATA: Short\n    val HAVE_ENOUGH_DATA: Short\n    val
ELEMENT_NODE: Short\n    val ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n    val
CDATA_SECTION_NODE: Short\n    val ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE:
Short\n    val PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val

```

DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n } \n} \n \n /** \n * Exposes the JavaScript
[Option](https://developer.mozilla.org/en/docs/Web/API/Option) to Kotlin \n * \n public external open class
Option(text: String = definedExternally, value: String = definedExternally, defaultSelected: Boolean =
definedExternally, selected: Boolean = definedExternally) : HTMLInputElement { \n override var onabort:
((Event) -> dynamic)? \n override var onblur: ((FocusEvent) -> dynamic)? \n override var oncancel: ((Event) ->
dynamic)? \n override var oncanplay: ((Event) -> dynamic)? \n override var oncanplaythrough: ((Event) ->
dynamic)? \n override var onchange: ((Event) -> dynamic)? \n override var onclick: ((MouseEvent) ->
dynamic)? \n override var onclose: ((Event) -> dynamic)? \n override var oncontextmenu: ((MouseEvent) ->
dynamic)? \n override var oncuechange: ((Event) -> dynamic)? \n override var ondblclick: ((MouseEvent) ->
dynamic)? \n override var ondrag: ((DragEvent) -> dynamic)? \n override var ondragend: ((DragEvent) ->
dynamic)? \n override var ondragenter: ((DragEvent) -> dynamic)? \n override var ondragexit: ((DragEvent) ->
dynamic)? \n override var ondragleave: ((DragEvent) -> dynamic)? \n override var ondragover: ((DragEvent) ->
dynamic)? \n override var ondragstart: ((DragEvent) -> dynamic)? \n override var ondrop: ((DragEvent) ->
dynamic)? \n override var ondurationchange: ((Event) -> dynamic)? \n override var onemptied: ((Event) ->
dynamic)? \n override var onended: ((Event) -> dynamic)? \n override var onerror: ((dynamic, String, Int, Int,
Any?) -> dynamic)? \n override var onfocus: ((FocusEvent) -> dynamic)? \n override var oninput: ((InputEvent) -
> dynamic)? \n override var oninvalid: ((Event) -> dynamic)? \n override var onkeydown: ((KeyboardEvent) ->
dynamic)? \n override var onkeypress: ((KeyboardEvent) -> dynamic)? \n override var onkeyup:
((KeyboardEvent) -> dynamic)? \n override var onload: ((Event) -> dynamic)? \n override var onloadeddata:
((Event) -> dynamic)? \n override var onloadedmetadata: ((Event) -> dynamic)? \n override var onloadend:
((Event) -> dynamic)? \n override var onloadstart: ((ProgressEvent) -> dynamic)? \n override var onmousedown:
((MouseEvent) -> dynamic)? \n override var onmouseenter: ((MouseEvent) -> dynamic)? \n override var
onmouseleave: ((MouseEvent) -> dynamic)? \n override var onmousemove: ((MouseEvent) -> dynamic)? \n
override var onmouseout: ((MouseEvent) -> dynamic)? \n override var onmouseover: ((MouseEvent) ->
dynamic)? \n override var onmouseup: ((MouseEvent) -> dynamic)? \n override var onwheel: ((WheelEvent) ->
dynamic)? \n override var onpause: ((Event) -> dynamic)? \n override var onplay: ((Event) -> dynamic)? \n
override var onplaying: ((Event) -> dynamic)? \n override var onprogress: ((ProgressEvent) -> dynamic)? \n
override var onratechange: ((Event) -> dynamic)? \n override var onreset: ((Event) -> dynamic)? \n override var
onresize: ((Event) -> dynamic)? \n override var onscroll: ((Event) -> dynamic)? \n override var onseeked:
((Event) -> dynamic)? \n override var onseeking: ((Event) -> dynamic)? \n override var onselect: ((Event) ->
dynamic)? \n override var onshow: ((Event) -> dynamic)? \n override var onstalled: ((Event) -> dynamic)? \n
override var onsubmit: ((Event) -> dynamic)? \n override var onsuspend: ((Event) -> dynamic)? \n override var
ontimeupdate: ((Event) -> dynamic)? \n override var ontoggle: ((Event) -> dynamic)? \n override var
onvolumechange: ((Event) -> dynamic)? \n override var onwaiting: ((Event) -> dynamic)? \n override var
ongotpointercapture: ((PointerEvent) -> dynamic)? \n override var onlostpointercapture: ((PointerEvent) ->
dynamic)? \n override var onpointerdown: ((PointerEvent) -> dynamic)? \n override var onpointermove:
((PointerEvent) -> dynamic)? \n override var onpointerup: ((PointerEvent) -> dynamic)? \n override var
onpointercancel: ((PointerEvent) -> dynamic)? \n override var onpointerover: ((PointerEvent) -> dynamic)? \n
override var onpointerout: ((PointerEvent) -> dynamic)? \n override var onpointerenter: ((PointerEvent) ->
dynamic)? \n override var onpointerleave: ((PointerEvent) -> dynamic)? \n override var oncopy:
((ClipboardEvent) -> dynamic)? \n override var oncut: ((ClipboardEvent) -> dynamic)? \n override var onpaste:
((ClipboardEvent) -> dynamic)? \n override var contentEditable: String \n override val isContentEditable:
Boolean \n override val style: CSSStyleDeclaration \n override val children: HTMLCollection \n override val

```

firstElementChild: Element? override val lastElementChild: Element? override val childElementCount: Int override val previousElementSibling: Element? override val nextElementSibling: Element? override val assignedSlot: HTMLSlotElement? override fun prepend(vararg nodes: dynamic) override fun append(vararg nodes: dynamic) override fun querySelector(selectors: String): Element? override fun querySelectorAll(selectors: String): NodeList override fun before(vararg nodes: dynamic) override fun after(vararg nodes: dynamic) override fun replaceWith(vararg nodes: dynamic) override fun remove() override fun getBoxQuads(options: BoxQuadOptions /* = definedExternally */): Array<DOMQuad> override fun convertQuadFromNode(quad: dynamic, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMQuad override fun convertRectFromNode(rect: DOMRectReadOnly, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMQuad override fun convertPointFromNode(point: DOMPointInit, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMPoint companion object { val ELEMENT_NODE: Short val ATTRIBUTE_NODE: Short val TEXT_NODE: Short val CDATA_SECTION_NODE: Short val ENTITY_REFERENCE_NODE: Short val ENTITY_NODE: Short val PROCESSING_INSTRUCTION_NODE: Short val COMMENT_NODE: Short val DOCUMENT_NODE: Short val DOCUMENT_TYPE_NODE: Short val DOCUMENT_FRAGMENT_NODE: Short val NOTATION_NODE: Short val DOCUMENT_POSITION_DISCONNECTED: Short val DOCUMENT_POSITION_PRECEDING: Short val DOCUMENT_POSITION_FOLLOWING: Short val DOCUMENT_POSITION_CONTAINS: Short val DOCUMENT_POSITION_CONTAINED_BY: Short val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short } public external interface UnionElementOrHTMLCollection public external interface UnionElementOrRadioNodeList public external interface UnionHTMLOptGroupElementOrHTMLOptionElement public external interface UnionAudioTrackOrTextTrackOrVideoTrack public external interface UnionElementOrMouseEvent public external interface UnionMessagePortOrWindowProxy public external interface MediaProvider public external interface RenderingContext public external interface HTMLOrSVGImageElement : CanvasImageSource public external interface CanvasImageSource : ImageBitmapSource public external interface ImageBitmapSource public external interface HTMLOrSVGScriptElement n/* please, don't implement this interface! */ n@JsName("null") n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE") public external interface DocumentReadyState { companion object } public inline val DocumentReadyState.Companion.LOADING: DocumentReadyState get() = "loading".asDynamic().unsafeCast<DocumentReadyState>() public inline val DocumentReadyState.Companion.INTERACTIVE: DocumentReadyState get() = "interactive".asDynamic().unsafeCast<DocumentReadyState>() public inline val DocumentReadyState.Companion.COMPLETE: DocumentReadyState get() = "complete".asDynamic().unsafeCast<DocumentReadyState>() n/* please, don't implement this interface! */ n@JsName("null") n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE") public external interface CanPlayTypeResult { companion object } public inline val CanPlayTypeResult.Companion.EMPTY: CanPlayTypeResult get() = "" public inline val CanPlayTypeResult.Companion.MAYBE: CanPlayTypeResult get() = "maybe".asDynamic().unsafeCast<CanPlayTypeResult>() public inline val CanPlayTypeResult.Companion.PROBABLY: CanPlayTypeResult get() = "probably".asDynamic().unsafeCast<CanPlayTypeResult>() n/* please, don't implement this interface! */ n@JsName("null") n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE") public external interface TextTrackMode { companion object } public inline val TextTrackMode.Companion.DISABLED: TextTrackMode get() = "disabled".asDynamic().unsafeCast<TextTrackMode>() public inline val TextTrackMode.Companion.HIDDEN: TextTrackMode get() =

```

```

"hidden".asDynamic().unsafeCast<TextTrackMode>()\n\npublic inline val
TextTrackMode.Companion.SHOWING: TextTrackMode get() =
"showing".asDynamic().unsafeCast<TextTrackMode>()\n\n/* please, don't implement this interface!
*\n\n@JsName("null")\n\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\n\npublic external
interface TextTrackKind {\n\n    companion object\n\n}\n\npublic inline val TextTrackKind.Companion.SUBTITLES:
TextTrackKind get() = "subtitles".asDynamic().unsafeCast<TextTrackKind>()\n\npublic inline val
TextTrackKind.Companion.CAPTIONS: TextTrackKind get() =
"captions".asDynamic().unsafeCast<TextTrackKind>()\n\npublic inline val
TextTrackKind.Companion.DESCRPTIONS: TextTrackKind get() =
"descriptions".asDynamic().unsafeCast<TextTrackKind>()\n\npublic inline val
TextTrackKind.Companion.CHAPTERS: TextTrackKind get() =
"chapters".asDynamic().unsafeCast<TextTrackKind>()\n\npublic inline val
TextTrackKind.Companion.METADATA: TextTrackKind get() =
"metadata".asDynamic().unsafeCast<TextTrackKind>()\n\n/* please, don't implement this interface!
*\n\n@JsName("null")\n\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\n\npublic external
interface SelectionMode {\n\n    companion object\n\n}\n\npublic inline val SelectionMode.Companion.SELECT:
SelectionMode get() = "select".asDynamic().unsafeCast<SelectionMode>()\n\npublic inline val
SelectionMode.Companion.START: SelectionMode get() =
"start".asDynamic().unsafeCast<SelectionMode>()\n\npublic inline val SelectionMode.Companion.END:
SelectionMode get() = "end".asDynamic().unsafeCast<SelectionMode>()\n\npublic inline val
SelectionMode.Companion.PRESERVE: SelectionMode get() =
"preserve".asDynamic().unsafeCast<SelectionMode>()\n\n/* please, don't implement this interface!
*\n\n@JsName("null")\n\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\n\npublic external
interface CanvasFillRule {\n\n    companion object\n\n}\n\npublic inline val CanvasFillRule.Companion.NONZERO:
CanvasFillRule get() = "nonzero".asDynamic().unsafeCast<CanvasFillRule>()\n\npublic inline val
CanvasFillRule.Companion.EVENODD: CanvasFillRule get() =
"evenodd".asDynamic().unsafeCast<CanvasFillRule>()\n\n/* please, don't implement this interface!
*\n\n@JsName("null")\n\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\n\npublic external
interface ImageSmoothingQuality {\n\n    companion object\n\n}\n\npublic inline val
ImageSmoothingQuality.Companion.LOW: ImageSmoothingQuality get() =
"low".asDynamic().unsafeCast<ImageSmoothingQuality>()\n\npublic inline val
ImageSmoothingQuality.Companion.MEDIUM: ImageSmoothingQuality get() =
"medium".asDynamic().unsafeCast<ImageSmoothingQuality>()\n\npublic inline val
ImageSmoothingQuality.Companion.HIGH: ImageSmoothingQuality get() =
"high".asDynamic().unsafeCast<ImageSmoothingQuality>()\n\n/* please, don't implement this interface!
*\n\n@JsName("null")\n\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\n\npublic external
interface CanvasLineCap {\n\n    companion object\n\n}\n\npublic inline val CanvasLineCap.Companion.BUTT:
CanvasLineCap get() = "butt".asDynamic().unsafeCast<CanvasLineCap>()\n\npublic inline val
CanvasLineCap.Companion.ROUND: CanvasLineCap get() =
"round".asDynamic().unsafeCast<CanvasLineCap>()\n\npublic inline val CanvasLineCap.Companion.SQUARE:
CanvasLineCap get() = "square".asDynamic().unsafeCast<CanvasLineCap>()\n\n/* please, don't implement this
interface! *\n\n@JsName("null")\n\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\n\npublic
external interface CanvasLineJoin {\n\n    companion object\n\n}\n\npublic inline val
CanvasLineJoin.Companion.ROUND: CanvasLineJoin get() =
"round".asDynamic().unsafeCast<CanvasLineJoin>()\n\npublic inline val CanvasLineJoin.Companion.BEVEL:
CanvasLineJoin get() = "bevel".asDynamic().unsafeCast<CanvasLineJoin>()\n\npublic inline val
CanvasLineJoin.Companion.MITER: CanvasLineJoin get() =
"miter".asDynamic().unsafeCast<CanvasLineJoin>()\n\n/* please, don't implement this interface!

```

```

*\n@jsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\npublic external
interface CanvasTextAlign {\n  companion object\n}\n\npublic inline val CanvasTextAlign.Companion.START:
CanvasTextAlign get() = "start".asDynamic().unsafeCast<CanvasTextAlign>()\n\npublic inline val
CanvasTextAlign.Companion.END: CanvasTextAlign get() =
"end".asDynamic().unsafeCast<CanvasTextAlign>()\n\npublic inline val CanvasTextAlign.Companion.LEFT:
CanvasTextAlign get() = "left".asDynamic().unsafeCast<CanvasTextAlign>()\n\npublic inline val
CanvasTextAlign.Companion.RIGHT: CanvasTextAlign get() =
"right".asDynamic().unsafeCast<CanvasTextAlign>()\n\npublic inline val
CanvasTextAlign.Companion.CENTER: CanvasTextAlign get() =
"center".asDynamic().unsafeCast<CanvasTextAlign>()\n\n/* please, don't implement this interface!
*\n@jsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\npublic external
interface CanvasTextBaseline {\n  companion object\n}\n\npublic inline val CanvasTextBaseline.Companion.TOP:
CanvasTextBaseline get() = "top".asDynamic().unsafeCast<CanvasTextBaseline>()\n\npublic inline val
CanvasTextBaseline.Companion.HANGING: CanvasTextBaseline get() =
"hanging".asDynamic().unsafeCast<CanvasTextBaseline>()\n\npublic inline val
CanvasTextBaseline.Companion.MIDDLE: CanvasTextBaseline get() =
"middle".asDynamic().unsafeCast<CanvasTextBaseline>()\n\npublic inline val
CanvasTextBaseline.Companion.ALPHABETIC: CanvasTextBaseline get() =
"alphabetic".asDynamic().unsafeCast<CanvasTextBaseline>()\n\npublic inline val
CanvasTextBaseline.Companion.IDEOGRAPHIC: CanvasTextBaseline get() =
"ideographic".asDynamic().unsafeCast<CanvasTextBaseline>()\n\npublic inline val
CanvasTextBaseline.Companion.BOTTOM: CanvasTextBaseline get() =
"bottom".asDynamic().unsafeCast<CanvasTextBaseline>()\n\n/* please, don't implement this interface!
*\n@jsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\npublic external
interface CanvasDirection {\n  companion object\n}\n\npublic inline val CanvasDirection.Companion.LTR:
CanvasDirection get() = "ltr".asDynamic().unsafeCast<CanvasDirection>()\n\npublic inline val
CanvasDirection.Companion.RTL: CanvasDirection get() =
"rtl".asDynamic().unsafeCast<CanvasDirection>()\n\npublic inline val CanvasDirection.Companion.INHERIT:
CanvasDirection get() = "inherit".asDynamic().unsafeCast<CanvasDirection>()\n\n/* please, don't implement this
interface! *\n@jsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\npublic
external interface ScrollRestoration {\n  companion object\n}\n\npublic inline val
ScrollRestoration.Companion.AUTO: ScrollRestoration get() =
"auto".asDynamic().unsafeCast<ScrollRestoration>()\n\npublic inline val
ScrollRestoration.Companion.MANUAL: ScrollRestoration get() =
"manual".asDynamic().unsafeCast<ScrollRestoration>()\n\n/* please, don't implement this interface!
*\n@jsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\npublic external
interface ImageOrientation {\n  companion object\n}\n\npublic inline val ImageOrientation.Companion.NONE:
ImageOrientation get() = "none".asDynamic().unsafeCast<ImageOrientation>()\n\npublic inline val
ImageOrientation.Companion.FLIPY: ImageOrientation get() =
"flipY".asDynamic().unsafeCast<ImageOrientation>()\n\n/* please, don't implement this interface!
*\n@jsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\npublic external
interface PremultiplyAlpha {\n  companion object\n}\n\npublic inline val PremultiplyAlpha.Companion.NONE:
PremultiplyAlpha get() = "none".asDynamic().unsafeCast<PremultiplyAlpha>()\n\npublic inline val
PremultiplyAlpha.Companion.PREMULTIPLY: PremultiplyAlpha get() =
"premultiply".asDynamic().unsafeCast<PremultiplyAlpha>()\n\npublic inline val
PremultiplyAlpha.Companion.DEFAULT: PremultiplyAlpha get() =
"default".asDynamic().unsafeCast<PremultiplyAlpha>()\n\n/* please, don't implement this interface!
*\n@jsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\npublic external

```

```

interface ColorSpaceConversion {\n  companion object\n}\n\npublic inline val
ColorSpaceConversion.Companion.NONE: ColorSpaceConversion get() =
\"none\".asDynamic().unsafeCast<ColorSpaceConversion>()\n\npublic inline val
ColorSpaceConversion.Companion.DEFAULT: ColorSpaceConversion get() =
\"default\".asDynamic().unsafeCast<ColorSpaceConversion>()\n\n/* please, don't implement this interface!
*\n\n@JsName(\"null\")\n\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic external
interface ResizeQuality {\n  companion object\n}\n\npublic inline val ResizeQuality.Companion.PIXELATED:
ResizeQuality get() = \"pixelated\".asDynamic().unsafeCast<ResizeQuality>()\n\npublic inline val
ResizeQuality.Companion.LOW: ResizeQuality get() =
\"low\".asDynamic().unsafeCast<ResizeQuality>()\n\npublic inline val ResizeQuality.Companion.MEDIUM:
ResizeQuality get() = \"medium\".asDynamic().unsafeCast<ResizeQuality>()\n\npublic inline val
ResizeQuality.Companion.HIGH: ResizeQuality get() = \"high\".asDynamic().unsafeCast<ResizeQuality>()\n\n/*
please, don't implement this interface!
*\n\n@JsName(\"null\")\n\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic external
interface BinaryType {\n  companion object\n}\n\npublic inline val BinaryType.Companion.BLOB: BinaryType
get() = \"blob\".asDynamic().unsafeCast<BinaryType>()\n\npublic inline val
BinaryType.Companion.ARRAYBUFFER: BinaryType get() =
\"arraybuffer\".asDynamic().unsafeCast<BinaryType>()\n\n/* please, don't implement this interface!
*\n\n@JsName(\"null\")\n\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic external
interface WorkerType {\n  companion object\n}\n\npublic inline val WorkerType.Companion.CLASSIC:
WorkerType get() = \"classic\".asDynamic().unsafeCast<WorkerType>()\n\npublic inline val
WorkerType.Companion.MODULE: WorkerType get() =
\"module\".asDynamic().unsafeCast<WorkerType>()\n\n/* please, don't implement this interface!
*\n\n@JsName(\"null\")\n\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic external
interface ShadowRootMode {\n  companion object\n}\n\npublic inline val ShadowRootMode.Companion.OPEN:
ShadowRootMode get() = \"open\".asDynamic().unsafeCast<ShadowRootMode>()\n\npublic inline val
ShadowRootMode.Companion.CLOSED: ShadowRootMode get() =
\"closed\".asDynamic().unsafeCast<ShadowRootMode>()\n\n/* please, don't implement this interface!
*\n\n@JsName(\"null\")\n\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic external
interface ScrollBehavior {\n  companion object\n}\n\npublic inline val ScrollBehavior.Companion.AUTO:
ScrollBehavior get() = \"auto\".asDynamic().unsafeCast<ScrollBehavior>()\n\npublic inline val
ScrollBehavior.Companion.INSTANT: ScrollBehavior get() =
\"instant\".asDynamic().unsafeCast<ScrollBehavior>()\n\npublic inline val ScrollBehavior.Companion.SMOOTH:
ScrollBehavior get() = \"smooth\".asDynamic().unsafeCast<ScrollBehavior>()\n\n/* please, don't implement this
interface! *\n\n@JsName(\"null\")\n\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic
external interface ScrollLogicalPosition {\n  companion object\n}\n\npublic inline val
ScrollLogicalPosition.Companion.START: ScrollLogicalPosition get() =
\"start\".asDynamic().unsafeCast<ScrollLogicalPosition>()\n\npublic inline val
ScrollLogicalPosition.Companion.CENTER: ScrollLogicalPosition get() =
\"center\".asDynamic().unsafeCast<ScrollLogicalPosition>()\n\npublic inline val
ScrollLogicalPosition.Companion.END: ScrollLogicalPosition get() =
\"end\".asDynamic().unsafeCast<ScrollLogicalPosition>()\n\npublic inline val
ScrollLogicalPosition.Companion.NEAREST: ScrollLogicalPosition get() =
\"nearest\".asDynamic().unsafeCast<ScrollLogicalPosition>()\n\n/* please, don't implement this interface!
*\n\n@JsName(\"null\")\n\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic external
interface CSSBoxType {\n  companion object\n}\n\npublic inline val CSSBoxType.Companion.MARGIN:
CSSBoxType get() = \"margin\".asDynamic().unsafeCast<CSSBoxType>()\n\npublic inline val
CSSBoxType.Companion.BORDER: CSSBoxType get() =

```

```

\"border\".asDynamic().unsafeCast<CSSBoxType>()\n\npublic inline val CSSBoxType.Companion.PADDING:
CSSBoxType get() = \"padding\".asDynamic().unsafeCast<CSSBoxType>()\n\npublic inline val
CSSBoxType.Companion.CONTENT: CSSBoxType get() =
\"content\".asDynamic().unsafeCast<CSSBoxType>()\", /*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n *^\n\n// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!\n\n//
See github.com/kotlin/dukat for details\n\npackage org.w3c.fetch\n\nimport kotlin.js.*\nimport
org.khronos.webgl.*\nimport org.w3c.files.*\nimport org.w3c.xhr.*\n\n/**\n * Exposes the JavaScript
[Headers](https://developer.mozilla.org/en/docs/Web/API/Headers) to Kotlin\n *^\n\npublic external open class
Headers(init: dynamic = definedExternally) {\n fun append(name: String, value: String)\n fun delete(name:
String)\n fun get(name: String): String?\n fun has(name: String): Boolean\n fun set(name: String, value:
String)\n}\n\n/**\n * Exposes the JavaScript [Body](https://developer.mozilla.org/en/docs/Web/API/Body) to
Kotlin\n *^\n\npublic external interface Body {\n val bodyUsed: Boolean\n fun arrayBuffer():
Promise<ArrayBuffer>\n fun blob(): Promise<Blob>\n fun formData(): Promise<FormData>\n fun json():
Promise<Any?>\n fun text(): Promise<String>\n}\n\n/**\n * Exposes the JavaScript
[Request](https://developer.mozilla.org/en/docs/Web/API/Request) to Kotlin\n *^\n\npublic external open class
Request(input: dynamic, init: RequestInit = definedExternally) : Body {\n open val method: String\n open val
url: String\n open val headers: Headers\n open val type: RequestType\n open val destination:
RequestDestination\n open val referrer: String\n open val referrerPolicy: dynamic\n open val mode:
RequestMode\n open val credentials: RequestCredentials\n open val cache: RequestCache\n open val redirect:
RequestRedirect\n open val integrity: String\n open val keepalive: Boolean\n override val bodyUsed:
Boolean\n fun clone(): Request\n override fun arrayBuffer(): Promise<ArrayBuffer>\n override fun blob():
Promise<Blob>\n override fun formData(): Promise<FormData>\n override fun json(): Promise<Any?>\n
override fun text(): Promise<String>\n}\n\npublic external interface RequestInit {\n var method: String?\n
get() = definedExternally\n set(value) = definedExternally\n var headers: dynamic\n get() =
definedExternally\n set(value) = definedExternally\n var body: dynamic\n get() = definedExternally\n
set(value) = definedExternally\n var referrer: String?\n get() = definedExternally\n set(value) =
definedExternally\n var referrerPolicy: dynamic\n get() = definedExternally\n set(value) =
definedExternally\n var mode: RequestMode?\n get() = definedExternally\n set(value) =
definedExternally\n var credentials: RequestCredentials?\n get() = definedExternally\n set(value) =
definedExternally\n var cache: RequestCache?\n get() = definedExternally\n set(value) =
definedExternally\n var redirect: RequestRedirect?\n get() = definedExternally\n set(value) =
definedExternally\n var integrity: String?\n get() = definedExternally\n set(value) = definedExternally\n
var keepalive: Boolean?\n get() = definedExternally\n set(value) = definedExternally\n var window:
Any?\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n\n@kotlin.internal.InlineOnly\n\npublic inline fun RequestInit(method: String? =
undefined, headers: dynamic = undefined, body: dynamic = undefined, referrer: String? = undefined, referrerPolicy:
dynamic = undefined, mode: RequestMode? = undefined, credentials: RequestCredentials? = undefined, cache:
RequestCache? = undefined, redirect: RequestRedirect? = undefined, integrity: String? = undefined, keepalive:
Boolean? = undefined, window: Any? = undefined): RequestInit {\n val o = js(\"({})\")\n o[\"method\"] =
method\n o[\"headers\"] = headers\n o[\"body\"] = body\n o[\"referrer\"] = referrer\n o[\"referrerPolicy\"] =
referrerPolicy\n o[\"mode\"] = mode\n o[\"credentials\"] = credentials\n o[\"cache\"] = cache\n o[\"redirect\"]
= redirect\n o[\"integrity\"] = integrity\n o[\"keepalive\"] = keepalive\n o[\"window\"] = window\n return
o\n}\n\n/**\n * Exposes the JavaScript [Response](https://developer.mozilla.org/en/docs/Web/API/Response) to
Kotlin\n *^\n\npublic external open class Response(body: dynamic = definedExternally, init: ResponseInit =
definedExternally) : Body {\n open val type: ResponseType\n open val url: String\n open val redirected:
Boolean\n open val status: Short\n open val ok: Boolean\n open val statusText: String\n open val headers:

```

```

Headers\n  open val body: dynamic\n  open val trailer: Promise<Headers>\n  override val bodyUsed: Boolean\nfun clone(): Response\n  override fun arrayBuffer(): Promise<ArrayBuffer>\n  override fun blob():
Promise<Blob>\n  override fun formData(): Promise<FormData>\n  override fun json(): Promise<Any?>\n
override fun text(): Promise<String>\n\n  companion object {\n    fun error(): Response\n    fun redirect(url:
String, status: Short = definedExternally): Response\n  }\n\npublic external interface ResponseInit {\n  var
status: Short? /* = 200 */\n    get() = definedExternally\n    set(value) = definedExternally\n  var statusText:
String? /* = \"OK\" */\n    get() = definedExternally\n    set(value) = definedExternally\n  var headers:
dynamic\n    get() = definedExternally\n    set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ResponseInit(status: Short? = 200,
statusText: String? = \"OK\", headers: dynamic = undefined): ResponseInit {\n  val o = js(\"({})\")\n  o[\"status\"]
= status\n  o[\"statusText\"] = statusText\n  o[\"headers\"] = headers\n  return o\n}\n\n/* please, don't implement
this interface! */\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic
external interface RequestType {\n  companion object\n}\n\npublic inline val RequestType.Companion.EMPTY:
RequestType get() = \"\".asDynamic().unsafeCast<RequestType>()\n\npublic inline val
RequestType.Companion.AUDIO: RequestType get() =
\"audio\".asDynamic().unsafeCast<RequestType>()\n\npublic inline val RequestType.Companion.FONT:
RequestType get() = \"font\".asDynamic().unsafeCast<RequestType>()\n\npublic inline val
RequestType.Companion.IMAGE: RequestType get() =
\"image\".asDynamic().unsafeCast<RequestType>()\n\npublic inline val RequestType.Companion.SCRIPT:
RequestType get() = \"script\".asDynamic().unsafeCast<RequestType>()\n\npublic inline val
RequestType.Companion.STYLE: RequestType get() =
\"style\".asDynamic().unsafeCast<RequestType>()\n\npublic inline val RequestType.Companion.TRACK:
RequestType get() = \"track\".asDynamic().unsafeCast<RequestType>()\n\npublic inline val
RequestType.Companion.VIDEO: RequestType get() = \"video\".asDynamic().unsafeCast<RequestType>()\n\n/*
please, don't implement this interface!
*/\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external
interface RequestDestination {\n  companion object\n}\n\npublic inline val
RequestDestination.Companion.EMPTY: RequestDestination get() =
\"\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.DOCUMENT: RequestDestination get() =
\"document\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.EMBED: RequestDestination get() =
\"embed\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.FONT: RequestDestination get() =
\"font\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.IMAGE: RequestDestination get() =
\"image\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.MANIFEST: RequestDestination get() =
\"manifest\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.MEDIA: RequestDestination get() =
\"media\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.OBJECT: RequestDestination get() =
\"object\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.REPORT: RequestDestination get() =
\"report\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.SCRIPT: RequestDestination get() =
\"script\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val

```

```

RequestDestination.Companion.SERVICWORKER: RequestDestination get() =
    \"serviceworker\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.SHAREDWORKER: RequestDestination get() =
    \"sharedworker\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.STYLE: RequestDestination get() =
    \"style\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.WORKER: RequestDestination get() =
    \"worker\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.XSLT: RequestDestination get() =
    \"xslt\".asDynamic().unsafeCast<RequestDestination>()\n\n/* please, don't implement this interface!
*\n\n@JsName(\"null\")\n\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic external
interface RequestMode {\n    companion object\n}\n\npublic inline val RequestMode.Companion.NAVIGATE:
RequestMode get() = \"navigate\".asDynamic().unsafeCast<RequestMode>()\n\npublic inline val
RequestMode.Companion.SAME_ORIGIN: RequestMode get() = \"same-
origin\".asDynamic().unsafeCast<RequestMode>()\n\npublic inline val RequestMode.Companion.NO_CORS:
RequestMode get() = \"no-cors\".asDynamic().unsafeCast<RequestMode>()\n\npublic inline val
RequestMode.Companion.CORS: RequestMode get() = \"cors\".asDynamic().unsafeCast<RequestMode>()\n\n/*
please, don't implement this interface!
*\n\n@JsName(\"null\")\n\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic external
interface RequestCredentials {\n    companion object\n}\n\npublic inline val RequestCredentials.Companion.OMIT:
RequestCredentials get() = \"omit\".asDynamic().unsafeCast<RequestCredentials>()\n\npublic inline val
RequestCredentials.Companion.SAME_ORIGIN: RequestCredentials get() = \"same-
origin\".asDynamic().unsafeCast<RequestCredentials>()\n\npublic inline val
RequestCredentials.Companion.INCLUDE: RequestCredentials get() =
    \"include\".asDynamic().unsafeCast<RequestCredentials>()\n\n/* please, don't implement this interface!
*\n\n@JsName(\"null\")\n\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic external
interface RequestCache {\n    companion object\n}\n\npublic inline val RequestCache.Companion.DEFAULT:
RequestCache get() = \"default\".asDynamic().unsafeCast<RequestCache>()\n\npublic inline val
RequestCache.Companion.NO_STORE: RequestCache get() = \"no-
store\".asDynamic().unsafeCast<RequestCache>()\n\npublic inline val RequestCache.Companion.RELOAD:
RequestCache get() = \"reload\".asDynamic().unsafeCast<RequestCache>()\n\npublic inline val
RequestCache.Companion.NO_CACHE: RequestCache get() = \"no-
cache\".asDynamic().unsafeCast<RequestCache>()\n\npublic inline val
RequestCache.Companion.FORCE_CACHE: RequestCache get() = \"force-
cache\".asDynamic().unsafeCast<RequestCache>()\n\npublic inline val
RequestCache.Companion.ONLY_IF_CACHED: RequestCache get() = \"only-if-
cached\".asDynamic().unsafeCast<RequestCache>()\n\n/* please, don't implement this interface!
*\n\n@JsName(\"null\")\n\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic external
interface RequestRedirect {\n    companion object\n}\n\npublic inline val RequestRedirect.Companion.FOLLOW:
RequestRedirect get() = \"follow\".asDynamic().unsafeCast<RequestRedirect>()\n\npublic inline val
RequestRedirect.Companion.ERROR: RequestRedirect get() =
    \"error\".asDynamic().unsafeCast<RequestRedirect>()\n\npublic inline val RequestRedirect.Companion.MANUAL:
RequestRedirect get() = \"manual\".asDynamic().unsafeCast<RequestRedirect>()\n\n/* please, don't implement this
interface! *\n\n@JsName(\"null\")\n\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\npublic
external interface ResponseType {\n    companion object\n}\n\npublic inline val ResponseType.Companion.BASIC:
ResponseType get() = \"basic\".asDynamic().unsafeCast<ResponseType>()\n\npublic inline val
ResponseType.Companion.CORS: ResponseType get() =
    \"cors\".asDynamic().unsafeCast<ResponseType>()\n\npublic inline val ResponseType.Companion.DEFAULT:

```

```

ResponseType get() = \"default\".asDynamic().unsafeCast<ResponseType>()\n\npublic inline val
ResponseType.Companion.ERROR: ResponseType get() =
\"error\".asDynamic().unsafeCast<ResponseType>()\n\npublic inline val ResponseType.Companion.OPAQUE:
ResponseType get() = \"opaque\".asDynamic().unsafeCast<ResponseType>()\n\npublic inline val
ResponseType.Companion.OPAQUEREDIRECT: ResponseType get() =
\"opaqueredirect\".asDynamic().unsafeCast<ResponseType>()", /*\n * Copyright 2010-2021 JetBrains s.r.o. and
Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file.\n */\n\n// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT
EDIT!\n\n See github.com/kotlin/dukat for details\n\npackage org.w3c.dom.mediacapture\n\nimport
kotlin.js.*\nimport org.khronos.webgl.*\nimport org.w3c.dom.*\nimport org.w3c.dom.events.*\n\n/**\n * Exposes
the JavaScript [MediaStream](https://developer.mozilla.org/en/docs/Web/API/MediaStream) to Kotlin\n */\n\npublic
external open class MediaStream() : EventTarget, MediaProvider {\n    constructor(stream: MediaStream)\n
constructor(tracks: Array<MediaStreamTrack>)\n    open val id: String\n    open val active: Boolean\n    var
onaddtrack: ((MediaStreamTrackEvent) -> dynamic)?\n    var onremovetrack: ((MediaStreamTrackEvent) ->
dynamic)?\n    fun getAudioTracks(): Array<MediaStreamTrack>\n    fun getVideoTracks():
Array<MediaStreamTrack>\n    fun getTracks(): Array<MediaStreamTrack>\n    fun getTrackById(trackId: String):
MediaStreamTrack?\n    fun addTrack(track: MediaStreamTrack)\n    fun removeTrack(track: MediaStreamTrack)\n
    fun clone(): MediaStream\n}\n\n/**\n * Exposes the JavaScript
[MediaStreamTrack](https://developer.mozilla.org/en/docs/Web/API/MediaStreamTrack) to Kotlin\n */\n\npublic
external abstract class MediaStreamTrack : EventTarget {\n    open val kind: String\n    open val id: String\n    open
val label: String\n    open var enabled: Boolean\n    open val muted: Boolean\n    open var onmute: ((Event) ->
dynamic)?\n    open var onunmute: ((Event) -> dynamic)?\n    open val readyState: MediaStreamTrackState\n
open var onended: ((Event) -> dynamic)?\n    open var onoverconstrained: ((Event) -> dynamic)?\n    fun clone():
MediaStreamTrack\n    fun stop()\n    fun getCapabilities(): MediaTrackCapabilities\n    fun getConstraints():
MediaTrackConstraints\n    fun getSettings(): MediaTrackSettings\n    fun applyConstraints(constraints:
MediaTrackConstraints = definedExternally): Promise<Unit>\n}\n\n/**\n * Exposes the JavaScript
[MediaTrackSupportedConstraints](https://developer.mozilla.org/en/docs/Web/API/MediaTrackSupportedConstrain
ts) to Kotlin\n */\n\npublic external interface MediaTrackSupportedConstraints {\n    var width: Boolean? /* = true
*/\n    get() = definedExternally\n    set(value) = definedExternally\n    var height: Boolean? /* = true */\n
get() = definedExternally\n    set(value) = definedExternally\n    var aspectRatio: Boolean? /* = true */\n
get() =
definedExternally\n    set(value) = definedExternally\n    var frameRate: Boolean? /* = true */\n    get() =
definedExternally\n    set(value) = definedExternally\n    var facingMode: Boolean? /* = true */\n    get() =
definedExternally\n    set(value) = definedExternally\n    var resizeMode: Boolean? /* = true */\n    get() =
definedExternally\n    set(value) = definedExternally\n    var volume: Boolean? /* = true */\n    get() =
definedExternally\n    set(value) = definedExternally\n    var sampleRate: Boolean? /* = true */\n    get() =
definedExternally\n    set(value) = definedExternally\n    var sampleSize: Boolean? /* = true */\n    get() =
definedExternally\n    set(value) = definedExternally\n    var echoCancellation: Boolean? /* = true */\n    get()
= definedExternally\n    set(value) = definedExternally\n    var autoGainControl: Boolean? /* = true */\n    get()
= definedExternally\n    set(value) = definedExternally\n    var noiseSuppression: Boolean? /* = true */\n
get() = definedExternally\n    set(value) = definedExternally\n    var latency: Boolean? /* = true */\n    get() =
definedExternally\n    set(value) = definedExternally\n    var channelCount: Boolean? /* = true */\n    get() =
definedExternally\n    set(value) = definedExternally\n    var deviceId: Boolean? /* = true */\n    get() =
definedExternally\n    set(value) = definedExternally\n    var groupId: Boolean? /* = true */\n    get() =
definedExternally\n    set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n\n@kotlin.internal.InlineOnly\n\npublic inline fun
MediaTrackSupportedConstraints(width: Boolean? = true, height: Boolean? = true, aspectRatio: Boolean? = true,
frameRate: Boolean? = true, facingMode: Boolean? = true, resizeMode: Boolean? = true, volume: Boolean? = true,
sampleRate: Boolean? = true, sampleSize: Boolean? = true, echoCancellation: Boolean? = true, autoGainControl:

```

```

Boolean? = true, noiseSuppression: Boolean? = true, latency: Boolean? = true, channelCount: Boolean? = true,
deviceId: Boolean? = true, groupId: Boolean? = true): MediaTrackSupportedConstraints {
    val o = js("{}")
    o["width"] = width
    o["height"] = height
    o["aspectRatio"] = aspectRatio
    o["frameRate"] =
frameRate
    o["facingMode"] = facingMode
    o["resizeMode"] = resizeMode
    o["volume"] = volume
    o["sampleRate"] = sampleRate
    o["sampleSize"] = sampleSize
    o["echoCancellation"] =
echoCancellation
    o["autoGainControl"] = autoGainControl
    o["noiseSuppression"] = noiseSuppression
    o["latency"] = latency
    o["channelCount"] = channelCount
    o["deviceId"] = deviceId
    o["groupId"] =
groupId
    return o
}

public external interface MediaTrackCapabilities {
    var width: ULongRange?
    get() = definedExternally
    set(value) = definedExternally
    var height: ULongRange?
    get() =
definedExternally
    set(value) = definedExternally
    var aspectRatio: DoubleRange?
    get() =
definedExternally
    set(value) = definedExternally
    var frameRate: DoubleRange?
    get() =
definedExternally
    set(value) = definedExternally
    var facingMode: Array<String>?
    get() =
definedExternally
    set(value) = definedExternally
    var resizeMode: Array<String>?
    get() =
definedExternally
    set(value) = definedExternally
    var volume: DoubleRange?
    get() =
definedExternally
    set(value) = definedExternally
    var sampleRate: ULongRange?
    get() =
definedExternally
    set(value) = definedExternally
    var sampleSize: ULongRange?
    get() =
definedExternally
    set(value) = definedExternally
    var echoCancellation: Array<Boolean>?
    get() =
definedExternally
    set(value) = definedExternally
    var autoGainControl: Array<Boolean>?
    get() =
definedExternally
    set(value) = definedExternally
    var noiseSuppression: Array<Boolean>?
    get() =
definedExternally
    set(value) = definedExternally
    var latency: DoubleRange?
    get() =
definedExternally
    set(value) = definedExternally
    var channelCount: ULongRange?
    get() =
definedExternally
    set(value) = definedExternally
    var deviceId: String?
    get() = definedExternally
    set(value) = definedExternally
    var groupId: String?
    get() = definedExternally
    set(value) =
definedExternally
}

@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline fun MediaTrackCapabilities(width:
ULongRange? = undefined, height: ULongRange? = undefined, aspectRatio: DoubleRange? = undefined,
frameRate: DoubleRange? = undefined, facingMode: Array<String>? = undefined, resizeMode: Array<String>? =
undefined, volume: DoubleRange? = undefined, sampleRate: ULongRange? = undefined, sampleSize:
ULongRange? = undefined, echoCancellation: Array<Boolean>? = undefined, autoGainControl: Array<Boolean>?
= undefined, noiseSuppression: Array<Boolean>? = undefined, latency: DoubleRange? = undefined, channelCount:
ULongRange? = undefined, deviceId: String? = undefined, groupId: String? = undefined): MediaTrackCapabilities
{
    val o = js("{}")
    o["width"] = width
    o["height"] = height
    o["aspectRatio"] = aspectRatio
    o["frameRate"] = frameRate
    o["facingMode"] = facingMode
    o["resizeMode"] = resizeMode
    o["volume"] = volume
    o["sampleRate"] = sampleRate
    o["sampleSize"] = sampleSize
    o["echoCancellation"] = echoCancellation
    o["autoGainControl"] = autoGainControl
    o["noiseSuppression"] = noiseSuppression
    o["latency"] = latency
    o["channelCount"] = channelCount
    o["deviceId"] = deviceId
    o["groupId"] = groupId
    return o
}

/**
 * Exposes the JavaScript
 [MediaTrackConstraints](https://developer.mozilla.org/en/docs/Web/API/MediaTrackConstraints) to Kotlin
 */
public external interface MediaTrackConstraints : MediaTrackConstraintSet {
    var advanced:
Array<MediaTrackConstraintSet>?
    get() = definedExternally
    set(value) =
definedExternally
}

@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline fun MediaTrackConstraints(advanced:
Array<MediaTrackConstraintSet>? = undefined, width: dynamic = undefined, height: dynamic = undefined,
aspectRatio: dynamic = undefined, frameRate: dynamic = undefined, facingMode: dynamic = undefined,
resizeMode: dynamic = undefined, volume: dynamic = undefined, sampleRate: dynamic = undefined, sampleSize:
dynamic = undefined, echoCancellation: dynamic = undefined, autoGainControl: dynamic = undefined,
noiseSuppression: dynamic = undefined, latency: dynamic = undefined, channelCount: dynamic = undefined,
deviceId: dynamic = undefined, groupId: dynamic = undefined): MediaTrackConstraints {
    val o = js("{}")

```

```

o["advanced"] = advanced\n o["width"] = width\n o["height"] = height\n o["aspectRatio"] =
aspectRatio\n o["frameRate"] = frameRate\n o["facingMode"] = facingMode\n o["resizeMode"] =
resizeMode\n o["volume"] = volume\n o["sampleRate"] = sampleRate\n o["sampleSize"] = sampleSize\n
o["echoCancellation"] = echoCancellation\n o["autoGainControl"] = autoGainControl\n
o["noiseSuppression"] = noiseSuppression\n o["latency"] = latency\n o["channelCount"] = channelCount\n
o["deviceId"] = deviceId\n o["groupId"] = groupId\n return o\n}\n\npublic external interface
MediaTrackConstraintSet {\n var width: dynamic\n get() = definedExternally\n set(value) =
definedExternally\n var height: dynamic\n get() = definedExternally\n set(value) = definedExternally\n
var aspectRatio: dynamic\n get() = definedExternally\n set(value) = definedExternally\n var frameRate:
dynamic\n get() = definedExternally\n set(value) = definedExternally\n var facingMode: dynamic\n
get() = definedExternally\n set(value) = definedExternally\n var resizeMode: dynamic\n get() =
definedExternally\n set(value) = definedExternally\n var volume: dynamic\n get() = definedExternally\n
set(value) = definedExternally\n var sampleRate: dynamic\n get() = definedExternally\n set(value) =
definedExternally\n var sampleSize: dynamic\n get() = definedExternally\n set(value) =
definedExternally\n var echoCancellation: dynamic\n get() = definedExternally\n set(value) =
definedExternally\n var autoGainControl: dynamic\n get() = definedExternally\n set(value) =
definedExternally\n var noiseSuppression: dynamic\n get() = definedExternally\n set(value) =
definedExternally\n var latency: dynamic\n get() = definedExternally\n set(value) = definedExternally\n
var channelCount: dynamic\n get() = definedExternally\n set(value) = definedExternally\n var deviceId:
dynamic\n get() = definedExternally\n set(value) = definedExternally\n var groupId: dynamic\n get()
= definedExternally\n set(value) = definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun MediaTrackConstraintSet(width:
dynamic = undefined, height: dynamic = undefined, aspectRatio: dynamic = undefined, frameRate: dynamic =
undefined, facingMode: dynamic = undefined, resizeMode: dynamic = undefined, volume: dynamic = undefined,
sampleRate: dynamic = undefined, sampleSize: dynamic = undefined, echoCancellation: dynamic = undefined,
autoGainControl: dynamic = undefined, noiseSuppression: dynamic = undefined, latency: dynamic = undefined,
channelCount: dynamic = undefined, deviceId: dynamic = undefined, groupId: dynamic = undefined):
MediaTrackConstraintSet {\n val o = js("{}")\n o["width"] = width\n o["height"] = height\n
o["aspectRatio"] = aspectRatio\n o["frameRate"] = frameRate\n o["facingMode"] = facingMode\n
o["resizeMode"] = resizeMode\n o["volume"] = volume\n o["sampleRate"] = sampleRate\n
o["sampleSize"] = sampleSize\n o["echoCancellation"] = echoCancellation\n o["autoGainControl"] =
autoGainControl\n o["noiseSuppression"] = noiseSuppression\n o["latency"] = latency\n
o["channelCount"] = channelCount\n o["deviceId"] = deviceId\n o["groupId"] = groupId\n return
o\n}\n\n/**\n * Exposes the JavaScript
[MediaTrackSettings](https://developer.mozilla.org/en/docs/Web/API/MediaTrackSettings) to Kotlin\n *\n\npublic
external interface MediaTrackSettings {\n var width: Int?\n get() = definedExternally\n set(value) =
definedExternally\n var height: Int?\n get() = definedExternally\n set(value) = definedExternally\n var
aspectRatio: Double?\n get() = definedExternally\n set(value) = definedExternally\n var frameRate:
Double?\n get() = definedExternally\n set(value) = definedExternally\n var facingMode: String?\n
get() = definedExternally\n set(value) = definedExternally\n var resizeMode: String?\n get() =
definedExternally\n set(value) = definedExternally\n var volume: Double?\n get() = definedExternally\n
set(value) = definedExternally\n var sampleRate: Int?\n get() = definedExternally\n set(value) =
definedExternally\n var sampleSize: Int?\n get() = definedExternally\n set(value) = definedExternally\n
var echoCancellation: Boolean?\n get() = definedExternally\n set(value) = definedExternally\n var
autoGainControl: Boolean?\n get() = definedExternally\n set(value) = definedExternally\n var
noiseSuppression: Boolean?\n get() = definedExternally\n set(value) = definedExternally\n var latency:
Double?\n get() = definedExternally\n set(value) = definedExternally\n var channelCount: Int?\n
get() = definedExternally\n set(value) = definedExternally\n var deviceId: String?\n get() =

```

```

definedExternally\n    set(value) = definedExternally\n    var groupId: String?\n    get() = definedExternally\n    set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun MediaTrackSettings(width: Int? =
undefined, height: Int? = undefined, aspectRatio: Double? = undefined, frameRate: Double? = undefined,
facingMode: String? = undefined, resizeMode: String? = undefined, volume: Double? = undefined, sampleRate: Int?
= undefined, sampleSize: Int? = undefined, echoCancellation: Boolean? = undefined, autoGainControl: Boolean? =
undefined, noiseSuppression: Boolean? = undefined, latency: Double? = undefined, channelCount: Int? = undefined,
deviceId: String? = undefined, groupId: String? = undefined): MediaTrackSettings {\n    val o = js(\"({})\")\n    o[\"width\"] = width\n    o[\"height\"] = height\n    o[\"aspectRatio\"] = aspectRatio\n    o[\"frameRate\"] =
frameRate\n    o[\"facingMode\"] = facingMode\n    o[\"resizeMode\"] = resizeMode\n    o[\"volume\"] = volume\n    o[\"sampleRate\"] = sampleRate\n    o[\"sampleSize\"] = sampleSize\n    o[\"echoCancellation\"] =
echoCancellation\n    o[\"autoGainControl\"] = autoGainControl\n    o[\"noiseSuppression\"] = noiseSuppression\n    o[\"latency\"] = latency\n    o[\"channelCount\"] = channelCount\n    o[\"deviceId\"] = deviceId\n    o[\"groupId\"] =
groupId\n    return o\n}\n\n/**\n * Exposes the JavaScript
[MediaStreamTrackEvent](https://developer.mozilla.org/en/docs/Web/API/MediaStreamTrackEvent) to Kotlin\n
*/\npublic external open class MediaStreamTrackEvent(type: String, eventInitDict: MediaStreamTrackEventInit) :
Event {\n    open val track: MediaStreamTrack\n\n    companion object {\n        val NONE: Short\n        val
CAPTURING_PHASE: Short\n        val AT_TARGET: Short\n        val BUBBLING_PHASE: Short\n    }\n\n    public external interface MediaStreamTrackEventInit : EventInit {\n        var track:
MediaStreamTrack?\n    }\n\n    @Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n    @kotlin.internal.InlineOnly\n    public inline fun MediaStreamTrackEventInit(track:
MediaStreamTrack?, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false):
MediaStreamTrackEventInit {\n        val o = js(\"({})\")\n        o[\"track\"] = track\n        o[\"bubbles\"] = bubbles\n
        o[\"cancelable\"] = cancelable\n        o[\"composed\"] = composed\n        return o\n    }\n\n    public external open class
OverconstrainedErrorEvent(type: String, eventInitDict: OverconstrainedErrorEventInit) : Event {\n        open val error:
dynamic\n\n        companion object {\n            val NONE: Short\n            val CAPTURING_PHASE: Short\n            val
AT_TARGET: Short\n            val BUBBLING_PHASE: Short\n        }\n\n        public external interface
OverconstrainedErrorEventInit : EventInit {\n            var error: dynamic /* = null */\n            get() = definedExternally\n
            set(value) = definedExternally\n    }\n\n    @Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n    @kotlin.internal.InlineOnly\n    public inline fun OverconstrainedErrorEventInit(error:
dynamic = null, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false):
OverconstrainedErrorEventInit {\n        val o = js(\"({})\")\n        o[\"error\"] = error\n        o[\"bubbles\"] = bubbles\n
        o[\"cancelable\"] = cancelable\n        o[\"composed\"] = composed\n        return o\n    }\n\n    /**\n     * Exposes the JavaScript
[MediaDevices](https://developer.mozilla.org/en/docs/Web/API/MediaDevices) to Kotlin\n     */\n    public external
abstract class MediaDevices : EventTarget {\n        open var ondevicechange: ((Event) -> dynamic)?\n        fun
enumerateDevices(): Promise<Array<MediaDeviceInfo>>\n        fun getSupportedConstraints():
MediaTrackSupportedConstraints\n        fun getUserMedia(constraints: MediaStreamConstraints = definedExternally):
Promise<MediaStream>\n    }\n\n    /**\n     * Exposes the JavaScript
[MediaDeviceInfo](https://developer.mozilla.org/en/docs/Web/API/MediaDeviceInfo) to Kotlin\n     */\n    public
external abstract class MediaDeviceInfo {\n        open val deviceId: String\n        open val kind: MediaDeviceKind\n
        open val label: String\n        open val groupId: String\n        fun toJSON(): dynamic\n    }\n\n    public external abstract class
InputDeviceInfo : MediaDeviceInfo {\n        fun getCapabilities(): MediaTrackCapabilities\n    }\n\n    /**\n     * Exposes the
JavaScript [MediaStreamConstraints](https://developer.mozilla.org/en/docs/Web/API/MediaStreamConstraints) to
Kotlin\n     */\n    public external interface MediaStreamConstraints {\n        var video: dynamic /* = false */\n        get() =
definedExternally\n        set(value) = definedExternally\n        var audio: dynamic /* = false */\n        get() =
definedExternally\n        set(value) = definedExternally\n    }\n\n    @Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n    @kotlin.internal.InlineOnly\n    public inline fun MediaStreamConstraints(video:
dynamic = false, audio: dynamic = false): MediaStreamConstraints {\n        val o = js(\"({})\")\n        o[\"video\"] =

```

```

video\n  o["audio"] = audio\n  return o\n}\n\npublic external interface ConstrainingPattern {\n  var
onoverconstrained: ((Event) -> dynamic)?\n  get() = definedExternally\n  set(value) = definedExternally\n
fun getCapabilities(): Capabilities\n  fun getConstraints(): Constraints\n  fun getSettings(): Settings\n  fun
applyConstraints(constraints: Constraints = definedExternally): Promise<Unit>\n}\n\n/**\n * Exposes the
JavaScript [DoubleRange](https://developer.mozilla.org/en/docs/Web/API/DoubleRange) to Kotlin\n */\n\npublic
external interface DoubleRange {\n  var max: Double?\n  get() = definedExternally\n  set(value) =
definedExternally\n  var min: Double?\n  get() = definedExternally\n  set(value) =
definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun DoubleRange(max: Double? =
undefined, min: Double? = undefined): DoubleRange {\n  val o = js("{}")\n  o["max"] = max\n  o["min"] =
min\n  return o\n}\n\npublic external interface ConstrainDoubleRange : DoubleRange {\n  var exact: Double?\n
get() = definedExternally\n  set(value) = definedExternally\n  var ideal: Double?\n  get() =
definedExternally\n  set(value) = definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun ConstrainDoubleRange(exact: Double?
= undefined, ideal: Double? = undefined, max: Double? = undefined, min: Double? = undefined):
ConstrainDoubleRange {\n  val o = js("{}")\n  o["exact"] = exact\n  o["ideal"] = ideal\n  o["max"] =
max\n  o["min"] = min\n  return o\n}\n\npublic external interface ULongRange {\n  var max: Int?\n  get() =
definedExternally\n  set(value) = definedExternally\n  var min: Int?\n  get() = definedExternally\n
set(value) = definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun ULongRange(max: Int? = undefined,
min: Int? = undefined): ULongRange {\n  val o = js("{}")\n  o["max"] = max\n  o["min"] = min\n  return
o\n}\n\npublic external interface ConstrainULongRange : ULongRange {\n  var exact: Int?\n  get() =
definedExternally\n  set(value) = definedExternally\n  var ideal: Int?\n  get() = definedExternally\n
set(value) = definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun ConstrainULongRange(exact: Int? =
undefined, ideal: Int? = undefined, max: Int? = undefined, min: Int? = undefined): ConstrainULongRange {\n
val o = js("{}")\n  o["exact"] = exact\n  o["ideal"] = ideal\n  o["max"] = max\n  o["min"] = min\n  return
o\n}\n\n/**\n * Exposes the JavaScript
[ConstrainBooleanParameters](https://developer.mozilla.org/en/docs/Web/API/ConstrainBooleanParameters) to
Kotlin\n */\n\npublic external interface ConstrainBooleanParameters {\n  var exact: Boolean?\n  get() =
definedExternally\n  set(value) = definedExternally\n  var ideal: Boolean?\n  get() = definedExternally\n
set(value) = definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun ConstrainBooleanParameters(exact:
Boolean? = undefined, ideal: Boolean? = undefined): ConstrainBooleanParameters {\n  val o = js("{}")\n
o["exact"] = exact\n  o["ideal"] = ideal\n  return o\n}\n\n/**\n * Exposes the JavaScript
[ConstrainDOMStringParameters](https://developer.mozilla.org/en/docs/Web/API/ConstrainDOMStringParameters)
to Kotlin\n */\n\npublic external interface ConstrainDOMStringParameters {\n  var exact: dynamic\n  get() =
definedExternally\n  set(value) = definedExternally\n  var ideal: dynamic\n  get() = definedExternally\n
set(value) = definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun
ConstrainDOMStringParameters(exact: dynamic = undefined, ideal: dynamic = undefined):
ConstrainDOMStringParameters {\n  val o = js("{}")\n  o["exact"] = exact\n  o["ideal"] = ideal\n  return
o\n}\n\npublic external interface Capabilities\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun Capabilities(): Capabilities {\n  val o
= js("{}")\n  return o\n}\n\npublic external interface Settings\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun Settings(): Settings {\n  val o =
js("{}")\n  return o\n}\n\npublic external interface ConstraintSet\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun ConstraintSet(): ConstraintSet {\n

```

```

val o = js("{}")\n return o\n\npublic external interface Constraints : ConstraintSet {\n var advanced:
Array<ConstraintSet>?\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun Constraints(advanced:
Array<ConstraintSet>? = undefined): Constraints {\n val o = js("{}")\n o["advanced"] = advanced\n
return o\n}\n\n/* please, don't implement this interface!
*\n\n@JsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\npublic external
interface MediaStreamTrackState {\n companion object\n}\n\npublic inline val
MediaStreamTrackState.Companion.LIVE: MediaStreamTrackState get() =
"live".asDynamic().unsafeCast<MediaStreamTrackState>()\n\npublic inline val
MediaStreamTrackState.Companion.ENDED: MediaStreamTrackState get() =
"ended".asDynamic().unsafeCast<MediaStreamTrackState>()\n\n/* please, don't implement this interface!
*\n\n@JsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\npublic external
interface VideoFacingModeEnum {\n companion object\n}\n\npublic inline val
VideoFacingModeEnum.Companion.USER: VideoFacingModeEnum get() =
"user".asDynamic().unsafeCast<VideoFacingModeEnum>()\n\npublic inline val
VideoFacingModeEnum.Companion.ENVIRONMENT: VideoFacingModeEnum get() =
"environment".asDynamic().unsafeCast<VideoFacingModeEnum>()\n\npublic inline val
VideoFacingModeEnum.Companion.LEFT: VideoFacingModeEnum get() =
"left".asDynamic().unsafeCast<VideoFacingModeEnum>()\n\npublic inline val
VideoFacingModeEnum.Companion.RIGHT: VideoFacingModeEnum get() =
"right".asDynamic().unsafeCast<VideoFacingModeEnum>()\n\n/* please, don't implement this interface!
*\n\n@JsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\npublic external
interface VideoResizeModeEnum {\n companion object\n}\n\npublic inline val
VideoResizeModeEnum.Companion.NONE: VideoResizeModeEnum get() =
"none".asDynamic().unsafeCast<VideoResizeModeEnum>()\n\npublic inline val
VideoResizeModeEnum.Companion.CROP_AND_SCALE: VideoResizeModeEnum get() = "crop-and-
scale".asDynamic().unsafeCast<VideoResizeModeEnum>()\n\n/* please, don't implement this interface!
*\n\n@JsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\npublic external
interface MediaDeviceKind {\n companion object\n}\n\npublic inline val
MediaDeviceKind.Companion.AUDIOINPUT: MediaDeviceKind get() =
"audioinput".asDynamic().unsafeCast<MediaDeviceKind>()\n\npublic inline val
MediaDeviceKind.Companion.AUDIOOUTPUT: MediaDeviceKind get() =
"audiooutput".asDynamic().unsafeCast<MediaDeviceKind>()\n\npublic inline val
MediaDeviceKind.Companion.VIDEOINPUT: MediaDeviceKind get() =
"videoinput".asDynamic().unsafeCast<MediaDeviceKind>()"}\n\n/*\n * Copyright 2010-2021 JetBrains s.r.o. and
Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file.\n *\n\n// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT
EDIT!\n\n// See github.com/kotlin/dukat for details\n\npackage org.w3c.dom.mediasource\n\nimport
kotlin.js.*\nimport org.khronos.webgl.*\nimport org.w3c.dom.*\nimport org.w3c.dom.events.*\n\n/**\n * Exposes
the JavaScript [MediaSource](https://developer.mozilla.org/en/docs/Web/API/MediaSource) to Kotlin\n *\n\npublic
external open class MediaSource : EventTarget, MediaProvider {\n open val sourceBuffers: SourceBufferList\n
open val activeSourceBuffers: SourceBufferList\n open val readyState: ReadyState\n var duration: Double\n
var onsourceopen: ((Event) -> dynamic)?\n var onsourceended: ((Event) -> dynamic)?\n var onsourceclose:
((Event) -> dynamic)?\n fun addSourceBuffer(type: String): SourceBuffer\n fun
removeSourceBuffer(sourceBuffer: SourceBuffer)\n fun endOfStream(error: EndOfStreamError =
definedExternally)\n fun setLiveSeekableRange(start: Double, end: Double)\n fun clearLiveSeekableRange()\n\n
companion object {\n fun isTypeSupported(type: String): Boolean\n }\n}\n\n/**\n * Exposes the JavaScript

```

[SourceBuffer](https://developer.mozilla.org/en/docs/Web/API/SourceBuffer) to Kotlin\n */\npublic external abstract class SourceBuffer : EventTarget {\n open var mode: AppendMode\n open val updating: Boolean\n open val buffered: TimeRanges\n open var timestampOffset: Double\n open val audioTracks: AudioTrackList\n open val videoTracks: VideoTrackList\n open val textTracks: TextTrackList\n open var appendWindowStart: Double\n open var appendWindowEnd: Double\n open var onupdatestart: ((Event) -> dynamic)?\n open var onupdate: ((Event) -> dynamic)?\n open var onupdateend: ((Event) -> dynamic)?\n open var onerror: ((Event) -> dynamic)?\n open var onabort: ((Event) -> dynamic)?\n fun appendBuffer(data: dynamic)\n fun abort()\n fun remove(start: Double, end: Double)\n}\n\n/**\n * Exposes the JavaScript

[SourceBufferList](https://developer.mozilla.org/en/docs/Web/API/SourceBufferList) to Kotlin\n */\npublic external abstract class SourceBufferList : EventTarget {\n open val length: Int\n open var onaddsourcebuffer: ((Event) -> dynamic)?\n open var onremovesourcebuffer: ((Event) -> dynamic)?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun SourceBufferList.get(index: Int): SourceBuffer? = asDynamic()[index]\n\n/* please, don't implement this interface!\n */\n\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external interface ReadyState {\n companion object\n}\n\npublic inline val ReadyState.Companion.CLOSED: ReadyState get() = \"closed\".asDynamic().unsafeCast<ReadyState>()\n\npublic inline val ReadyState.Companion.OPEN: ReadyState get() = \"open\".asDynamic().unsafeCast<ReadyState>()\n\npublic inline val ReadyState.Companion.ENDED: ReadyState get() = \"ended\".asDynamic().unsafeCast<ReadyState>()\n\n/* please, don't implement this interface!\n */\n\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external interface EndOfStreamError {\n companion object\n}\n\npublic inline val EndOfStreamError.Companion.NETWORK: EndOfStreamError get() = \"network\".asDynamic().unsafeCast<EndOfStreamError>()\n\npublic inline val EndOfStreamError.Companion.DECODE: EndOfStreamError get() = \"decode\".asDynamic().unsafeCast<EndOfStreamError>()\n\n/* please, don't implement this interface!\n */\n\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external interface AppendMode {\n companion object\n}\n\npublic inline val AppendMode.Companion.SEGMENTS: AppendMode get() = \"segments\".asDynamic().unsafeCast<AppendMode>()\n\npublic inline val AppendMode.Companion.SEQUENCE: AppendMode get() = \"sequence\".asDynamic().unsafeCast<AppendMode>()\n\n/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n/*\n * NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!\n */\n\n/*\n * See github.com/kotlin/dukat for details\n */\n\npackage org.w3c.dom.pointerevents\n\nimport kotlin.js.*\nimport org.khronos.webgl.*\nimport org.w3c.dom.*\nimport org.w3c.dom.events.*\n\npublic external interface PointerEventInit : MouseEventInit {\n var pointerId: Int? /* = 0 */\n get() = definedExternally\n set(value) = definedExternally\n var width: Double? /* = 1.0 */\n get() = definedExternally\n set(value) = definedExternally\n var height: Double? /* = 1.0 */\n get() = definedExternally\n set(value) = definedExternally\n var pressure: Float? /* = 0f */\n get() = definedExternally\n set(value) = definedExternally\n var tangentialPressure: Float? /* = 0f */\n get() = definedExternally\n set(value) = definedExternally\n var tiltX: Int? /* = 0 */\n get() = definedExternally\n set(value) = definedExternally\n var tiltY: Int? /* = 0 */\n get() = definedExternally\n set(value) = definedExternally\n var twist: Int? /* = 0 */\n get() = definedExternally\n set(value) = definedExternally\n var pointerType: String? /* = \"\" */\n get() = definedExternally\n set(value) = definedExternally\n var isPrimary: Boolean? /* = false */\n get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun PointerEventInit(pointerId: Int? = 0, width: Double? = 1.0, height: Double? = 1.0, pressure: Float? = 0f, tangentialPressure: Float? = 0f, tiltX: Int? = 0, tiltY: Int? = 0, twist: Int? = 0, pointerType: String? = \"\", isPrimary: Boolean? = false, screenX: Int? = 0, screenY: Int? = 0) = PointerEventInit(pointerId, width, height, pressure, tangentialPressure, tiltX, tiltY, twist, pointerType, isPrimary, screenX, screenY)

```

Int? = 0, clientX: Int? = 0, clientY: Int? = 0, button: Short? = 0, buttons: Short? = 0, relatedTarget: EventTarget? =
null, region: String? = null, ctrlKey: Boolean? = false, shiftKey: Boolean? = false, altKey: Boolean? = false,
metaKey: Boolean? = false, modifierAltGraph: Boolean? = false, modifierCapsLock: Boolean? = false, modifierFn:
Boolean? = false, modifierFnLock: Boolean? = false, modifierHyper: Boolean? = false, modifierNumLock:
Boolean? = false, modifierScrollLock: Boolean? = false, modifierSuper: Boolean? = false, modifierSymbol:
Boolean? = false, modifierSymbolLock: Boolean? = false, view: Window? = null, detail: Int? = 0, bubbles:
Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): PointerEventInit {\n val o =
js("{}")\n o["pointerId"] = pointerId\n o["width"] = width\n o["height"] = height\n o["pressure"] =
pressure\n o["tangentialPressure"] = tangentialPressure\n o["tiltX"] = tiltX\n o["tiltY"] = tiltY\n
o["twist"] = twist\n o["pointerType"] = pointerType\n o["isPrimary"] = isPrimary\n o["screenX"] =
screenX\n o["screenY"] = screenY\n o["clientX"] = clientX\n o["clientY"] = clientY\n o["button"] =
button\n o["buttons"] = buttons\n o["relatedTarget"] = relatedTarget\n o["region"] = region\n
o["ctrlKey"] = ctrlKey\n o["shiftKey"] = shiftKey\n o["altKey"] = altKey\n o["metaKey"] = metaKey\n
o["modifierAltGraph"] = modifierAltGraph\n o["modifierCapsLock"] = modifierCapsLock\n
o["modifierFn"] = modifierFn\n o["modifierFnLock"] = modifierFnLock\n o["modifierHyper"] =
modifierHyper\n o["modifierNumLock"] = modifierNumLock\n o["modifierScrollLock"] =
modifierScrollLock\n o["modifierSuper"] = modifierSuper\n o["modifierSymbol"] = modifierSymbol\n
o["modifierSymbolLock"] = modifierSymbolLock\n o["view"] = view\n o["detail"] = detail\n
o["bubbles"] = bubbles\n o["cancelable"] = cancelable\n o["composed"] = composed\n return
o}\n}\n\n/**\n * Exposes the JavaScript

```

```

[PointerEvent](https://developer.mozilla.org/en/docs/Web/API/PointerEvent) to Kotlin\n */\npublic external open
class PointerEvent(type: String, eventInitDict: PointerEventInit = definedExternally) : MouseEvent {\n open val
pointerId: Int\n open val width: Double\n open val height: Double\n open val pressure: Float\n open val
tangentialPressure: Float\n open val tiltX: Int\n open val tiltY: Int\n open val twist: Int\n open val
pointerType: String\n open val isPrimary: Boolean\n\n companion object {\n val NONE: Short\n val
CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val BUBBLING_PHASE: Short\n }\n}"/*\n
* Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code
is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n// NOTE: THIS
FILE IS AUTO-GENERATED, DO NOT EDIT!\n\n See github.com/kotlin/dukat for details\n\npackage
org.w3c.dom.svg\n\nimport kotlin.js.*\nimport org.khronos.webgl.*\nimport org.w3c.dom.*\nimport
org.w3c.dom.css.*\n\n/**\n * Exposes the JavaScript

```

```

[SVGElement](https://developer.mozilla.org/en/docs/Web/API/SVGElement) to Kotlin\n */\npublic external
abstract class SVGElement : Element, ElementCSSInlineStyle, GlobalEventHandlers, SVGElementInstance {\n
open val dataset: DOMStringMap\n open val ownerSVGElement: SVGSVGElement?\n open val
viewportElement: SVGElement?\n open var tabIndex: Int\n fun focus()\n fun blur()\n\n companion object
{\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n
val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val
ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE:
Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\npublic external interface
SVGBoundingBoxOptions {\n var fill: Boolean? /* = true */\n get() = definedExternally\n set(value) =
definedExternally\n var stroke: Boolean? /* = false */\n get() = definedExternally\n set(value) =
definedExternally\n var markers: Boolean? /* = false */\n get() = definedExternally\n set(value) =
definedExternally\n var clipped: Boolean? /* = false */\n get() = definedExternally\n set(value) =

```

```

definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun SVGBoundingBoxOptions(fill:
Boolean? = true, stroke: Boolean? = false, markers: Boolean? = false, clipped: Boolean? = false):
SVGBoundingBoxOptions {\n val o = js(\"({})\")\n o[\"fill\"] = fill\n o[\"stroke\"] = stroke\n o[\"markers\"]
= markers\n o[\"clipped\"] = clipped\n return o\n}\n\n/**\n * Exposes the JavaScript
[SVGGraphicsElement](https://developer.mozilla.org/en/docs/Web/API/SVGGraphicsElement) to Kotlin\n
*\npublic external abstract class SVGGraphicsElement : SVGElement, SVGTests {\n open val transform:
SVGAnimatedTransformList\n fun getBBox(options: SVGBoundingBoxOptions = definedExternally):
DOMRect\n fun getCTM(): DOMMatrix?\n fun getScreenCTM(): DOMMatrix?\n\n companion object {\n
val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[SVGGeometryElement](https://developer.mozilla.org/en/docs/Web/API/SVGGeometryElement) to Kotlin\n
*\npublic external abstract class SVGGeometryElement : SVGGraphicsElement {\n open val pathLength:
SVGAnimatedNumber\n fun isPointInFill(point: DOMPoint): Boolean\n fun isPointInStroke(point: DOMPoint):
Boolean\n fun getTotalLength(): Float\n fun getPointAtLength(distance: Float): DOMPoint\n\n companion
object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE:
Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val
ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE:
Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[SVGNumber](https://developer.mozilla.org/en/docs/Web/API/SVGNumber) to Kotlin\n\n*\npublic external
abstract class SVGNumber {\n open var value: Float\n}\n\n/**\n * Exposes the JavaScript
[SVGLength](https://developer.mozilla.org/en/docs/Web/API/SVGLength) to Kotlin\n\n*\npublic external
abstract class SVGLength {\n open val unitType: Short\n open var value: Float\n open var valueInSpecifiedUnits:
Float\n open var valueAsString: String\n fun newValueSpecifiedUnits(unitType: Short, valueInSpecifiedUnits:
Float)\n fun convertToSpecifiedUnits(unitType: Short)\n\n companion object {\n val
SVG_LENGTHTYPE_UNKNOWN: Short\n val SVG_LENGTHTYPE_NUMBER: Short\n val
SVG_LENGTHTYPE_PERCENTAGE: Short\n val SVG_LENGTHTYPE_EMS: Short\n val
SVG_LENGTHTYPE_EXS: Short\n val SVG_LENGTHTYPE_PX: Short\n val
SVG_LENGTHTYPE_CM: Short\n val SVG_LENGTHTYPE_MM: Short\n val
SVG_LENGTHTYPE_IN: Short\n val SVG_LENGTHTYPE_PT: Short\n val SVG_LENGTHTYPE_PC:
Short\n }\n}\n\n/**\n * Exposes the JavaScript
[SVGAngle](https://developer.mozilla.org/en/docs/Web/API/SVGAngle) to Kotlin\n\n*\npublic external abstract
class SVGAngle {\n open val unitType: Short\n open var value: Float\n open var valueInSpecifiedUnits:
Float\n open var valueAsString: String\n fun newValueSpecifiedUnits(unitType: Short, valueInSpecifiedUnits:
Float)\n fun convertToSpecifiedUnits(unitType: Short)\n\n companion object {\n val
SVG_ANGLETYPE_UNKNOWN: Short\n val SVG_ANGLETYPE_UNSPECIFIED: Short\n val

```

```

SVG_ANGLETYPE_DEG: Short\n    val SVG_ANGLETYPE_RAD: Short\n    val
SVG_ANGLETYPE_GRAD: Short\n    }\n}\n\npublic external abstract class SVGNameList {\n    open val length:
Int\n    open val numberOfItems: Int\n    fun clear()\n    fun initialize(newItem: dynamic): dynamic\n    fun
insertItemBefore(newItem: dynamic, index: Int): dynamic\n    fun replaceItem(newItem: dynamic, index: Int):
dynamic\n    fun removeItem(index: Int): dynamic\n    fun appendItem(newItem: dynamic): dynamic\n    fun
getItem(index: Int): dynamic\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\n\npublic inline operator fun SVGNameList.get(index: Int):
dynamic = asDynamic()[index]\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\n\npublic inline operator fun SVGNameList.set(index: Int,
newItem: dynamic) { asDynamic()[index] = newItem }\n\n/**\n * Exposes the JavaScript
[SVGNumberList](https://developer.mozilla.org/en/docs/Web/API/SVGNumberList) to Kotlin\n *\n\npublic external
abstract class SVGNumberList {\n    open val length: Int\n    open val numberOfItems: Int\n    fun clear()\n    fun
initialize(newItem: SVGNumber): SVGNumber\n    fun insertItemBefore(newItem: SVGNumber, index: Int):
SVGNumber\n    fun replaceItem(newItem: SVGNumber, index: Int): SVGNumber\n    fun removeItem(index: Int):
SVGNumber\n    fun appendItem(newItem: SVGNumber): SVGNumber\n    fun getItem(index: Int):
SVGNumber\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\n\npublic inline operator fun SVGNumberList.get(index:
Int): SVGNumber? = asDynamic()[index]\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\n\npublic inline operator fun SVGNumberList.set(index:
Int, newItem: SVGNumber) { asDynamic()[index] = newItem }\n\n/**\n * Exposes the JavaScript
[SVGLengthList](https://developer.mozilla.org/en/docs/Web/API/SVGLengthList) to Kotlin\n *\n\npublic external
abstract class SVGLengthList {\n    open val length: Int\n    open val numberOfItems: Int\n    fun clear()\n    fun
initialize(newItem: SVGLength): SVGLength\n    fun insertItemBefore(newItem: SVGLength, index: Int):
SVGLength\n    fun replaceItem(newItem: SVGLength, index: Int): SVGLength\n    fun removeItem(index: Int):
SVGLength\n    fun appendItem(newItem: SVGLength): SVGLength\n    fun getItem(index: Int):
SVGLength\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\n\npublic inline operator fun SVGLengthList.get(index:
Int): SVGLength? = asDynamic()[index]\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\n\npublic inline operator fun SVGLengthList.set(index: Int,
newItem: SVGLength) { asDynamic()[index] = newItem }\n\n/**\n * Exposes the JavaScript
[SVGAnimatedBoolean](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedBoolean) to Kotlin\n *\n\npublic external abstract class SVGAnimatedBoolean {\n    open var baseVal: Boolean\n    open val animVal:
Boolean\n}\n\n/**\n * Exposes the JavaScript
[SVGAnimatedEnumeration](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedEnumeration) to
Kotlin\n *\n\npublic external abstract class SVGAnimatedEnumeration {\n    open var baseVal: Short\n    open val
animVal: Short\n}\n\n/**\n * Exposes the JavaScript
[SVGAnimatedInteger](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedInteger) to Kotlin\n *\n\npublic external abstract class SVGAnimatedInteger {\n    open var baseVal: Int\n    open val animVal:
Int\n}\n\n/**\n * Exposes the JavaScript
[SVGAnimatedNumber](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedNumber) to Kotlin\n *\n\npublic external abstract class SVGAnimatedNumber {\n    open var baseVal: Float\n    open val animVal:
Float\n}\n\n/**\n * Exposes the JavaScript
[SVGAnimatedLength](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedLength) to Kotlin\n *\n\npublic external abstract class SVGAnimatedLength {\n    open val baseVal: SVGLength\n    open val animVal:
SVGLength\n}\n\n/**\n * Exposes the JavaScript
[SVGAnimatedAngle](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedAngle) to Kotlin\n *\n\npublic
external abstract class SVGAnimatedAngle {\n    open val baseVal: SVGAngle\n    open val animVal:
SVGAngle\n}\n\n/**\n * Exposes the JavaScript

```

```

[SVGAnimatedString](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedString) to Kotlin\n *^npublic
external abstract class SVGAnimatedString {\n  open var baseVal: String\n  open val animVal: String\n}\n\n/**\n * Exposes the JavaScript [SVGAnimatedRect](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedRect)
to Kotlin\n *^npublic external abstract class SVGAnimatedRect {\n  open val baseVal: DOMRect\n  open val
animVal: DOMRectReadOnly\n}\n\n/**\n * Exposes the JavaScript
[SVGAnimatedNumberList](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedNumberList) to Kotlin\n
*^npublic external abstract class SVGAnimatedNumberList {\n  open val baseVal: SVGNumberList\n  open val
animVal: SVGNumberList\n}\n\n/**\n * Exposes the JavaScript
[SVGAnimatedLengthList](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedLengthList) to Kotlin\n
*^npublic external abstract class SVGAnimatedLengthList {\n  open val baseVal: SVGLengthList\n  open val
animVal: SVGLengthList\n}\n\n/**\n * Exposes the JavaScript
[SVGStringList](https://developer.mozilla.org/en/docs/Web/API/SVGStringList) to Kotlin\n *^npublic external
abstract class SVGStringList {\n  open val length: Int\n  open val numberOfItems: Int\n  fun clear()\n  fun
initialize(newItem: String): String\n  fun insertItemBefore(newItem: String, index: Int): String\n  fun
replaceItem(newItem: String, index: Int): String\n  fun removeItem(index: Int): String\n  fun
appendItem(newItem: String): String\n  fun getItem(index: Int):
String\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun SVGStringList.get(index:
Int): String? = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun SVGStringList.set(index: Int,
newItem: String) { asDynamic()[index] = newItem }\n\n/**\n * Exposes the JavaScript
[SVGUnitTypes](https://developer.mozilla.org/en/docs/Web/API/SVGUnitTypes) to Kotlin\n
*^n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external interface SVGUnitTypes
{\n  companion object {\n    val SVG_UNIT_TYPE_UNKNOWN: Short\n    val
SVG_UNIT_TYPE_USERSPACEONUSE: Short\n    val SVG_UNIT_TYPE_OBJECTBOUNDINGBOX:
Short\n  }\n}\n\n/**\n * Exposes the JavaScript
[SVGTTests](https://developer.mozilla.org/en/docs/Web/API/SVGTTests) to Kotlin\n *^npublic external interface
SVGTTests {\n  val requiredExtensions: SVGStringList\n  val systemLanguage: SVGStringList\n}\n\npublic
external interface SVGFitToViewBox {\n  val viewBox: SVGAnimatedRect\n  val preserveAspectRatio:
SVGAnimatedPreserveAspectRatio\n}\n\n/**\n * Exposes the JavaScript
[SVGZoomAndPan](https://developer.mozilla.org/en/docs/Web/API/SVGZoomAndPan) to Kotlin\n
*^n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external interface
SVGZoomAndPan {\n  var zoomAndPan: Short\n\n  companion object {\n    val
SVG_ZOOMANDPAN_UNKNOWN: Short\n    val SVG_ZOOMANDPAN_DISABLE: Short\n    val
SVG_ZOOMANDPAN_MAGNIFY: Short\n  }\n}\n\n/**\n * Exposes the JavaScript
[SVGURIReference](https://developer.mozilla.org/en/docs/Web/API/SVGURIReference) to Kotlin\n *^npublic
external interface SVGURIReference {\n  val href: SVGAnimatedString\n}\n\n/**\n * Exposes the JavaScript
[SVGSVGElement](https://developer.mozilla.org/en/docs/Web/API/SVGSVGElement) to Kotlin\n *^npublic
external abstract class SVGSVGElement : SVGGraphicsElement, SVGFitToViewBox, SVGZoomAndPan,
WindowEventHandlers {\n  open val x: SVGAnimatedLength\n  open val y: SVGAnimatedLength\n  open val
width: SVGAnimatedLength\n  open val height: SVGAnimatedLength\n  open var currentScale: Float\n  open
val currentTranslate: DOMPointReadOnly\n  fun getIntersectionList(rect: DOMRectReadOnly, referenceElement:
SVGElement?): NodeList\n  fun getEnclosureList(rect: DOMRectReadOnly, referenceElement: SVGElement?):
NodeList\n  fun checkIntersection(element: SVGElement, rect: DOMRectReadOnly): Boolean\n  fun
checkEnclosure(element: SVGElement, rect: DOMRectReadOnly): Boolean\n  fun deselectAll()\n  fun
createSVGNumber(): SVGNumber\n  fun createSVGLength(): SVGLength\n  fun createSVGAngle():
SVGAngle\n  fun createSVGPoint(): DOMPoint\n  fun createSVGMatrix(): DOMMatrix\n  fun
createSVGRect(): DOMRect\n  fun createSVGTransform(): SVGTransform\n  fun

```



```

Short\n    val ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n    val CDATA_SECTION_NODE:
Short\n    val ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE: Short\n    val
PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val
DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[SVGMetadataElement](https://developer.mozilla.org/en/docs/Web/API/SVGMetadataElement) to Kotlin\n
*\npublic external abstract class SVGMetadataElement : SVGElement {\n    companion object {\n    val
ELEMENT_NODE: Short\n    val ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n    val
CDATA_SECTION_NODE: Short\n    val ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE:
Short\n    val PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val
DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[SVGTitleElement](https://developer.mozilla.org/en/docs/Web/API/SVGTitleElement) to Kotlin\n
*\npublic
external abstract class SVGTitleElement : SVGElement {\n    companion object {\n    val ELEMENT_NODE:
Short\n    val ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n    val CDATA_SECTION_NODE:
Short\n    val ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE: Short\n    val
PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val
DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[SVGSymbolElement](https://developer.mozilla.org/en/docs/Web/API/SVGSymbolElement) to Kotlin\n
*\npublic
external abstract class SVGSymbolElement : SVGGraphicsElement, SVGFitToViewBox {\n    companion object
{\n    val ELEMENT_NODE: Short\n    val ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n
val CDATA_SECTION_NODE: Short\n    val ENTITY_REFERENCE_NODE: Short\n    val
ENTITY_NODE: Short\n    val PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE:
Short\n    val DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[SVGUseElement](https://developer.mozilla.org/en/docs/Web/API/SVGUseElement) to Kotlin\n
*\npublic external
abstract class SVGUseElement : SVGGraphicsElement, SVGURIReference {\n    open val x:
SVGAnimatedLength\n    open val y: SVGAnimatedLength\n    open val width: SVGAnimatedLength\n    open val
height: SVGAnimatedLength\n    open val instanceRoot: SVGElement?\n    open val animatedInstanceRoot:
SVGElement?\n\n    companion object {\n    val ELEMENT_NODE: Short\n    val ATTRIBUTE_NODE:
Short\n    val TEXT_NODE: Short\n    val CDATA_SECTION_NODE: Short\n    val

```

```

ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE: Short\n    val
PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val
DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\npublic external open class
SVGUseElementShadowRoot : ShadowRoot {\n    companion object {\n        val ELEMENT_NODE: Short\n
    val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n
    val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val
PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\npublic external interface
SVGElementInstance {\n    val correspondingElement: SVGElement?\n        get() = definedExternally\n    val
    correspondingUseElement: SVGUseElement?\n        get() = definedExternally\n}\n\npublic external open class
ShadowAnimation(source: dynamic, newTarget: dynamic) {\n    open val sourceAnimation: dynamic\n}\n\n/**\n *
Exposes the JavaScript [SVGSwitchElement](https://developer.mozilla.org/en/docs/Web/API/SVGSwitchElement)
to Kotlin\n */\n\npublic external abstract class SVGSwitchElement : SVGGraphicsElement {\n    companion object
{\n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n
    val CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val
ENTITY_NODE: Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE:
Short\n        val DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\npublic external interface
GetSVGDocument {\n    fun getSVGDocument(): Document\n}\n\n/**\n * Exposes the JavaScript
[SVGStyleElement](https://developer.mozilla.org/en/docs/Web/API/SVGStyleElement) to Kotlin\n */\n\npublic
external abstract class SVGStyleElement : SVGElement, LinkStyle {\n    open var type: String\n    open var media:
String\n    open var title: String\n\n    companion object {\n        val ELEMENT_NODE: Short\n        val
ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val
ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val
PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[SVGTransform](https://developer.mozilla.org/en/docs/Web/API/SVGTransform) to Kotlin\n */\n\npublic external
abstract class SVGTransform {\n    open val type: Short\n    open val matrix: DOMMatrix\n    open val angle:
Float\n    fun setMatrix(matrix: DOMMatrixReadOnly)\n    fun setTranslate(tx: Float, ty: Float)\n    fun setScale(sx:

```

```

Float, sy: Float)\n fun setRotate(angle: Float, cx: Float, cy: Float)\n fun setSkewX(angle: Float)\n fun
setSkewY(angle: Float)\n\n companion object {\n val SVG_TRANSFORM_UNKNOWN: Short\n val
SVG_TRANSFORM_MATRIX: Short\n val SVG_TRANSFORM_TRANSLATE: Short\n val
SVG_TRANSFORM_SCALE: Short\n val SVG_TRANSFORM_ROTATE: Short\n val
SVG_TRANSFORM_SKEWX: Short\n val SVG_TRANSFORM_SKEWY: Short\n }\n\n/**\n * Exposes
the JavaScript [SVGTransformList](https://developer.mozilla.org/en/docs/Web/API/SVGTransformList) to Kotlin\n
*\npublic external abstract class SVGTransformList {\n open val length: Int\n open val numberOfItems: Int\n
fun clear()\n fun initialize(newItem: SVGTransform): SVGTransform\n fun insertItemBefore(newItem:
SVGTransform, index: Int): SVGTransform\n fun replaceItem(newItem: SVGTransform, index: Int):
SVGTransform\n fun removeItem(index: Int): SVGTransform\n fun appendItem(newItem: SVGTransform):
SVGTransform\n fun createSVGTransformFromMatrix(matrix: DOMMatrixReadOnly): SVGTransform\n fun
consolidate(): SVGTransform?\n fun getItem(index: Int):
SVGTransform\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun SVGTransformList.get(index:
Int): SVGTransform? = asDynamic()[index]\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline operator fun SVGTransformList.set(index:
Int, newItem: SVGTransform) { asDynamic()[index] = newItem }\n\n/**\n * Exposes the JavaScript
[SVGAnimatedTransformList](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedTransformList) to
Kotlin\n *\npublic external abstract class SVGAnimatedTransformList {\n open val baseVal:
SVGTransformList\n open val animVal: SVGTransformList\n}\n\n/**\n * Exposes the JavaScript
[SVGPreserveAspectRatio](https://developer.mozilla.org/en/docs/Web/API/SVGPreserveAspectRatio) to Kotlin\n
*\npublic external abstract class SVGPreserveAspectRatio {\n open var align: Short\n open var meetOrSlice:
Short\n\n companion object {\n val SVG_PRESERVEASPECTRATIO_UNKNOWN: Short\n val
SVG_PRESERVEASPECTRATIO_NONE: Short\n val SVG_PRESERVEASPECTRATIO_XMINYMIN:
Short\n val SVG_PRESERVEASPECTRATIO_XMIDYMIN: Short\n val
SVG_PRESERVEASPECTRATIO_XMAXYMIN: Short\n val
SVG_PRESERVEASPECTRATIO_XMINYMID: Short\n val
SVG_PRESERVEASPECTRATIO_XMIDYMID: Short\n val
SVG_PRESERVEASPECTRATIO_XMAXYMID: Short\n val
SVG_PRESERVEASPECTRATIO_XMINYMAX: Short\n val
SVG_PRESERVEASPECTRATIO_XMIDYMAX: Short\n val
SVG_PRESERVEASPECTRATIO_XMAXYMAX: Short\n val SVG_MEETORSLICE_UNKNOWN: Short\n
val SVG_MEETORSLICE_MEET: Short\n val SVG_MEETORSLICE_SLICE: Short\n }\n}\n\n/**\n *
Exposes the JavaScript
[SVGAnimatedPreserveAspectRatio](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedPreserveAspect
Ratio) to Kotlin\n *\npublic external abstract class SVGAnimatedPreserveAspectRatio {\n open val baseVal:
SVGPreserveAspectRatio\n open val animVal: SVGPreserveAspectRatio\n}\n\n/**\n * Exposes the JavaScript
[SVGPathElement](https://developer.mozilla.org/en/docs/Web/API/SVGPathElement) to Kotlin\n *\npublic
external abstract class SVGPathElement : SVGGeometryElement {\n companion object {\n val
ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript

```

```

[SVGRectElement](https://developer.mozilla.org/en/docs/Web/API/SVGRectElement) to Kotlin\n * public
external abstract class SVGRectElement : SVGGeometryElement {\n open val x: SVGAnimatedLength\n open
val y: SVGAnimatedLength\n open val width: SVGAnimatedLength\n open val height: SVGAnimatedLength\n
open val rx: SVGAnimatedLength\n open val ry: SVGAnimatedLength\n\n companion object {\n val
ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[SVGCircleElement](https://developer.mozilla.org/en/docs/Web/API/SVGCircleElement) to Kotlin\n * public
external abstract class SVGCircleElement : SVGGeometryElement {\n open val cx: SVGAnimatedLength\n
open val cy: SVGAnimatedLength\n open val r: SVGAnimatedLength\n\n companion object {\n val
ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:
Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[SVGEllipseElement](https://developer.mozilla.org/en/docs/Web/API/SVGEllipseElement) to Kotlin\n * public
external abstract class SVGEllipseElement : SVGGeometryElement {\n open val cx: SVGAnimatedLength\n
open val cy: SVGAnimatedLength\n open val rx: SVGAnimatedLength\n open val ry: SVGAnimatedLength\n\n
companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val
TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE:
Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val
COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n
val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[SVGLineElement](https://developer.mozilla.org/en/docs/Web/API/SVGLineElement) to Kotlin\n * public
external abstract class SVGLineElement : SVGGeometryElement {\n open val x1: SVGAnimatedLength\n open
val y1: SVGAnimatedLength\n open val x2: SVGAnimatedLength\n open val y2: SVGAnimatedLength\n\n
companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val
TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE:
Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val
COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n
val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val

```

DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n } \n\n\n * Exposes the JavaScript [SVGMeshElement](https://developer.mozilla.org/en/docs/Web/API/SVGMeshElement) to Kotlin\n *\npublic external abstract class SVGMeshElement : SVGGeometryElement, SVGURIReference {\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val

DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n } \n\n\n * Exposes the JavaScript [SVGAnimatedPoints](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedPoints) to Kotlin\n *\npublic external interface SVGAnimatedPoints {\n val points: SVGPointList\n val animatedPoints: SVGPointList\n}\n\npublic external abstract class SVGPointList {\n open val length: Int\n open val numberOfItems: Int\n fun clear()\n fun initialize(newItem: DOMPoint): DOMPoint\n fun insertItemBefore(newItem: DOMPoint, index: Int): DOMPoint\n fun replaceItem(newItem: DOMPoint, index: Int): DOMPoint\n fun removeItem(index: Int): DOMPoint\n fun appendItem(newItem: DOMPoint): DOMPoint\n fun getItem(index: Int): DOMPoint\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun SVGPointList.get(index: Int): DOMPoint? = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun SVGPointList.set(index: Int, newItem: DOMPoint) { asDynamic()[index] = newItem }\n\n\n * Exposes the JavaScript [SVGPolylineElement](https://developer.mozilla.org/en/docs/Web/API/SVGPolylineElement) to Kotlin\n *\npublic external abstract class SVGPolylineElement : SVGGeometryElement, SVGAnimatedPoints {\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val

DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n } \n\n\n * Exposes the JavaScript [SVGPolygonElement](https://developer.mozilla.org/en/docs/Web/API/SVGPolygonElement) to Kotlin\n *\npublic external abstract class SVGPolygonElement : SVGGeometryElement, SVGAnimatedPoints {\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val

DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n } \n\n\n * Exposes the JavaScript [SVGTextContentElement](https://developer.mozilla.org/en/docs/Web/API/SVGTextContentElement) to Kotlin\n *\npublic external abstract class SVGTextContentElement : SVGGraphicsElement {\n open val textLength: SVGAnimatedLength\n open val lengthAdjust: SVGAnimatedEnumeration\n fun getNumberOfChars(): Int\n

```

fun getComputedTextLength(): Float\n fun getSubStringLength(charnum: Int, nchars: Int): Float\n fun
getStartPositionOfChar(charnum: Int): DOMPoint\n fun getEndPositionOfChar(charnum: Int): DOMPoint\n fun
getExtentOfChar(charnum: Int): DOMRect\n fun getRotationOfChar(charnum: Int): Float\n fun
getCharNumAtPosition(point: DOMPoint): Int\n fun selectSubString(charnum: Int, nchars: Int)\n\n companion
object {\n val LENGTHADJUST_UNKNOWN: Short\n val LENGTHADJUST_SPACING: Short\n
val LENGTHADJUST_SPACINGANDGLYPHS: Short\n val ELEMENT_NODE: Short\n val
ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[SVGTextPositioningElement](https://developer.mozilla.org/en/docs/Web/API/SVGTextPositioningElement) to
Kotlin\n */\npublic external abstract class SVGTextPositioningElement : SVGTextContentElement {\n open val x:
SVGAnimatedLengthList\n open val y: SVGAnimatedLengthList\n open val dx: SVGAnimatedLengthList\n
open val dy: SVGAnimatedLengthList\n open val rotate: SVGAnimatedNumberList\n\n companion object {\n
val LENGTHADJUST_UNKNOWN: Short\n val LENGTHADJUST_SPACING: Short\n val
LENGTHADJUST_SPACINGANDGLYPHS: Short\n val ELEMENT_NODE: Short\n val
ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[SVGTextElement](https://developer.mozilla.org/en/docs/Web/API/SVGTextElement) to Kotlin\n */\npublic
external abstract class SVGTextElement : SVGTextPositioningElement {\n companion object {\n val
LENGTHADJUST_UNKNOWN: Short\n val LENGTHADJUST_SPACING: Short\n val
LENGTHADJUST_SPACINGANDGLYPHS: Short\n val ELEMENT_NODE: Short\n val
ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[SVGTSpanElement](https://developer.mozilla.org/en/docs/Web/API/SVGTSpanElement) to Kotlin\n */\npublic
external abstract class SVGTSpanElement : SVGTextPositioningElement {\n companion object {\n val
LENGTHADJUST_UNKNOWN: Short\n val LENGTHADJUST_SPACING: Short\n val
LENGTHADJUST_SPACINGANDGLYPHS: Short\n val ELEMENT_NODE: Short\n val
ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val

```

```

ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE: Short\n    val
PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val
DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val
DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    } \n} \n \n /** \n * Exposes the JavaScript
[SVGTextPathElement](https://developer.mozilla.org/en/docs/Web/API/SVGTextPathElement) to Kotlin \n
*/ \n public external abstract class SVGTextPathElement : SVGTextContentElement, SVGURIReference { \n    open
val startOffset: SVGAnimatedLength \n    open val method: SVGAnimatedEnumeration \n    open val spacing:
SVGAnimatedEnumeration \n \n    companion object { \n        val TEXTPATH_METHODTYPE_UNKNOWN:
Short\n        val TEXTPATH_METHODTYPE_ALIGN: Short\n        val
TEXTPATH_METHODTYPE_STRETCH: Short\n        val TEXTPATH_SPACINGTYPE_UNKNOWN: Short\n
        val TEXTPATH_SPACINGTYPE_AUTO: Short\n        val TEXTPATH_SPACINGTYPE_EXACT: Short\n
        val LENGTHADJUST_UNKNOWN: Short\n        val LENGTHADJUST_SPACING: Short\n        val
LENGTHADJUST_SPACINGANDGLYPHS: Short\n        val ELEMENT_NODE: Short\n        val
ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val
ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val
PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    } \n} \n \n /** \n * Exposes the JavaScript
[SVGImageElement](https://developer.mozilla.org/en/docs/Web/API/SVGImageElement) to Kotlin \n
*/ \n public external abstract class SVGImageElement : SVGGraphicsElement, SVGURIReference,
HTMLorSVGImageElement { \n    open val x: SVGAnimatedLength \n    open val y: SVGAnimatedLength \n    open
val width: SVGAnimatedLength \n    open val height: SVGAnimatedLength \n    open val preserveAspectRatio:
SVGAnimatedPreserveAspectRatio \n    open var crossOrigin: String? \n \n    companion object { \n        val
ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val
CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE:
Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    } \n} \n \n /** \n * Exposes the JavaScript
[SVGForeignObjectElement](https://developer.mozilla.org/en/docs/Web/API/SVGForeignObjectElement) to
Kotlin \n
*/ \n public external abstract class SVGForeignObjectElement : SVGGraphicsElement { \n    open val x:
SVGAnimatedLength \n    open val y: SVGAnimatedLength \n    open val width: SVGAnimatedLength \n    open val
height: SVGAnimatedLength \n \n    companion object { \n        val ELEMENT_NODE: Short\n        val
ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val
ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val
PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val

```

```

DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val
DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }n}\n\npublic external abstract class
SVGMarkerElement : SVGElement, SVGFitToViewBox {\n    open val refX: SVGAnimatedLength\n    open val
refY: SVGAnimatedLength\n    open val markerUnits: SVGAnimatedEnumeration\n    open val markerWidth:
SVGAnimatedLength\n    open val markerHeight: SVGAnimatedLength\n    open val orientType:
SVGAnimatedEnumeration\n    open val orientAngle: SVGAnimatedAngle\n    open var orient: String\n    fun
setOrientToAuto()\n    fun setOrientToAngle(angle: SVGAngle)\n\n    companion object {\n        val
SVG_MARKERUNITS_UNKNOWN: Short\n        val SVG_MARKERUNITS_USERSPACEONUSE: Short\n
        val SVG_MARKERUNITS_STROKEWIDTH: Short\n        val SVG_MARKER_ORIENT_UNKNOWN: Short\n
        val SVG_MARKER_ORIENT_AUTO: Short\n        val SVG_MARKER_ORIENT_ANGLE: Short\n        val
ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val
CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE:
Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }n}\n\n/**\n * Exposes the JavaScript
[SVGSolidcolorElement](https://developer.mozilla.org/en/docs/Web/API/SVGSolidcolorElement) to Kotlin\n
*/\npublic external abstract class SVGSolidcolorElement : SVGElement {\n    companion object {\n        val
ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val
CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE:
Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }n}\n\n/**\n * Exposes the JavaScript
[SVGGradientElement](https://developer.mozilla.org/en/docs/Web/API/SVGGradientElement) to Kotlin\n
*/\npublic external abstract class SVGGradientElement : SVGElement, SVGURIReference, SVGUnitTypes {\n
    open val gradientUnits: SVGAnimatedEnumeration\n    open val gradientTransform: SVGAnimatedTransformList\n
    open val spreadMethod: SVGAnimatedEnumeration\n\n    companion object {\n        val
SVG_SPREADMETHOD_UNKNOWN: Short\n        val SVG_SPREADMETHOD_PAD: Short\n        val
SVG_SPREADMETHOD_REFLECT: Short\n        val SVG_SPREADMETHOD_REPEAT: Short\n        val
SVG_UNIT_TYPE_UNKNOWN: Short\n        val SVG_UNIT_TYPE_USERSPACEONUSE: Short\n        val
SVG_UNIT_TYPE_OBJECTBOUNDINGBOX: Short\n        val ELEMENT_NODE: Short\n        val
ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val
ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val
PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n

```

```

    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[SVGLinearGradientElement](https://developer.mozilla.org/en/docs/Web/API/SVGLinearGradientElement) to
Kotlin\n */\npublic external abstract class SVGLinearGradientElement : SVGGradientElement {\n    open val x1:
SVGAnimatedLength\n    open val y1: SVGAnimatedLength\n    open val x2: SVGAnimatedLength\n    open val
y2: SVGAnimatedLength\n\n    companion object {\n        val SVG_SPREADMETHOD_UNKNOWN: Short\n        val
SVG_SPREADMETHOD_PAD: Short\n        val SVG_SPREADMETHOD_REFLECT: Short\n        val
SVG_SPREADMETHOD_REPEAT: Short\n        val SVG_UNIT_TYPE_UNKNOWN: Short\n        val
SVG_UNIT_TYPE_USERSPACEONUSE: Short\n        val SVG_UNIT_TYPE_OBJECTBOUNDINGBOX:
Short\n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n
        val CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val
ENTITY_NODE: Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE:
Short\n        val DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[SVGRadialGradientElement](https://developer.mozilla.org/en/docs/Web/API/SVGRadialGradientElement) to
Kotlin\n */\npublic external abstract class SVGRadialGradientElement : SVGGradientElement {\n    open val cx:
SVGAnimatedLength\n    open val cy: SVGAnimatedLength\n    open val r: SVGAnimatedLength\n    open val fx:
SVGAnimatedLength\n    open val fy: SVGAnimatedLength\n    open val fr: SVGAnimatedLength\n\n    companion
object {\n        val SVG_SPREADMETHOD_UNKNOWN: Short\n        val SVG_SPREADMETHOD_PAD:
Short\n        val SVG_SPREADMETHOD_REFLECT: Short\n        val SVG_SPREADMETHOD_REPEAT:
Short\n        val SVG_UNIT_TYPE_UNKNOWN: Short\n        val SVG_UNIT_TYPE_USERSPACEONUSE:
Short\n        val SVG_UNIT_TYPE_OBJECTBOUNDINGBOX: Short\n        val ELEMENT_NODE: Short\n
        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n
        val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val
PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\npublic external abstract class
SVGMeshGradientElement : SVGGradientElement {\n    companion object {\n        val
SVG_SPREADMETHOD_UNKNOWN: Short\n        val SVG_SPREADMETHOD_PAD: Short\n        val
SVG_SPREADMETHOD_REFLECT: Short\n        val SVG_SPREADMETHOD_REPEAT: Short\n        val
SVG_UNIT_TYPE_UNKNOWN: Short\n        val SVG_UNIT_TYPE_USERSPACEONUSE: Short\n        val
SVG_UNIT_TYPE_OBJECTBOUNDINGBOX: Short\n        val ELEMENT_NODE: Short\n        val
ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val
ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val
PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val

```

```

DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\npublic external abstract class
SVGMeshrowElement : SVGElement {\n companion object {\n val ELEMENT_NODE: Short\n val
ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\npublic external abstract class
SVGMeshpatchElement : SVGElement {\n companion object {\n val ELEMENT_NODE: Short\n val
ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[SVGStopElement](https://developer.mozilla.org/en/docs/Web/API/SVGStopElement) to Kotlin\n *\npublic
external abstract class SVGStopElement : SVGElement {\n open val offset: SVGAnimatedNumber\n\n
companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val
TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE:
Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val
COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n
val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[SVGPatternElement](https://developer.mozilla.org/en/docs/Web/API/SVGPatternElement) to Kotlin\n *\npublic
external abstract class SVGPatternElement : SVGElement, SVGFitToViewBox, SVGURIReference,
SVGUnitTypes {\n open val patternUnits: SVGAnimatedEnumeration\n open val patternContentUnits:
SVGAnimatedEnumeration\n open val patternTransform: SVGAnimatedTransformList\n open val x:
SVGAnimatedLength\n open val y: SVGAnimatedLength\n open val width: SVGAnimatedLength\n open val
height: SVGAnimatedLength\n\n companion object {\n val SVG_UNIT_TYPE_UNKNOWN: Short\n
val SVG_UNIT_TYPE_USERSPACEONUSE: Short\n val SVG_UNIT_TYPE_OBJECTBOUNDINGBOX:
Short\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n
val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val
ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE:
Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\npublic external abstract class

```



```

SVGAnimatedString companion object {
    val ELEMENT_NODE: Short
    val ATTRIBUTE_NODE: Short
    val TEXT_NODE: Short
    val CDATA_SECTION_NODE: Short
    val ENTITY_REFERENCE_NODE: Short
    val ENTITY_NODE: Short
    val PROCESSING_INSTRUCTION_NODE: Short
    val COMMENT_NODE: Short
    val DOCUMENT_NODE: Short
    val DOCUMENT_TYPE_NODE: Short
    val DOCUMENT_FRAGMENT_NODE: Short
    val NOTATION_NODE: Short
    val DOCUMENT_POSITION_DISCONNECTED: Short
    val DOCUMENT_POSITION_PRECEDING: Short
    val DOCUMENT_POSITION_FOLLOWING: Short
    val DOCUMENT_POSITION_CONTAINS: Short
    val DOCUMENT_POSITION_CONTAINED_BY: Short
}

DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short }

/** Exposes the JavaScript [SVGViewElement](https://developer.mozilla.org/en/docs/Web/API/SVGViewElement) to Kotlin
 * public external abstract class SVGViewElement : SVGElement, SVGFitToViewBox, SVGZoomAndPan companion
 * object {
    val SVG_ZOOMANDPAN_UNKNOWN: Short
    val SVG_ZOOMANDPAN_DISABLE: Short
    val SVG_ZOOMANDPAN_MAGNIFY: Short
    val ELEMENT_NODE: Short
    val ATTRIBUTE_NODE: Short
    val TEXT_NODE: Short
    val CDATA_SECTION_NODE: Short
    val ENTITY_REFERENCE_NODE: Short
    val ENTITY_NODE: Short
    val PROCESSING_INSTRUCTION_NODE: Short
    val COMMENT_NODE: Short
    val DOCUMENT_NODE: Short
    val DOCUMENT_TYPE_NODE: Short
    val DOCUMENT_FRAGMENT_NODE: Short
    val NOTATION_NODE: Short
    val DOCUMENT_POSITION_DISCONNECTED: Short
    val DOCUMENT_POSITION_PRECEDING: Short
    val DOCUMENT_POSITION_FOLLOWING: Short
    val DOCUMENT_POSITION_CONTAINS: Short
    val DOCUMENT_POSITION_CONTAINED_BY: Short
}

DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short }

/** Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 * // NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!
 * // See github.com/kotlin/dukat for details
 * package org.w3c.files
 * import kotlin.js.*
 * import org.khronos.webgl.*
 * import org.w3c.dom.*
 * import org.w3c.dom.events.*
 * import org.w3c.xhr.*
 * /** Exposes the JavaScript [Blob](https://developer.mozilla.org/en/docs/Web/API/Blob) to Kotlin
 * public external open class Blob(blobParts: Array<dynamic> = definedExternally, options: BlobPropertyBag = definedExternally) : MediaProvider, ImageBitmapSource {
    open val size: Number
    open val type: String
    open val isClosed: Boolean
    fun slice(start: Int = definedExternally, end: Int = definedExternally, contentType: String = definedExternally): Blob
    fun close()
}

public external interface BlobPropertyBag {
    var type: String? = ""
    fun get() = definedExternally
    fun set(value) = definedExternally
}

@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline fun BlobPropertyBag(type: String? = ""): BlobPropertyBag {
    val o = js("{}")
    o["type"] = type
    return o
}

/** Exposes the JavaScript [File](https://developer.mozilla.org/en/docs/Web/API/File) to Kotlin
 * public external open class File(fileBits: Array<dynamic>, fileName: String, options: FilePropertyBag = definedExternally) : Blob {
    open val name: String
    open val lastModified: Int
}

public external interface FilePropertyBag : BlobPropertyBag {
    var lastModified: Int?
    fun get() = definedExternally
    fun set(value) = definedExternally
}

@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline fun FilePropertyBag(lastModified: Int? = undefined, type: String? = ""): FilePropertyBag {
    val o = js("{}")
    o["lastModified"] = lastModified
    o["type"] = type
    return o
}

/** Exposes the JavaScript [FileList](https://developer.mozilla.org/en/docs/Web/API/FileList) to Kotlin
 * public external abstract class FileList : ItemArrayLike<File> {
    override fun item(index: Int): File?
}

@Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")
@kotlin.internal.InlineOnly
public inline operator fun FileList.get(index: Int): File?

```

```

= asDynamic()[index]\n\n/**\n * Exposes the JavaScript
[FileReader](https://developer.mozilla.org/en/docs/Web/API/FileReader) to Kotlin\n */\npublic external open class
FileReader : EventTarget {\n    open val readyState: Short\n    open val result: dynamic\n    open val error:
dynamic\n    var onloadstart: ((ProgressEvent) -> dynamic)?\n    var onprogress: ((ProgressEvent) -> dynamic)?\n
var onload: ((Event) -> dynamic)?\n    var onabort: ((Event) -> dynamic)?\n    var onerror: ((Event) -> dynamic)?\n
var onloadend: ((Event) -> dynamic)?\n    fun readAsArrayBuffer(blob: Blob)\n    fun readAsBinaryString(blob:
Blob)\n    fun readAsText(blob: Blob, label: String = definedExternally)\n    fun readAsDataURL(blob: Blob)\n
fun abort()\n\n    companion object {\n        val EMPTY: Short\n        val LOADING: Short\n        val DONE:
Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[FileReaderSync](https://developer.mozilla.org/en/docs/Web/API/FileReaderSync) to Kotlin\n */\npublic external
open class FileReaderSync {\n    fun readAsArrayBuffer(blob: Blob): ArrayBuffer\n    fun readAsBinaryString(blob:
Blob): String\n    fun readAsText(blob: Blob, label: String = definedExternally): String\n    fun
readAsDataURL(blob: Blob): String\n}\n\n/**\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\n// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!\n\n// See
github.com/kotlin/dukat for details\n\npackage org.w3c.notifications\n\nimport kotlin.js.*\nimport
org.khronos.webgl.*\nimport org.w3c.dom.events.*\nimport org.w3c.workers.*\n\n/**\n * Exposes the JavaScript
[Notification](https://developer.mozilla.org/en/docs/Web/API/Notification) to Kotlin\n */\npublic external open
class Notification(title: String, options: NotificationOptions = definedExternally) : EventTarget {\n    var onclick:
((MouseEvent) -> dynamic)?\n    var onerror: ((Event) -> dynamic)?\n    open val title: String\n    open val dir:
NotificationDirection\n    open val lang: String\n    open val body: String\n    open val tag: String\n    open val
image: String\n    open val icon: String\n    open val badge: String\n    open val sound: String\n    open val vibrate:
Array<out Int>\n    open val timestamp: Number\n    open val renotify: Boolean\n    open val silent: Boolean\n
open val noscreen: Boolean\n    open val requireInteraction: Boolean\n    open val sticky: Boolean\n    open val data:
Any?\n    open val actions: Array<out NotificationAction>\n    fun close()\n\n    companion object {\n        val
permission: NotificationPermission\n        val maxActions: Int\n        fun requestPermission(deprecatedCallback:
(NotificationPermission) -> Unit = definedExternally): Promise<NotificationPermission>\n    }\n}\n\npublic
external interface NotificationOptions {\n    var dir: NotificationDirection? /* = NotificationDirection.AUTO */\n
get() = definedExternally\n    set(value) = definedExternally\n    var lang: String? /* = "" */\n    get() =
definedExternally\n    set(value) = definedExternally\n    var body: String? /* = "" */\n    get() =
definedExternally\n    set(value) = definedExternally\n    var tag: String? /* = "" */\n    get() =
definedExternally\n    set(value) = definedExternally\n    var image: String?\n    get() = definedExternally\n
set(value) = definedExternally\n    var icon: String?\n    get() = definedExternally\n    set(value) =
definedExternally\n    var badge: String?\n    get() = definedExternally\n    set(value) = definedExternally\n
var sound: String?\n    get() = definedExternally\n    set(value) = definedExternally\n    var vibrate: dynamic\n
get() = definedExternally\n    set(value) = definedExternally\n    var timestamp: Number?\n    get() =
definedExternally\n    set(value) = definedExternally\n    var renotify: Boolean? /* = false */\n    get() =
definedExternally\n    set(value) = definedExternally\n    var silent: Boolean? /* = false */\n    get() =
definedExternally\n    set(value) = definedExternally\n    var noscreen: Boolean? /* = false */\n    get() =
definedExternally\n    set(value) = definedExternally\n    var requireInteraction: Boolean? /* = false */\n    get()
= definedExternally\n    set(value) = definedExternally\n    var sticky: Boolean? /* = false */\n    get() =
definedExternally\n    set(value) = definedExternally\n    var data: Any? /* = null */\n    get() =
definedExternally\n    set(value) = definedExternally\n    var actions: Array<NotificationAction>? /* = arrayOf()
*/\n    get() = definedExternally\n    set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n\n@kotlin.internal.InlineOnly\n\npublic inline fun NotificationOptions(dir:
NotificationDirection? = NotificationDirection.AUTO, lang: String? = \"\", body: String? = \"\", tag: String? = \"\",
image: String? = undefined, icon: String? = undefined, badge: String? = undefined, sound: String? = undefined,

```

```

vibrate: dynamic = undefined, timestamp: Number? = undefined, renotify: Boolean? = false, silent: Boolean? =
false, noscreen: Boolean? = false, requireInteraction: Boolean? = false, sticky: Boolean? = false, data: Any? = null,
actions: Array<NotificationAction>? = arrayOf(): NotificationOptions { \n  val o = js("{}") \n  o["dir"] = dir \n
  o["lang"] = lang \n  o["body"] = body \n  o["tag"] = tag \n  o["image"] = image \n  o["icon"] = icon \n
o["badge"] = badge \n  o["sound"] = sound \n  o["vibrate"] = vibrate \n  o["timestamp"] = timestamp \n
o["renotify"] = renotify \n  o["silent"] = silent \n  o["noscreen"] = noscreen \n  o["requireInteraction"] =
requireInteraction \n  o["sticky"] = sticky \n  o["data"] = data \n  o["actions"] = actions \n  return
o \n } \n \n public external interface NotificationAction { \n  var action: String? \n  var title: String? \n  var icon:
String? \n  get() = definedExternally \n  set(value) =
definedExternally \n } \n \n @Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER") \n @kotlin.internal.InlineOnly \n public inline fun NotificationAction(action: String?,
title: String?, icon: String? = undefined): NotificationAction { \n  val o = js("{}") \n  o["action"] = action \n
o["title"] = title \n  o["icon"] = icon \n  return o \n } \n \n public external interface GetNotificationOptions { \n  var
tag: String? /* = "" */ \n  get() = definedExternally \n  set(value) =
definedExternally \n } \n \n @Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER") \n @kotlin.internal.InlineOnly \n public inline fun GetNotificationOptions(tag: String? =
""): GetNotificationOptions { \n  val o = js("{}") \n  o["tag"] = tag \n  return o \n } \n \n /** \n * Exposes the
JavaScript [NotificationEvent](https://developer.mozilla.org/en/docs/Web/API/NotificationEvent) to Kotlin \n
*/ \n public external open class NotificationEvent(type: String, eventInitDict: NotificationEventInit) :
ExtendableEvent { \n  open val notification: Notification \n  open val action: String \n \n  companion object { \n
val NONE: Short \n  val CAPTURING_PHASE: Short \n  val AT_TARGET: Short \n  val
BUBBLING_PHASE: Short \n  } \n } \n \n public external interface NotificationEventInit : ExtendableEventInit { \n
var notification: Notification? \n  var action: String? /* = "" */ \n  get() = definedExternally \n  set(value) =
definedExternally \n } \n \n @Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER") \n @kotlin.internal.InlineOnly \n public inline fun NotificationEventInit(notification:
Notification?, action: String? = "", bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? =
false): NotificationEventInit { \n  val o = js("{}") \n  o["notification"] = notification \n  o["action"] =
action \n  o["bubbles"] = bubbles \n  o["cancelable"] = cancelable \n  o["composed"] = composed \n  return
o \n } \n \n /* please, don't implement this interface!
*/ \n @JsName("null") \n @Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE") \n public external
interface NotificationPermission { \n  companion object \n } \n \n public inline val
NotificationPermission.Companion.DEFAULT: NotificationPermission get() =
"default".asDynamic().unsafeCast<NotificationPermission>() \n \n public inline val
NotificationPermission.Companion.DENIED: NotificationPermission get() =
"denied".asDynamic().unsafeCast<NotificationPermission>() \n \n public inline val
NotificationPermission.Companion.GRANTED: NotificationPermission get() =
"granted".asDynamic().unsafeCast<NotificationPermission>() \n \n /* please, don't implement this interface!
*/ \n @JsName("null") \n @Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE") \n public external
interface NotificationDirection { \n  companion object \n } \n \n public inline val
NotificationDirection.Companion.AUTO: NotificationDirection get() =
"auto".asDynamic().unsafeCast<NotificationDirection>() \n \n public inline val
NotificationDirection.Companion.LTR: NotificationDirection get() =
"ltr".asDynamic().unsafeCast<NotificationDirection>() \n \n public inline val
NotificationDirection.Companion.RTL: NotificationDirection get() =
"rtl".asDynamic().unsafeCast<NotificationDirection>() \n \n /* Copyright 2010-2021 JetBrains s.r.o. and Kotlin
Programming Language contributors. \n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file. \n * \n \n // NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT! \n //
See github.com/kotlin/dukat for details \n \n package org.w3c.workers \n \n import kotlin.js.* \n import

```

```

org.khronos.webgl.*\nimport org.w3c.dom.*\nimport org.w3c.dom.events.*\nimport org.w3c.fetch.*\nimport
org.w3c.notifications.*\n\n/**\n * Exposes the JavaScript
[ServiceWorker](https://developer.mozilla.org/en/docs/Web/API/ServiceWorker) to Kotlin\n *\npublic external
abstract class ServiceWorker : EventTarget, AbstractWorker, UnionMessagePortOrServiceWorker,
UnionClientOrMessagePortOrServiceWorker {\n    open val scriptURL: String\n    open val state:
ServiceWorkerState\n    open var onstatechange: ((Event) -> dynamic)?\n    fun postMessage(message: Any?,
transfer: Array<dynamic> = definedExternally)\n}\n\n/**\n * Exposes the JavaScript
[ServiceWorkerRegistration](https://developer.mozilla.org/en/docs/Web/API/ServiceWorkerRegistration) to
Kotlin\n *\npublic external abstract class ServiceWorkerRegistration : EventTarget {\n    open val installing:
ServiceWorker?\n    open val waiting: ServiceWorker?\n    open val active: ServiceWorker?\n    open val scope:
String\n    open var onupdatefound: ((Event) -> dynamic)?\n    open val APISpace: dynamic\n    fun update():
Promise<Unit>\n    fun unregister(): Promise<Boolean>\n    fun showNotification(title: String, options:
NotificationOptions = definedExternally): Promise<Unit>\n    fun getNotifications(filter: GetNotificationOptions =
definedExternally): Promise<Array<Notification>>\n    fun methodName(): Promise<dynamic>\n}\n\n/**\n *
Exposes the JavaScript
[ServiceWorkerContainer](https://developer.mozilla.org/en/docs/Web/API/ServiceWorkerContainer) to Kotlin\n
*\npublic external abstract class ServiceWorkerContainer : EventTarget {\n    open val controller:
ServiceWorker?\n    open val ready: Promise<ServiceWorkerRegistration>\n    open var oncontrollerchange:
((Event) -> dynamic)?\n    open var onmessage: ((MessageEvent) -> dynamic)?\n    fun register(scriptURL: String,
options: RegistrationOptions = definedExternally): Promise<ServiceWorkerRegistration>\n    fun
getRegistration(clientURL: String = definedExternally): Promise<Any?>\n    fun getRegistrations():
Promise<Array<ServiceWorkerRegistration>>\n    fun startMessages()\n}\n\npublic external interface
RegistrationOptions {\n    var scope: String?\n    get() = definedExternally\n    set(value) = definedExternally\n
    var type: WorkerType? /* = WorkerType.CLASSIC */\n    get() = definedExternally\n    set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun RegistrationOptions(scope: String? =
undefined, type: WorkerType? = WorkerType.CLASSIC): RegistrationOptions {\n    val o = js(\"{\}\")\n    o[\"scope\"] = scope\n    o[\"type\"] = type\n    return o\n}\n\n/**\n * Exposes the JavaScript
[ServiceWorkerMessageEvent](https://developer.mozilla.org/en/docs/Web/API/ServiceWorkerMessageEvent) to
Kotlin\n *\npublic external open class ServiceWorkerMessageEvent(type: String, eventInitDict:
ServiceWorkerMessageEventInit = definedExternally) : Event {\n    open val data: Any?\n    open val origin:
String\n    open val lastEventId: String\n    open val source: UnionMessagePortOrServiceWorker?\n    open val
ports: Array<out MessagePort>?\n    companion object {\n        val NONE: Short\n        val
CAPTURING_PHASE: Short\n        val AT_TARGET: Short\n        val BUBBLING_PHASE: Short\n    }\n}\n\npublic external interface ServiceWorkerMessageEventInit : EventInit {\n    var data: Any?\n    get() =
definedExternally\n    set(value) = definedExternally\n    var origin: String?\n    get() = definedExternally\n
set(value) = definedExternally\n    var lastEventId: String?\n    get() = definedExternally\n    set(value) =
definedExternally\n    var source: UnionMessagePortOrServiceWorker?\n    get() = definedExternally\n
set(value) = definedExternally\n    var ports: Array<MessagePort>?\n    get() = definedExternally\n    set(value)
= definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ServiceWorkerMessageEventInit(data:
Any? = undefined, origin: String? = undefined, lastEventId: String? = undefined, source:
UnionMessagePortOrServiceWorker? = undefined, ports: Array<MessagePort>? = undefined, bubbles: Boolean? =
false, cancelable: Boolean? = false, composed: Boolean? = false): ServiceWorkerMessageEventInit {\n    val o =
js(\"{\}\")\n    o[\"data\"] = data\n    o[\"origin\"] = origin\n    o[\"lastEventId\"] = lastEventId\n    o[\"source\"] =
source\n    o[\"ports\"] = ports\n    o[\"bubbles\"] = bubbles\n    o[\"cancelable\"] = cancelable\n    o[\"composed\"] =
composed\n    return o\n}\n\n/**\n * Exposes the JavaScript
[ServiceWorkerGlobalScope](https://developer.mozilla.org/en/docs/Web/API/ServiceWorkerGlobalScope) to

```

```

Kotlin\n *\npublic external abstract class ServiceWorkerGlobalScope : WorkerGlobalScope {\n  open val clients:
Clients\n  open val registration: ServiceWorkerRegistration\n  open var oninstall: ((Event) -> dynamic)?\n  open
var onactivate: ((Event) -> dynamic)?\n  open var onfetch: ((FetchEvent) -> dynamic)?\n  open var
onforeignfetch: ((Event) -> dynamic)?\n  open var onmessage: ((MessageEvent) -> dynamic)?\n  open var
onnotificationclick: ((NotificationEvent) -> dynamic)?\n  open var onnotificationclose: ((NotificationEvent) ->
dynamic)?\n  open var onfunctionalevent: ((Event) -> dynamic)?\n  fun skipWaiting():
Promise<Unit>\n}\n\n/**\n * Exposes the JavaScript
[Client](https://developer.mozilla.org/en/docs/Web/API/Client) to Kotlin\n *\npublic external abstract class Client :
UnionClientOrMessagePortOrServiceWorker {\n  open val url: String\n  open val frameType: FrameType\n
open val id: String\n  fun postMessage(message: Any?, transfer: Array<dynamic> = definedExternally)\n}\n\n/**\n
* Exposes the JavaScript [WindowClient](https://developer.mozilla.org/en/docs/Web/API/WindowClient) to
Kotlin\n *\npublic external abstract class WindowClient : Client {\n  open val visibilityState: dynamic\n  open
val focused: Boolean\n  fun focus(): Promise<WindowClient>\n  fun navigate(url: String):
Promise<WindowClient>\n}\n\n/**\n * Exposes the JavaScript
[Clients](https://developer.mozilla.org/en/docs/Web/API/Clients) to Kotlin\n *\npublic external abstract class
Clients {\n  fun get(id: String): Promise<Any?>\n  fun matchAll(options: ClientQueryOptions =
definedExternally): Promise<Array<Client>>\n  fun openWindow(url: String): Promise<WindowClient?>\n  fun
claim(): Promise<Unit>\n}\n\npublic external interface ClientQueryOptions {\n  var includeUncontrolled:
Boolean? /* = false *\n  get() = definedExternally\n  set(value) = definedExternally\n  var type:
ClientType? /* = ClientType.WINDOW *\n  get() = definedExternally\n  set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun
ClientQueryOptions(includeUncontrolled: Boolean? = false, type: ClientType? = ClientType.WINDOW):
ClientQueryOptions {\n  val o = js(\"({})\")\n  o[\"includeUncontrolled\"] = includeUncontrolled\n  o[\"type\"] =
type\n  return o\n}\n\n/**\n * Exposes the JavaScript
[ExtendableEvent](https://developer.mozilla.org/en/docs/Web/API/ExtendableEvent) to Kotlin\n *\npublic external
open class ExtendableEvent(type: String, eventInitDict: ExtendableEventInit = definedExternally) : Event {\n  fun
waitUntil(f: Promise<Any?>)\n\n  companion object {\n    val NONE: Short\n    val CAPTURING_PHASE:
Short\n    val AT_TARGET: Short\n    val BUBBLING_PHASE: Short\n  }\n}\n\npublic external interface
ExtendableEventInit : EventInit\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ExtendableEventInit(bubbles:
Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): ExtendableEventInit {\n  val o =
js(\"({})\")\n  o[\"bubbles\"] = bubbles\n  o[\"cancelable\"] = cancelable\n  o[\"composed\"] = composed\n
return o\n}\n\n/**\n * Exposes the JavaScript
[InstallEvent](https://developer.mozilla.org/en/docs/Web/API/InstallEvent) to Kotlin\n *\npublic external open
class InstallEvent(type: String, eventInitDict: ExtendableEventInit = definedExternally) : ExtendableEvent {\n  fun
registerForeignFetch(options: ForeignFetchOptions)\n\n  companion object {\n    val NONE: Short\n    val
CAPTURING_PHASE: Short\n    val AT_TARGET: Short\n    val BUBBLING_PHASE: Short\n  }\n}\n\npublic external interface ForeignFetchOptions {\n  var scopes: Array<String>?\n  var origins:
Array<String>?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ForeignFetchOptions(scopes:
Array<String>?, origins: Array<String>?): ForeignFetchOptions {\n  val o = js(\"({})\")\n  o[\"scopes\"] =
scopes\n  o[\"origins\"] = origins\n  return o\n}\n\n/**\n * Exposes the JavaScript
[FetchEvent](https://developer.mozilla.org/en/docs/Web/API/FetchEvent) to Kotlin\n *\npublic external open class
FetchEvent(type: String, eventInitDict: FetchEventInit) : ExtendableEvent {\n  open val request: Request\n  open
val clientId: String?\n  open val isReload: Boolean\n  fun respondWith(r: Promise<Response>)\n\n  companion
object {\n    val NONE: Short\n    val CAPTURING_PHASE: Short\n    val AT_TARGET: Short\n    val
BUBBLING_PHASE: Short\n  }\n}\n\npublic external interface FetchEventInit : ExtendableEventInit {\n  var

```

```

request: Request? \n var clientId: String? /* = null */ \n get() = definedExternally \n set(value) =
definedExternally \n var isReload: Boolean? /* = false */ \n get() = definedExternally \n set(value) =
definedExternally \n \n \n @Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER") \n @kotlin.internal.InlineOnly \n public inline fun FetchEventInit(request: Request?,
clientId: String? = null, isReload: Boolean? = false, bubbles: Boolean? = false, cancelable: Boolean? = false,
composed: Boolean? = false): FetchEventInit { \n val o = js("{}") \n o["request"] = request \n o["clientId"]
= clientId \n o["isReload"] = isReload \n o["bubbles"] = bubbles \n o["cancelable"] = cancelable \n
o["composed"] = composed \n return o \n \n \n public external open class ForeignFetchEvent(type: String,
eventInitDict: ForeignFetchEventInit) : ExtendableEvent { \n open val request: Request \n open val origin:
String \n fun respondWith(r: Promise<ForeignFetchResponse>) \n \n companion object { \n val NONE:
Short \n val CAPTURING_PHASE: Short \n val AT_TARGET: Short \n val BUBBLING_PHASE:
Short \n } \n \n \n public external interface ForeignFetchEventInit : ExtendableEventInit { \n var request:
Request? \n var origin: String? /* = "null" */ \n get() = definedExternally \n set(value) =
definedExternally \n \n \n @Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER") \n @kotlin.internal.InlineOnly \n public inline fun ForeignFetchEventInit(request:
Request?, origin: String? = "null", bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? =
false): ForeignFetchEventInit { \n val o = js("{}") \n o["request"] = request \n o["origin"] = origin \n
o["bubbles"] = bubbles \n o["cancelable"] = cancelable \n o["composed"] = composed \n return
o \n \n \n public external interface ForeignFetchResponse { \n var response: Response? \n var origin: String? \n
get() = definedExternally \n set(value) = definedExternally \n var headers: Array<String>? \n get() =
definedExternally \n set(value) = definedExternally \n \n \n @Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER") \n @kotlin.internal.InlineOnly \n public inline fun ForeignFetchResponse(response:
Response?, origin: String? = undefined, headers: Array<String>? = undefined): ForeignFetchResponse { \n val o =
js("{}") \n o["response"] = response \n o["origin"] = origin \n o["headers"] = headers \n return
o \n \n \n /** \n * Exposes the JavaScript
[ExtendableMessageEvent](https://developer.mozilla.org/en/docs/Web/API/ExtendableMessageEvent) to Kotlin \n
*/ \n public external open class ExtendableMessageEvent(type: String, eventInitDict: ExtendableMessageEventInit =
definedExternally) : ExtendableEvent { \n open val data: Any? \n open val origin: String \n open val lastEventId:
String \n open val source: UnionClientOrMessagePortOrServiceWorker? \n open val ports: Array<out
MessagePort>? \n \n companion object { \n val NONE: Short \n val CAPTURING_PHASE: Short \n val
AT_TARGET: Short \n val BUBBLING_PHASE: Short \n } \n \n \n public external interface
ExtendableMessageEventInit : ExtendableEventInit { \n var data: Any? \n get() = definedExternally \n
set(value) = definedExternally \n var origin: String? \n get() = definedExternally \n set(value) =
definedExternally \n var lastEventId: String? \n get() = definedExternally \n set(value) =
definedExternally \n var source: UnionClientOrMessagePortOrServiceWorker? \n get() = definedExternally \n
set(value) = definedExternally \n var ports: Array<MessagePort>? \n get() = definedExternally \n
set(value) = definedExternally \n \n \n @Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER") \n @kotlin.internal.InlineOnly \n public inline fun ExtendableMessageEventInit(data:
Any? = undefined, origin: String? = undefined, lastEventId: String? = undefined, source:
UnionClientOrMessagePortOrServiceWorker? = undefined, ports: Array<MessagePort>? = undefined, bubbles:
Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): ExtendableMessageEventInit { \n
val o = js("{}") \n o["data"] = data \n o["origin"] = origin \n o["lastEventId"] = lastEventId \n
o["source"] = source \n o["ports"] = ports \n o["bubbles"] = bubbles \n o["cancelable"] = cancelable \n
o["composed"] = composed \n return o \n \n \n /** \n * Exposes the JavaScript
[Cache](https://developer.mozilla.org/en/docs/Web/API/Cache) to Kotlin \n */ \n public external abstract class Cache
{ \n fun match(request: dynamic, options: CacheQueryOptions = definedExternally): Promise<Any?> \n fun
matchAll(request: dynamic = definedExternally, options: CacheQueryOptions = definedExternally):
Promise<Array<Response>> \n fun add(request: dynamic): Promise<Unit> \n fun addAll(requests:

```

```

Array<dynamic>): Promise<Unit>\n fun put(request: dynamic, response: Response): Promise<Unit>\n fun
delete(request: dynamic, options: CacheQueryOptions = definedExternally): Promise<Boolean>\n fun
keys(request: dynamic = definedExternally, options: CacheQueryOptions = definedExternally):
Promise<Array<Request>>\n\n\npublic external interface CacheQueryOptions {\n var ignoreSearch: Boolean? /*
= false */\n get() = definedExternally\n set(value) = definedExternally\n var ignoreMethod: Boolean? /*
= false */\n get() = definedExternally\n set(value) = definedExternally\n var ignoreVary: Boolean? /*
= false */\n get() = definedExternally\n set(value) = definedExternally\n var cacheName: String?\n
get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun CacheQueryOptions(ignoreSearch:
Boolean? = false, ignoreMethod: Boolean? = false, ignoreVary: Boolean? = false, cacheName: String? = undefined):
CacheQueryOptions {\n val o = js(\"({})\")\n o[\"ignoreSearch\"] = ignoreSearch\n o[\"ignoreMethod\"] =
ignoreMethod\n o[\"ignoreVary\"] = ignoreVary\n o[\"cacheName\"] = cacheName\n return o\n}\n\npublic
external interface CacheBatchOperation {\n var type: String?\n get() = definedExternally\n set(value) =
definedExternally\n var request: Request?\n get() = definedExternally\n set(value) = definedExternally\n
var response: Response?\n get() = definedExternally\n set(value) = definedExternally\n var options:
CacheQueryOptions?\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun CacheBatchOperation(type: String? =
undefined, request: Request? = undefined, response: Response? = undefined, options: CacheQueryOptions? =
undefined): CacheBatchOperation {\n val o = js(\"({})\")\n o[\"type\"] = type\n o[\"request\"] = request\n
o[\"response\"] = response\n o[\"options\"] = options\n return o\n}\n\n/*\n * Exposes the JavaScript
[CacheStorage](https://developer.mozilla.org/en/docs/Web/API/CacheStorage) to Kotlin\n */\n\npublic external
abstract class CacheStorage {\n fun match(request: dynamic, options: CacheQueryOptions = definedExternally):
Promise<Any?>\n fun has(cacheName: String): Promise<Boolean>\n fun open(cacheName: String):
Promise<Cache>\n fun delete(cacheName: String): Promise<Boolean>\n fun keys():
Promise<Array<String>>}\n\n\npublic external open class FunctionalEvent : ExtendableEvent {\n companion
object {\n val NONE: Short\n val CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val
BUBBLING_PHASE: Short\n }\n}\n\n\npublic external interface UnionMessagePortOrServiceWorker\n\n\npublic
external interface UnionClientOrMessagePortOrServiceWorker\n\n\n/* please, don't implement this interface!
*/\n\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\n\npublic external
interface ServiceWorkerState {\n companion object\n}\n\n\npublic inline val
ServiceWorkerState.Companion.INSTALLING: ServiceWorkerState get() =
\"installing\".asDynamic().unsafeCast<ServiceWorkerState>()\n\n\npublic inline val
ServiceWorkerState.Companion.INSTALLED: ServiceWorkerState get() =
\"installed\".asDynamic().unsafeCast<ServiceWorkerState>()\n\n\npublic inline val
ServiceWorkerState.Companion.ACTIVATING: ServiceWorkerState get() =
\"activating\".asDynamic().unsafeCast<ServiceWorkerState>()\n\n\npublic inline val
ServiceWorkerState.Companion.ACTIVATED: ServiceWorkerState get() =
\"activated\".asDynamic().unsafeCast<ServiceWorkerState>()\n\n\npublic inline val
ServiceWorkerState.Companion.REDUNDANT: ServiceWorkerState get() =
\"redundant\".asDynamic().unsafeCast<ServiceWorkerState>()\n\n\n/* please, don't implement this interface!
*/\n\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\n\n\npublic external
interface FrameType {\n companion object\n}\n\n\npublic inline val FrameType.Companion.AUXILIARY:
FrameType get() = \"auxiliary\".asDynamic().unsafeCast<FrameType>()\n\n\npublic inline val
FrameType.Companion.TOP_LEVEL: FrameType get() = \"top-
level\".asDynamic().unsafeCast<FrameType>()\n\n\npublic inline val FrameType.Companion.NESTED: FrameType
get() = \"nested\".asDynamic().unsafeCast<FrameType>()\n\n\npublic inline val FrameType.Companion.NONE:
FrameType get() = \"none\".asDynamic().unsafeCast<FrameType>()\n\n\n/* please, don't implement this interface!

```

```

*\n@jsName("null")\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\npublic external
interface ClientType {\n  companion object{\n}\n\npublic inline val ClientType.Companion.WINDOW: ClientType
get() = "window".asDynamic().unsafeCast<ClientType>()\n\npublic inline val ClientType.Companion.WORKER:
ClientType get() = "worker".asDynamic().unsafeCast<ClientType>()\n\npublic inline val
ClientType.Companion.SHAREDWORKER: ClientType get() =
"sharedworker".asDynamic().unsafeCast<ClientType>()\n\npublic inline val ClientType.Companion.ALL:
ClientType get() = "all".asDynamic().unsafeCast<ClientType>()"}/*\n * Copyright 2010-2021 JetBrains s.r.o. and
Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file.\n */\n\n// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT
EDIT!\n// See github.com/kotlin/dukat for details\n\npackage org.w3c.xhr\n\nimport kotlin.js.*\nimport
org.khronos.webgl.*\nimport org.w3c.dom.*\nimport org.w3c.dom.events.*\nimport org.w3c.files.*\n\n/**\n *
Exposes the JavaScript
[XMLHttpRequestEventTarget](https://developer.mozilla.org/en/docs/Web/API/XMLHttpRequestEventTarget) to
Kotlin\n */\n\npublic external abstract class XMLHttpRequestEventTarget : EventTarget {\n  open var onloadstart:
((ProgressEvent) -> dynamic)?\n  open var onprogress: ((ProgressEvent) -> dynamic)?\n  open var onabort:
((Event) -> dynamic)?\n  open var onerror: ((Event) -> dynamic)?\n  open var onload: ((Event) -> dynamic)?\n
open var ontimeout: ((Event) -> dynamic)?\n  open var onloadend: ((Event) -> dynamic)?\n}\n\npublic external
abstract class XMLHttpRequestUpload : XMLHttpRequestEventTarget{\n\n/**\n * Exposes the JavaScript
[XMLHttpRequest](https://developer.mozilla.org/en/docs/Web/API/XMLHttpRequest) to Kotlin\n */\n\npublic
external open class XMLHttpRequest : XMLHttpRequestEventTarget {\n  var onreadystatechange: ((Event) ->
dynamic)?\n  open val readyState: Short\n  var timeout: Int\n  var withCredentials: Boolean\n  open val upload:
XMLHttpRequestUpload\n  open val responseURL: String\n  open val status: Short\n  open val statusText:
String\n  var responseType: XMLHttpRequestResponseType\n  open val response: Any?\n  open val
responseText: String\n  open val responseXML: Document?\n  fun open(method: String, url: String)\n  fun
open(method: String, url: String, async: Boolean, username: String? = definedExternally, password: String? =
definedExternally)\n  fun setRequestHeader(name: String, value: String)\n  fun send(body: dynamic =
definedExternally)\n  fun abort()\n  fun getResponseHeader(name: String): String?\n  fun
getAllResponseHeaders(): String\n  fun overrideMimeType(mime: String)\n}\n\ncompanion object {\n  val
UNSENT: Short\n  val OPENED: Short\n  val HEADERS_RECEIVED: Short\n  val LOADING:
Short\n  val DONE: Short\n  }\n}\n\n/**\n * Exposes the JavaScript
[FormData](https://developer.mozilla.org/en/docs/Web/API/FormData) to Kotlin\n */\n\npublic external open class
FormData(form: HTMLFormElement = definedExternally) {\n  fun append(name: String, value: String)\n  fun
append(name: String, value: Blob, filename: String = definedExternally)\n  fun delete(name: String)\n  fun
get(name: String): dynamic\n  fun getAll(name: String): Array<dynamic>\n  fun has(name: String): Boolean\n
fun set(name: String, value: String)\n  fun set(name: String, value: Blob, filename: String =
definedExternally)\n}\n\n/**\n * Exposes the JavaScript
[ProgressEvent](https://developer.mozilla.org/en/docs/Web/API/ProgressEvent) to Kotlin\n */\n\npublic external open
class ProgressEvent(type: String, eventInitDict: ProgressEventInit = definedExternally) : Event {\n  open val
lengthComputable: Boolean\n  open val loaded: Number\n  open val total: Number\n\n  companion object {\n
val NONE: Short\n  val CAPTURING_PHASE: Short\n  val AT_TARGET: Short\n  val
BUBBLING_PHASE: Short\n  }\n}\n\npublic external interface ProgressEventInit : EventInit {\n  var
lengthComputable: Boolean? /* = false */\n  get() = definedExternally\n  set(value) = definedExternally\n
var loaded: Number? /* = 0 */\n  get() = definedExternally\n  set(value) = definedExternally\n  var total:
Number? /* = 0 */\n  get() = definedExternally\n  set(value) =
definedExternally\n}\n\n@Suppress("INVISIBLE_REFERENCE",
"INVISIBLE_MEMBER")\n@kotlin.internal.InlineOnly\npublic inline fun ProgressEventInit(lengthComputable:
Boolean? = false, loaded: Number? = 0, total: Number? = 0, bubbles: Boolean? = false, cancelable: Boolean? =
false, composed: Boolean? = false): ProgressEventInit {\n  val o = js("{}")\n  o["lengthComputable"] =

```

```

lengthComputable\n  o[\"loaded\"] = loaded\n  o[\"total\"] = total\n  o[\"bubbles\"] = bubbles\n
o[\"cancelable\"] = cancelable\n  o[\"composed\"] = composed\n  return o\n}\n\n/* please, don't implement this
interface! *\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic
external interface XMLHttpRequestResponseType {\n  companion object\n}\n\npublic inline val
XMLHttpRequestResponseType.Companion.EMPTY: XMLHttpRequestResponseType get() =
\"\".asDynamic().unsafeCast<XMLHttpRequestResponseType>()\n\npublic inline val
XMLHttpRequestResponseType.Companion.ARRAYBUFFER: XMLHttpRequestResponseType get() =
\"arraybuffer\".asDynamic().unsafeCast<XMLHttpRequestResponseType>()\n\npublic inline val
XMLHttpRequestResponseType.Companion.BLOB: XMLHttpRequestResponseType get() =
\"blob\".asDynamic().unsafeCast<XMLHttpRequestResponseType>()\n\npublic inline val
XMLHttpRequestResponseType.Companion.DOCUMENT: XMLHttpRequestResponseType get() =
\"document\".asDynamic().unsafeCast<XMLHttpRequestResponseType>()\n\npublic inline val
XMLHttpRequestResponseType.Companion.JSON: XMLHttpRequestResponseType get() =
\"json\".asDynamic().unsafeCast<XMLHttpRequestResponseType>()\n\npublic inline val
XMLHttpRequestResponseType.Companion.TEXT: XMLHttpRequestResponseType get() =
\"text\".asDynamic().unsafeCast<XMLHttpRequestResponseType>()\", \"/*\n * Copyright 2010-2018 JetBrains s.r.o.
and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license
that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin\n\nimport
kotlin.annotation.AnnotationRetention.BINARY\nimport kotlin.annotation.AnnotationRetention.SOURCE\nimport
kotlin.annotation.AnnotationTarget.*\nimport kotlin.internal.RequireKotlin\nimport
kotlin.internal.RequireKotlinVersionKind\nimport kotlin.reflect.KClass\n\n/**\n * Signals that the annotated
annotation class is a marker of an experimental API.\n * Any declaration annotated with that marker is
considered an experimental declaration\n * and its call sites should accept the experimental aspect of it either by
using [UseExperimental],\n * or by being annotated with that marker themselves, effectively causing further
propagation of that experimental aspect.\n * This class is deprecated in favor of a more general approach
provided by [RequiresOptIn]/[OptIn].\n
*/\n@Target(ANNOTATION_CLASS)\n@Retention(BINARY)\n@SinceKotlin(\"1.2\")\n@RequireKotlin(\"1.2.50
\", versionKind = RequireKotlinVersionKind.COMPILER_VERSION)\n@DeprecatedSinceKotlin(warningSince =
\"1.4\", errorSince = \"1.6\")\n@Deprecated(\"Please use RequiresOptIn instead.\")\npublic annotation class
Experimental(val level: Level = Level.ERROR) {\n  /**\n   * Severity of the diagnostic that should be reported on
usages of experimental API which did not explicitly accept the experimental aspect\n   * of that API either by using
[UseExperimental] or by being annotated with the corresponding marker annotation.\n   */\n  public enum class
Level {\n    /** Specifies that a warning should be reported on incorrect usages of this experimental API. */\n
WARNING,\n    /** Specifies that an error should be reported on incorrect usages of this experimental API. */\n
ERROR,\n  }\n}\n\n/**\n * Allows to use experimental API denoted by the given markers in the annotated file,
declaration, or expression.\n * If a declaration is annotated with [UseExperimental], its usages are not
required to opt-in to that experimental API.\n * This class is deprecated in favor of a more general approach
provided by [RequiresOptIn]/[OptIn].\n */\n@Target(\n  CLASS, PROPERTY, LOCAL_VARIABLE,
VALUE_PARAMETER, CONSTRUCTOR, FUNCTION, PROPERTY_GETTER, PROPERTY_SETTER,
EXPRESSION, FILE,
TYPEALIAS)\n@Retention(SOURCE)\n@SinceKotlin(\"1.2\")\n@RequireKotlin(\"1.2.50\", versionKind =
RequireKotlinVersionKind.COMPILER_VERSION)\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince =
\"1.6\")\n@Deprecated(\"Please use OptIn instead.\", ReplaceWith(\"OptIn(*markerClass)\",
\"kotlin.OptIn\"))\npublic annotation class UseExperimental(\n  vararg val markerClass: KClass<out
Annotation>)\n\n@Target(CLASS, PROPERTY, CONSTRUCTOR, FUNCTION,
TYPEALIAS)\n@Retention(BINARY)\ninternal annotation class WasExperimental(\n  vararg val markerClass:
KClass<out Annotation>)\n\n\", \"package kotlin\n\nimport kotlin.annotation.AnnotationTarget.*\n\n/**\n * This
annotation marks the standard library API that is considered experimental and is not subject to the\n * [general

```

compatibility guarantees](https://kotlinlang.org/docs/reference/evolution/components-stability.html) given for the standard library:

- * the behavior of such API may be changed or the API may be removed completely in any further release.
- * Beware using the annotated API especially if you're developing a library, since your library might become binary incompatible with the future versions of the standard library.
- * Any usage of a declaration annotated with `@ExperimentalStdlibApi` must be accepted either by
 - * annotating that usage with the `[OptIn]` annotation, e.g. `@OptIn(ExperimentalStdlibApi::class)`,
 - * or by using the compiler argument `-opt-in=kotlin.ExperimentalStdlibApi`.

```

@RequiresOptIn(level = RequiresOptIn.Level.ERROR)
@Retention(AnnotationRetention.BINARY)
@Target(
  CLASS,
  ANNOTATION_CLASS,
  PROPERTY,
  FIELD,
  LOCAL_VARIABLE,
  VALUE_PARAMETER,
  CONSTRUCTOR,
  FUNCTION,
  PROPERTY_GETTER,
  PROPERTY_SETTER,
  TYPEALIAS)
@MustBeDocumented
@SinceKotlin("1.3")
public annotation class ExperimentalStdlibApi
"/**
 * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */
package kotlin
import kotlin.annotation.AnnotationTarget
import kotlin.experimental.ExperimentalTypeInference
/**
 * Allows to infer generic type arguments of a function from the calls in the annotated function parameter of that function.
 * When this annotation is placed on a generic function parameter of a function, it enables to infer the type arguments of that generic function from the lambda body passed to that parameter.
 * The calls that affect inference are either members of the receiver type of an annotated function parameter or extensions for that type. The extensions must be themselves annotated with @BuilderInference.
 * Example: we declare
 * ```
 * fun <T> sequence(@BuilderInference block: suspend SequenceScope<T>().-> Unit): Sequence<T>
 * ```
 * and use it like
 * ```
 * val result = sequence { yield("result") }
 * ```
 * Here the type argument of the resulting sequence is inferred to `String` from the argument of the [SequenceScope.yield] function, that is called inside the lambda passed to [sequence].
 * Note: this annotation is experimental, see [ExperimentalTypeInference] on how to opt-in for it.
 */
@Target(VALUE_PARAMETER, FUNCTION, PROPERTY)
@Retention(AnnotationRetention.BINARY)
@SinceKotlin("1.3")
@ExperimentalTypeInference
public annotation class BuilderInference
/**
 * Enables overload selection based on the type of the value returned from lambda argument.
 * When two or more function overloads have otherwise the same parameter lists that differ only in the return type of a functional parameter, this annotation enables overload selection by the type of the value returned from the lambda function passed to this functional parameter.
 * Example:
 * ```
 * @OverloadResolutionByLambdaReturnType
 * fun create(intProducer: () -> Int): Int
 * fun create(doubleProducer: () -> Double): Double
 * val newValue = create { 3.14 }
 * ```
 * The annotation being applied to one of overloads allows to resolve this ambiguity by analyzing what value is returned from the lambda function.
 * This annotation is also used to discriminate the annotated overloads in case if overload selection still cannot choose one of them even taking in account the result of lambda parameter analysis. In that case a warning is reported.
 * Note: this annotation is experimental, see [ExperimentalTypeInference] on how to opt-in for it.
 */
@Target(FUNCTION)
@Retention(AnnotationRetention.BINARY)
@SinceKotlin("1.4")
@ExperimentalTypeInference
public annotation class OverloadResolutionByLambdaReturnType
"/**
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */
package kotlin
import kotlin.annotation.AnnotationTarget
import kotlin.internal.RequireKotlin
import kotlin.internal.RequireKotlinVersionKind
/**
 * The experimental multiplatform support API marker.
 * Any usage of a declaration annotated with @ExperimentalMultiplatform must be accepted either by
 * annotating that usage with the [OptIn] annotation, e.g. @OptIn(ExperimentalMultiplatform::class),
 * or by using the compiler argument -opt-in=kotlin.ExperimentalMultiplatform.
 */
@RequiresOptIn
@MustBeDocumented
@Target(
  CLASS,
  ANNOTATION_CLASS,
  PROPERTY,
  FIELD,
  LOCAL_VARIABLE,
  VALUE_PARAMETER,
  CONSTRUCTOR,

```

```

FUNCTION,\n PROPERTY_GETTER,\n PROPERTY_SETTER,\n
TYPEALIAS\n)\n@Retention(AnnotationRetention.BINARY)\n@RequireKotlin("1.2.50", versionKind =
RequireKotlinVersionKind.COMPILER_VERSION)\npublic annotation class ExperimentalMultiplatform\n\n/**\n * Marks an expected annotation class that it isn't required to have actual counterparts in all platforms.\n * This\n * annotation is only applicable to `expect` annotation classes in multi-platform projects and marks that class as\n * "optional".\n * Optional expected class is allowed to have no corresponding actual class on the platform. Optional\n * annotations can only be used\n * to annotate something, not as types in signatures. If an optional annotation has no\n * corresponding actual class on a platform,\n * the annotation entries where it's used are simply erased when\n * compiling code on that platform.\n * Note: this annotation is experimental, see [ExperimentalMultiplatform] on\n * how to opt-in for it.\n
*/\n@Target(ANNOTATION_CLASS)\n@Retention(AnnotationRetention.BINARY)\n@ExperimentalMultiplatfor
m\n@RequireKotlin("1.2.50", versionKind = RequireKotlinVersionKind.COMPILER_VERSION)\npublic
annotation class OptionalExpectation\n","/**\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\npackage kotlin\n\nimport
kotlin.annotation.AnnotationRetention.BINARY\nimport kotlin.annotation.AnnotationRetention.SOURCE\nimport
kotlin.annotation.AnnotationTarget.*\nimport kotlin.internal.RequireKotlin\nimport
kotlin.internal.RequireKotlinVersionKind\nimport kotlin.reflect.KClass\n\n/**\n * Signals that the annotated
annotation class is a marker of an API that requires an explicit opt-in.\n * Call sites of any declaration annotated
with that marker should opt in to the API either by using [OptIn],\n * or by being annotated with that marker
themselves, effectively causing further propagation of the opt-in requirement.\n * This class requires opt-in itself
and can only be used with the compiler argument `-opt-in=kotlin.RequiresOptIn`.\n * @property message
message to be reported on usages of API without an explicit opt-in, or empty string for the default message.\n *
The default message is: "This declaration is experimental and its usage should be marked with 'Marker'\n *
or '@OptIn(Marker::class)", where `Marker` is the opt-in requirement marker.\n * @property level specifies
how usages of API without an explicit opt-in are reported in code.\n
*/\n@Target(ANNOTATION_CLASS)\n@Retention(BINARY)\n@SinceKotlin("1.3")\n@RequireKotlin("1.3.70
", versionKind = RequireKotlinVersionKind.COMPILER_VERSION)\npublic annotation class RequiresOptIn(\n
val message: String = "",\n val level: Level = Level.ERROR)\n {\n /**\n * Severity of the diagnostic that
should be reported on usages which did not explicitly opted into\n * the API either by using [OptIn] or by being
annotated with the corresponding marker annotation.\n */\n public enum class Level {\n /** Specifies that a
warning should be reported on incorrect usages of this API. */\n WARNING,\n /** Specifies that an error
should be reported on incorrect usages of this API. */\n ERROR,\n } }\n\n/**\n * Allows to use the API
denoted by the given markers in the annotated file, declaration, or expression.\n * If a declaration is annotated with
[OptIn], its usages are **not** required to opt in to that API.\n * This class requires opt-in itself and can only be
used with the compiler argument `-opt-in=kotlin.RequiresOptIn`.\n */\n@Target(\n CLASS, PROPERTY,
LOCAL_VARIABLE, VALUE_PARAMETER, CONSTRUCTOR, FUNCTION, PROPERTY_GETTER,
PROPERTY_SETTER, EXPRESSION, FILE,
TYPEALIAS\n)\n@Retention(SOURCE)\n@SinceKotlin("1.3")\n@RequireKotlin("1.3.70", versionKind =
RequireKotlinVersionKind.COMPILER_VERSION)\npublic annotation class OptIn(\n vararg val markerClass:
KClass<out Annotation>\n)\n","/**\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\npackage kotlin.collections\n\nimport kotlin.js.JsName\n\n/**\n * Provides a skeletal
implementation of the read-only [Collection] interface.\n * @param E the type of elements contained in the
collection. The collection is covariant in its element type.\n */\n@SinceKotlin("1.1")\npublic abstract class
AbstractCollection<out E> protected constructor() : Collection<E> {\n abstract override val size: Int\n abstract
override fun iterator(): Iterator<E>\n\n override fun contains(element: @UnsafeVariance E): Boolean = any { it
== element }\n\n override fun containsAll(elements: Collection<@UnsafeVariance E>): Boolean =\n

```

```

elements.all { contains(it) } // use when js will support bound refs: elements.all(this::contains)\n\n override fun
isEmpty(): Boolean = size == 0\n\n override fun toString(): String = joinToString(", ", "[", "]") {\n if (it
=== this) \"(this Collection)\" else it.toString()\n }\n\n /**\n * Returns new array of type `Array<Any?>` with
the elements of this collection.\n *\n @JsName(\"toArray\")\n protected open fun toArray(): Array<Any?> =
copyToArrayImpl(this)\n\n /**\n * Fills the provided [array] or creates new array of the same type\n * and
fills it with the elements of this collection.\n *\n protected open fun <T> toArray(array: Array<T>): Array<T>
= copyToArrayImpl(this, array)\n}\n\", \"/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n *\n\npackage kotlin.collections\n\nprivate enum class State {\n Ready,\n
NotReady,\n Done,\n Failed\n}\n\n/**\n * A base class to simplify implementing iterators so that
implementations only have to implement [computeNext]\n * to implement the iterator, calling [done] when the
iteration is complete.\n *\npublic abstract class AbstractIterator<T> : Iterator<T> {\n private var state =
State.NotReady\n private var nextValue: T? = null\n\n override fun hasNext(): Boolean {\n require(state !=
State.Failed)\n return when (state) {\n State.Done -> false\n State.Ready -> true\n else ->
tryToComputeNext()\n }\n }\n\n override fun next(): T {\n if (!hasNext()) throw
NoSuchElementException()\n state = State.NotReady\n @Suppress(\"UNCHECKED_CAST\")\n return nextValue as T\n }\n\n private fun tryToComputeNext(): Boolean {\n state = State.Failed\n
computeNext()\n return state == State.Ready\n }\n\n /**\n * Computes the next item in the iterator.\n
*\n * This callback method should call one of these two methods:\n * * [setNext] with the next value of
the iteration\n * * [done] to indicate there are no more elements\n * * Failure to call either method will
result in the iteration terminating with a failed state\n *\n abstract protected fun computeNext(): Unit\n\n
/**\n * Sets the next value in the iteration, called from the [computeNext] function\n *\n protected fun
setNext(value: T): Unit {\n nextValue = value\n state = State.Ready\n }\n\n /**\n * Sets the state to
done so that the iteration terminates.\n *\n protected fun done() {\n state = State.Done\n }\n}\n\", \"/*\n
* Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code
is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n *\n\n/*\n * Based on
GWT AbstractList\n * Copyright 2007 Google Inc.\n *\n\npackage kotlin.collections\n\n/**\n * Provides a skeletal
implementation of the read-only [List] interface.\n *\n * This class is intended to help implementing read-only lists
so it doesn't support concurrent modification tracking.\n *\n * @param E the type of elements contained in the list.
The list is covariant in its element type.\n *\n @SinceKotlin(\"1.1\")\npublic abstract class AbstractList<out E>
protected constructor() : AbstractCollection<E>(), List<E> {\n abstract override val size: Int\n abstract override
fun get(index: Int): E\n\n override fun iterator(): Iterator<E> = IteratorImpl()\n\n override fun indexOf(element:
@UnsafeVariance E): Int = indexOfFirst { it == element }\n\n override fun lastIndexOf(element:
@UnsafeVariance E): Int = indexOfLast { it == element }\n\n override fun listIterator(): ListIterator<E> =
ListIteratorImpl(0)\n\n override fun listIterator(index: Int): ListIterator<E> = ListIteratorImpl(index)\n\n
override fun subList(fromIndex: Int, toIndex: Int): List<E> = SubList(this, fromIndex, toIndex)\n\n private class
SubList<out E>(private val list: AbstractList<E>, private val fromIndex: Int, toIndex: Int) : AbstractList<E>(),
RandomAccess {\n private var _size: Int = 0\n\n init {\n checkRangeIndexes(fromIndex, toIndex,
list.size)\n this._size = toIndex - fromIndex\n }\n\n override fun get(index: Int): E {\n
checkElementIndex(index, _size)\n return list[fromIndex + index]\n }\n\n override val size: Int
get() = _size\n }\n\n /**\n * Compares this list with other list instance with the ordered structural equality.\n
*\n * @return true, if [other] instance is a [List] of the same size, which contains the same elements in the same
order.\n *\n override fun equals(other: Any?): Boolean {\n if (other === this) return true\n if (other !is
List<*>) return false\n\n return orderedEquals(this, other)\n }\n\n /**\n * Returns the hash code value for
this list.\n *\n override fun hashCode(): Int = orderedHashCode(this)\n\n private open inner class IteratorImpl
: Iterator<E> {\n /** the index of the item that will be returned on the next call to [next]`() *\n protected
var index = 0\n\n override fun hasNext(): Boolean = index < size\n\n override fun next(): E {\n if
(!hasNext()) throw NoSuchElementException()\n return get(index++)\n }\n }\n\n /**\n *

```

```

Implementation of [ListIterator] for abstract lists.\n    */\n    private open inner class ListIteratorImpl(index: Int) :
IteratorImpl(), ListIterator<E> {\n\n        init {\n            checkPositionIndex(index, this@AbstractList.size)\n\n            this.index = index\n        }\n\n        override fun hasPrevious(): Boolean = index > 0\n\n        override fun
nextIndex(): Int = index\n\n        override fun previous(): E {\n            if (!hasPrevious()) throw
NoSuchElementException()\n            return get(--index)\n        }\n\n        override fun previousIndex(): Int = index -
1\n    }\n\n    internal companion object {\n        internal fun checkElementIndex(index: Int, size: Int) {\n            if
(index < 0 || index >= size) {\n                throw IndexOutOfBoundsException("index: $index, size: $size")\n            }\n        }\n\n        internal fun checkPositionIndex(index: Int, size: Int) {\n            if (index < 0 || index > size) {\n                throw IndexOutOfBoundsException("index: $index, size: $size")\n            }\n        }\n\n        internal fun checkRangeIndexes(fromIndex: Int, toIndex: Int, size: Int) {\n            if (fromIndex < 0 || toIndex > size) {\n                throw IndexOutOfBoundsException("fromIndex: $fromIndex, toIndex: $toIndex, size: $size")\n            }\n            if (fromIndex > toIndex) {\n                throw IllegalArgumentException("fromIndex: $fromIndex > toIndex:
$toIndex")\n            }\n        }\n\n        internal fun checkBoundsIndexes(startIndex: Int, endIndex: Int, size: Int) {\n            if (startIndex < 0 || endIndex > size) {\n                throw IndexOutOfBoundsException("startIndex:
$startIndex, endIndex: $endIndex, size: $size")\n            }\n            if (startIndex > endIndex) {\n                throw
IllegalArgumentException("startIndex: $startIndex > endIndex: $endIndex")\n            }\n        }\n\n        internal
fun orderedHashCode(c: Collection<*>): Int {\n            var hashCode = 1\n            for (e in c) {\n                hashCode = 31 * hashCode + (e?.hashCode() ?: 0)\n            }\n            return hashCode\n        }\n\n        internal fun
orderedEquals(c: Collection<*>, other: Collection<*>): Boolean {\n            if (c.size != other.size) return false\n\n            val otherIterator = other.iterator()\n            for (elem in c) {\n                val elemOther = otherIterator.next()\n                if (elem != elemOther) {\n                    return false\n                }\n                return true\n            }\n        }\n    }, "/*\n    * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n    * Use of this
source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n    */\n\n    *
Based on GWT AbstractMap\n    * Copyright 2007 Google Inc.\n    */\n\npackage kotlin.collections\n\n/**\n    * Provides
a skeletal implementation of the read-only [Map] interface.\n    * The implementor is required to implement
[entries] property, which should return read-only set of map entries.\n    * @param K the type of map keys. The
map is invariant in its key type.\n    * @param V the type of map values. The map is covariant in its value type.\n
    */\n\n@SinceKotlin("1.1")\npublic abstract class AbstractMap<K, out V> protected constructor() : Map<K, V>
{\n\n    override fun containsKey(key: K): Boolean {\n        return implFindEntry(key) != null\n    }\n\n    override
fun containsValue(value: @UnsafeVariance V): Boolean = entries.any { it.value == value }\n\n    internal fun
containsEntry(entry: Map.Entry<*, *>): Boolean {\n        // since entry comes from @UnsafeVariance parameters it
can be virtually anything\n        if (entry !is Map.Entry<*, *>) return false\n        val key = entry.key\n        val value
= entry.value\n        val ourValue = get(key)\n\n        if (value != ourValue) {\n            return false\n        }\n\n        //
Perhaps it was null and we don't contain the key?\n        if (ourValue == null && !containsKey(key)) {\n            return false\n        }\n        return true\n    }\n\n    /**\n        * Compares this map with other instance with the
ordered structural equality.\n        * @return true, if [other] instance is a [Map] of the same size, all entries of
which are contained in the [entries] set of this map.\n        */\n\n    override fun equals(other: Any?): Boolean {\n        if
(other === this) return true\n        if (other !is Map<*, *>) return false\n        if (size != other.size) return false\n\n        return other.entries.all { containsEntry(it) }\n    }\n\n    override operator fun get(key: K): V? =
implFindEntry(key)?.value\n\n    /**\n        * Returns the hash code value for this map.\n        * It is the same as
the hashCode of [entries] set.\n        */\n\n    override fun hashCode(): Int = entries.hashCode()\n\n    override fun
isEmpty(): Boolean = size == 0\n\n    override val size: Int get() = entries.size\n\n    /**\n        * Returns a read-only
[Set] of all keys in this map.\n        * Accessing this property first time creates a keys view from [entries].\n        *
All subsequent accesses just return the created instance.\n        */\n\n    override val keys: Set<K>\n        get() {\n            if (_keys == null) {\n                _keys = object : AbstractSet<K>() {\n                    override operator fun
contains(element: K): Boolean = containsKey(element)\n\n                    override operator fun iterator(): Iterator<K>
{\n                        val entryIterator = entries.iterator()\n                        return object : Iterator<K> {\n                            override fun hasNext(): Boolean = entryIterator.hasNext()\n                            override fun next(): K =

```

```

entryIterator.next().key\n          }\n          }\n          override val size: Int get() =
this@AbstractMap.size\n          }\n          }\n          return _keys!!\n          }\n          @kotlin.jvm.Volatile\nprivate var _keys: Set<K>? = null\n\n override fun toString(): String = entries.joinToString(", ", "\n", "\n") {
toString(it) }\n\n private fun toString(entry: Map.Entry<K, V>): String = toString(entry.key) + "=" +
toString(entry.value)\n\n private fun toString(o: Any?): String = if (o === this) "(this Map)" else o.toString()\n\n
/**\n * Returns a read-only [Collection] of all values in this map.\n * \n * Accessing this property first time
creates a values view from [entries].\n * All subsequent accesses just return the created instance.\n * \n
override val values: Collection<V>\n get() {\n if (_values == null) {\n _values = object :
AbstractCollection<V>() {\n override operator fun contains(element: @UnsafeVariance V): Boolean =
containsValue(element)\n\n override operator fun iterator(): Iterator<V> {\n val
entryIterator = entries.iterator()\n return object : Iterator<V> {\n override fun
hasNext(): Boolean = entryIterator.hasNext()\n\n override fun next(): V = entryIterator.next().value\n
}\n }\n\n override val size: Int get() = this@AbstractMap.size\n }\n
}\n return _values!!\n }\n\n @kotlin.jvm.Volatile\n private var _values: Collection<V>? = null\n\n
private fun implFindEntry(key: K): Map.Entry<K, V>? = entries.firstOrNull { it.key == key }\n\n internal
companion object {\n\n internal fun entryHashCode(e: Map.Entry<*, *>): Int = with(e) { (key?.hashCode() ?:
0) xor (value?.hashCode() ?: 0) }\n\n internal fun entryToString(e: Map.Entry<*, *>): String = with(e) {
"\$key=\$value" }\n\n internal fun entryEquals(e: Map.Entry<*, *>, other: Any?): Boolean {\n if (other !is
Map.Entry<*, *>) return false\n return e.key == other.key && e.value == other.value\n }\n
}\n\n"/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of
this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*\n*\npackage kotlin.collections\n\n/**\n * Provides a skeletal implementation of the read-only [Set] interface.\n * \n *
This class is intended to help implementing read-only sets so it doesn't support concurrent modification tracking.\n
*\n * @param E the type of elements contained in the set. The set is covariant in its element type.\n
*\n * @SinceKotlin("1.1")\n\npublic abstract class AbstractSet<out E> protected constructor() :
AbstractCollection<E>(), Set<E> {\n\n /**\n * Compares this set with other set instance with the unordered
structural equality.\n * \n * @return true, if [other] instance is a [Set] of the same size, all elements of which are
contained in this set.\n * \n override fun equals(other: Any?): Boolean {\n if (other === this) return true\n
if (other !is Set<*>) return false\n return setEquals(this, other)\n }\n\n /**\n * Returns the hash code
value for this set.\n * \n override fun hashCode(): Int = unorderedHashCode(this)\n\n internal companion
object {\n\n internal fun unorderedHashCode(c: Collection<*>): Int {\n var hashCode = 0\n for
(element in c) {\n hashCode += (element?.hashCode() ?: 0)\n }\n return hashCode\n
}\n\n internal fun setEquals(c: Set<*>, other: Set<*>): Boolean {\n if (c.size != other.size) return false\n
return c.containsAll(other)\n }\n }\n\n"/*\n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n *\n*\npackage kotlin.collections\n\n/**\n * Resizable-array implementation
of the deque data structure.\n * \n * The name deque is short for "double ended queue" and is usually pronounced
"deck".\n * \n * The collection provide methods for convenient access to the both ends.\n * It also implements
[MutableList] interface and supports efficient get/set operations by index.\n
*\n * @SinceKotlin("1.4")\n\n@WasExperimental(ExperimentalStdlibApi::class)\n\npublic class ArrayDeque<E> :
AbstractMutableList<E> {\n\n private var head: Int = 0\n private var elementData: Array<Any?>\n\n override
var size: Int = 0\n private set\n\n /**\n * Constructs an empty deque with specified [initialCapacity], or
throws [IllegalArgumentException] if [initialCapacity] is negative.\n * \n public constructor(initialCapacity:
Int) {\n elementData = when {\n initialCapacity == 0 -> emptyElementData\n initialCapacity > 0 -
> arrayOfNulls(initialCapacity)\n else -> throw IllegalArgumentException("Illegal Capacity:
\$initialCapacity")\n }\n }\n\n /**\n * Constructs an empty deque.\n * \n public constructor() {\n
elementData = emptyElementData\n }\n\n /**\n * Constructs a deque that contains the same elements as the
specified [elements] collection in the same order.\n * \n public constructor(elements: Collection<E>) {\n

```

```

elementData = elements.toArray()\n    size = elementData.size\n    if (elementData.isEmpty())
elementData = emptyElementData\n    }\n\n /**\n * Ensures that the capacity of this deque is at least equal to
the specified [minCapacity].\n * If the current capacity is less than the [minCapacity], a new backing
storage is allocated with greater capacity.\n * Otherwise, this method takes no action and simply returns.\n */\n
private fun ensureCapacity(minCapacity: Int) {\n    if (minCapacity < 0) throw IllegalStateException("Deque is
too big.") // overflow\n    if (minCapacity <= elementData.size) return\n    if (elementData ===
emptyElementData) {\n        elementData = arrayOfNulls(minCapacity.coerceAtLeast(defaultMinCapacity))\n
return\n    }\n\n    val newCapacity = newCapacity(elementData.size, minCapacity)\n
copyElements(newCapacity)\n    }\n\n /**\n * Creates a new array with the specified [newCapacity] size and
copies elements in the [elementData] array to it.\n */\n    private fun copyElements(newCapacity: Int) {\n    val
newElements = arrayOfNulls<Any?>(newCapacity)\n    elementData.copyInto(newElements, 0, head,
elementData.size)\n    elementData.copyInto(newElements, elementData.size - head, 0, head)\n    head = 0\n
elementData = newElements\n    }\n\n @kotlin.internal.InlineOnly\n    private inline fun
internalGet(internalIndex: Int): E {\n    @Suppress("UNCHECKED_CAST")\n    return
elementData[internalIndex] as E\n    }\n\n    private fun positiveMod(index: Int): Int = if (index >=
elementData.size) index - elementData.size else index\n    private fun negativeMod(index: Int): Int = if (index < 0)
index + elementData.size else index\n\n    @kotlin.internal.InlineOnly\n    private inline fun internalIndex(index:
Int): Int = positiveMod(head + index)\n\n    private fun incremented(index: Int): Int = if (index ==
elementData.lastIndex) 0 else index + 1\n\n    private fun decremented(index: Int): Int = if (index == 0)
elementData.lastIndex else index - 1\n\n    override fun isEmpty(): Boolean = size == 0\n\n /**\n * Returns the
first element, or throws [NoSuchElementException] if this deque is empty.\n */\n    public fun first(): E = if
(isEmpty()) throw NoSuchElementException("ArrayDeque is empty.") else internalGet(head)\n\n /**\n *
Returns the first element, or `null` if this deque is empty.\n */\n    public fun firstOrNull(): E? = if (isEmpty()) null
else internalGet(head)\n\n /**\n * Returns the last element, or throws [NoSuchElementException] if this deque
is empty.\n */\n    public fun last(): E = if (isEmpty()) throw NoSuchElementException("ArrayDeque is empty.")
else internalGet(internalIndex(lastIndex))\n\n /**\n * Returns the last element, or `null` if this deque is empty.\n
*/\n    public fun lastOrNull(): E? = if (isEmpty()) null else internalGet(internalIndex(lastIndex))\n\n /**\n *
Prepends the specified [element] to this deque.\n */\n    public fun addFirst(element: E) {\n
ensureCapacity(size + 1)\n\n    head = decremented(head)\n    elementData[head] = element\n    size += 1\n
}\n\n /**\n * Appends the specified [element] to this deque.\n */\n    public fun addLast(element: E) {\n
ensureCapacity(size + 1)\n\n    elementData[internalIndex(size)] = element\n    size += 1\n    }\n\n /**\n *
Removes the first element from this deque and returns that removed element, or throws [NoSuchElementException]
if this deque is empty.\n */\n    public fun removeFirst(): E {\n    if (isEmpty()) throw
NoSuchElementException("ArrayDeque is empty.")\n\n    val element = internalGet(head)\n
elementData[head] = null\n    head = incremented(head)\n    size -= 1\n    return element\n    }\n\n /**\n *
Removes the first element from this deque and returns that removed element, or returns `null` if this deque is
empty.\n */\n    public fun removeFirstOrNull(): E? = if (isEmpty()) null else removeFirst()\n\n /**\n *
Removes the last element from this deque and returns that removed element, or throws [NoSuchElementException]
if this deque is empty.\n */\n    public fun removeLast(): E {\n    if (isEmpty()) throw
NoSuchElementException("ArrayDeque is empty.")\n\n    val internalLastIndex = internalIndex(lastIndex)\n
val element = internalGet(internalLastIndex)\n    elementData[internalLastIndex] = null\n    size -= 1\n
return element\n    }\n\n /**\n * Removes the last element from this deque and returns that removed element, or
returns `null` if this deque is empty.\n */\n    public fun removeLastOrNull(): E? = if (isEmpty()) null else
removeLast()\n\n // MutableList, MutableCollection\n    public override fun add(element: E): Boolean {\n
addLast(element)\n    return true\n    }\n\n    public override fun add(index: Int, element: E) {\n
AbstractList.checkPositionIndex(index, size)\n\n    if (index == size) {\n        addLast(element)\n
return\n    } else if (index == 0) {\n        addFirst(element)\n        return\n    }\n\n    ensureCapacity(size
+ 1)\n\n    // Elements in circular array lay in 2 ways:\n    // 1. `head` is less than `tail`:    [#, #, e1, e2, e3,

```

```

#)\n // 2. `head` is greater than `tail`: [e3, #, #, #, e1, e2]\n // where head is the index of the first element
in the circular array,\n // and tail is the index following the last element.\n // At this point the
insertion index is not equal to head or tail.\n // Also the circular array can store at least one more element.\n
/>\n // Depending on where the given element must be inserted the preceding or the succeeding\n // elements
will be shifted to make room for the element to be inserted.\n // In case the preceding elements are
shifted:\n // * if the insertion index is greater than the head (regardless of circular array form)\n // ->
shift the preceding elements\n // * otherwise, the circular array has (2) form and the insertion index is less than
tail\n // -> shift all elements in the back of the array\n // -> shift preceding elements in the front of the
array\n // In case the succeeding elements are shifted:\n // * if the insertion index is less than the tail
(regardless of circular array form)\n // -> shift the succeeding elements\n // * otherwise, the circular
array has (2) form and the insertion index is greater than head\n // -> shift all elements in the front of the
array\n // -> shift succeeding elements in the back of the array\n\n val internalIndex =
internalIndex(index)\n\n if (index < (size + 1) shr 1) {\n // closer to the first element -> shift preceding
elements\n val decrementedInternalIndex = decremented(internalIndex)\n val decrementedHead =
decremented(head)\n\n if (decrementedInternalIndex >= head) {\n elementData[decrementedHead]
= elementData[head] // head can be zero\n elementData.copyInto(elementData, head, head + 1,
decrementedInternalIndex + 1)\n } else { // head > tail\n elementData.copyInto(elementData, head -
1, head, elementData.size) // head can't be zero\n elementData[elementData.size - 1] = elementData[0]\n
elementData.copyInto(elementData, 0, 1, decrementedInternalIndex + 1)\n }\n\n elementData[decrementedInternalIndex] = element\n head = decrementedHead\n } else {\n //
closer to the last element -> shift succeeding elements\n val tail = internalIndex(size)\n\n if
(internalIndex < tail) {\n elementData.copyInto(elementData, internalIndex + 1, internalIndex, tail)\n
} else { // head > tail\n elementData.copyInto(elementData, 1, 0, tail)\n elementData[0] =
elementData[elementData.size - 1]\n elementData.copyInto(elementData, internalIndex + 1, internalIndex,
elementData.size - 1)\n }\n\n elementData[internalIndex] = element\n }\n\n size += 1\n }\n\n private fun copyCollectionElements(internalIndex: Int, elements: Collection<E>) {\n val iterator =
elements.iterator()\n\n for (index in internalIndex until elementData.size) {\n if (!iterator.hasNext())
break\n elementData[index] = iterator.next()\n }\n\n for (index in 0 until head) {\n if
(!iterator.hasNext()) break\n elementData[index] = iterator.next()\n }\n\n size += elements.size\n
}\n\n public override fun addAll(elements: Collection<E>): Boolean {\n if (elements.isEmpty()) return false\n
ensureCapacity(this.size + elements.size)\n copyCollectionElements(internalIndex(size), elements)\n
return true\n }\n\n public override fun addAll(index: Int, elements: Collection<E>): Boolean {\n
AbstractList.checkPositionIndex(index, size)\n\n if (elements.isEmpty()) {\n return false\n } else if
(index == size) {\n return addAll(elements)\n }\n\n ensureCapacity(this.size + elements.size)\n\n
val tail = internalIndex(size)\n val internalIndex = internalIndex(index)\n val elementsSize =
elements.size\n\n if (index < (size + 1) shr 1) {\n // closer to the first element -> shift preceding
elements\n\n var shiftedHead = head - elementsSize\n\n if (internalIndex >= head) {\n if
(shiftedHead >= 0) {\n elementData.copyInto(elementData, shiftedHead, head, internalIndex)\n
} else { // head < tail, insertion leads to head >= tail\n shiftedHead += elementData.size\n val
elementsToShift = internalIndex - head\n val shiftToBack = elementData.size - shiftedHead\n\n
if (shiftToBack >= elementsToShift) {\n elementData.copyInto(elementData, shiftedHead, head,
internalIndex)\n } else {\n elementData.copyInto(elementData, shiftedHead, head, head +
shiftToBack)\n elementData.copyInto(elementData, 0, head + shiftToBack, internalIndex)\n
}\n }\n } else { // head > tail, internalIndex < tail\n elementData.copyInto(elementData,
shiftedHead, head, elementData.size)\n\n if (elementsSize >= internalIndex) {\n
elementData.copyInto(elementData, elementData.size - elementsSize, 0, internalIndex)\n } else {\n
elementData.copyInto(elementData, elementData.size - elementsSize, 0, elementsSize)\n
elementData.copyInto(elementData, 0, elementsSize, internalIndex)\n }\n }\n\n head =

```

```

shiftedHead\n        copyCollectionElements(negativeMod(internalIndex - elementsSize), elements)\n    } else
{\n        // closer to the last element -> shift succeeding elements\n        val shiftedInternalIndex =
internalIndex + elementsSize\n        if (internalIndex < tail) {\n            if (tail + elementsSize <=
elementData.size) {\n                elementData.copyInto(elementData, shiftedInternalIndex, internalIndex, tail)\n
            } else { // head < tail, insertion leads to head >= tail\n                if (shiftedInternalIndex >= elementData.size)
{\n                    elementData.copyInto(elementData, shiftedInternalIndex - elementData.size, internalIndex, tail)\n
                } else {\n                    val shiftToFront = tail + elementsSize - elementData.size\n
elementData.copyInto(elementData, 0, tail - shiftToFront, tail)\n                    elementData.copyInto(elementData,
shiftedInternalIndex, internalIndex, tail - shiftToFront)\n                }\n            } else { // head > tail,
internalIndex > head\n                elementData.copyInto(elementData, elementsSize, 0, tail)\n                if
(shiftedInternalIndex >= elementData.size) {\n                    elementData.copyInto(elementData, shiftedInternalIndex
- elementData.size, internalIndex, elementData.size)\n                } else {\n                    elementData.copyInto(elementData, 0, elementData.size - elementsSize, elementData.size)\n
                    elementData.copyInto(elementData, shiftedInternalIndex, internalIndex, elementData.size - elementsSize)\n
                }\n            }\n        }\n        copyCollectionElements(internalIndex, elements)\n    }\n\n    return true\n }\n\n public
override fun get(index: Int): E {\n    AbstractList.checkElementIndex(index, size)\n\n    return
internalGet(internalIndex(index))\n }\n\n public override fun set(index: Int, element: E): E {\n
AbstractList.checkElementIndex(index, size)\n    val internalIndex = internalIndex(index)\n    val oldElement
= internalGet(internalIndex)\n    elementData[internalIndex] = element\n\n    return oldElement\n }\n\n
public override fun contains(element: E): Boolean = indexOf(element) != -1\n\n public override fun
indexOf(element: E): Int {\n    val tail = internalIndex(size)\n\n    if (head < tail) {\n        for (index in head
until tail) {\n            if (element == elementData[index]) return index - head\n        }\n    } else if (head >=
tail) {\n        for (index in head until elementData.size) {\n            if (element == elementData[index]) return
index - head\n        }\n    }\n\n    for (index in 0 until tail) {\n        if (element == elementData[index]) return
index + elementData.size - head\n    }\n\n    return -1\n }\n\n public override fun
lastIndexOf(element: E): Int {\n    val tail = internalIndex(size)\n\n    if (head < tail) {\n        for (index in tail
- 1 downTo head) {\n            if (element == elementData[index]) return index - head\n        }\n    } else if
(head > tail) {\n        for (index in tail - 1 downTo 0) {\n            if (element == elementData[index]) return
index + elementData.size - head\n        }\n    }\n\n    for (index in elementData.lastIndex downTo head) {\n
if (element == elementData[index]) return index - head\n    }\n\n    return -1\n }\n\n public
override fun remove(element: E): Boolean {\n    val index = indexOf(element)\n    if (index == -1) return
false\n    removeAt(index)\n    return true\n }\n\n public override fun removeAt(index: Int): E {\n
AbstractList.checkElementIndex(index, size)\n\n    if (index == lastIndex) {\n        return removeLast()\n    }
else if (index == 0) {\n        return removeFirst()\n    }\n\n    val internalIndex = internalIndex(index)\n
val element = internalGet(internalIndex)\n\n    if (index < size shr 1) {\n        // closer to the first element ->
shift preceding elements\n        if (internalIndex >= head) {\n            elementData.copyInto(elementData, head
+ 1, head, internalIndex)\n        } else { // head > tail, internalIndex < head\n
elementData.copyInto(elementData, 1, 0, internalIndex)\n            elementData[0] = elementData[elementData.size
- 1]\n            elementData.copyInto(elementData, head + 1, head, elementData.size - 1)\n        }\n    }\n\n
elementData[head] = null\n    head = incremented(head)\n    } else {\n        // closer to the last element ->
shift succeeding elements\n        val internalLastIndex = internalIndex(lastIndex)\n\n        if (internalIndex <=
internalLastIndex) {\n            elementData.copyInto(elementData, internalIndex, internalIndex + 1,
internalLastIndex + 1)\n        } else { // head > tail, internalIndex > head\n
elementData.copyInto(elementData, internalIndex, internalIndex + 1, elementData.size)\n
elementData[elementData.size - 1] = elementData[0]\n            elementData.copyInto(elementData, 0, 1,
internalLastIndex + 1)\n        }\n    }\n\n    elementData[internalLastIndex] = null\n    }\n\n    size -= 1\n\n
return element\n }\n\n public override fun removeAll(elements: Collection<E>): Boolean = filterInPlace {
!elements.contains(it) }\n\n public override fun retainAll(elements: Collection<E>): Boolean = filterInPlace {

```

```

elements.contains(it) }\n\n private inline fun filterInPlace(predicate: (E) -> Boolean): Boolean {\n    if
(this.isEmpty() || elementData.isEmpty())\n        return false\n\n    val tail = internalIndex(size)\n    var
newTail = head\n    var modified = false\n\n    if (head < tail) {\n        for (index in head until tail) {\n
val element = elementData[index]\n            @Suppress(\"UNCHECKED_CAST\")\n            if
(predicate(element as E))\n                elementData[newTail++] = element\n            else\n                modified =
true\n        }\n        elementData.fill(null, newTail, tail)\n    } else {\n        for (index in head until
elementData.size) {\n            val element = elementData[index]\n            elementData[index] = null\n\n
@Suppress(\"UNCHECKED_CAST\")\n            if (predicate(element as E))\n                elementData[newTail++] = element\n            else\n                modified = true\n        }\n        newTail =
positiveMod(newTail)\n\n        for (index in 0 until tail) {\n            val element = elementData[index]\n
elementData[index] = null\n            @Suppress(\"UNCHECKED_CAST\")\n            if (predicate(element as
E)) {\n                elementData[newTail] = element\n                newTail = incremented(newTail)\n            }
else {\n                modified = true\n            }\n        }\n        if (modified)\n            size =
negativeMod(newTail - head)\n\n        return modified\n    }\n\n    public override fun clear() {\n        val tail =
internalIndex(size)\n        if (head < tail) {\n            elementData.fill(null, head, tail)\n        } else if (isEmpty())
{\n            elementData.fill(null, head, elementData.size)\n            elementData.fill(null, 0, tail)\n        }\n        head =
0\n        size = 0\n    }\n\n    @Suppress(\"NOTHING_TO_OVERRIDE\")\n    override fun <T> toArray(array:
Array<T>): Array<T> {\n        @Suppress(\"UNCHECKED_CAST\")\n        val dest = (if (array.size >= size) array
else arrayOfNulls(array, size)) as Array<Any?>\n\n        val tail = internalIndex(size)\n        if (head < tail) {\n
elementData.copyInto(dest, startIndex = head, endIndex = tail)\n        } else if (isEmpty()) {\n
elementData.copyInto(dest, destinationOffset = 0, startIndex = head, endIndex = elementData.size)\n
elementData.copyInto(dest, destinationOffset = elementData.size - head, startIndex = 0, endIndex = tail)\n    }\n
    if (dest.size > size) {\n        dest[size] = null // null-terminate\n    }\n\n    @Suppress(\"UNCHECKED_CAST\")\n    return dest as Array<T>\n }\n\n
@Suppress(\"NOTHING_TO_OVERRIDE\")\n    override fun toArray(): Array<Any?> {\n        return
toArray(arrayOfNulls<Any?>(size))\n    }\n\n    // for testing\n    internal fun <T> testToArray(array: Array<T>):
Array<T> = toArray(array)\n    internal fun testToArray(): Array<Any?> = toArray()\n\n    internal companion
object {\n        private val emptyElementData = emptyArray<Any?>()\n        private const val maxArraySize =
Int.MAX_VALUE - 8\n        private const val defaultMinCapacity = 10\n\n        internal fun
newCapacity(oldCapacity: Int, minCapacity: Int): Int {\n            // overflow-conscious\n            var newCapacity =
oldCapacity + (oldCapacity shr 1)\n            if (newCapacity - minCapacity < 0)\n                newCapacity =
minCapacity\n            if (newCapacity - maxArraySize > 0)\n                newCapacity = if (minCapacity >
maxArraySize) Int.MAX_VALUE else maxArraySize\n            return newCapacity\n        }\n    }\n\n    // For testing
only\n    internal fun internalStructure(structure: (head: Int, elements: Array<Any?>) -> Unit) {\n        val tail =
internalIndex(size)\n        val head = if (isEmpty() || head < tail) head else head - elementData.size\n
structure(head, toArray())\n    }\n}
\", \"/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n
*\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName(\"ArraysKt\")\n\npackage
kotlin.collections\n\nimport kotlin.contracts.*\n\n/**\n * Returns a single list of all elements from all arrays in the
given array.\n * @sample samples.collections.Arrays.Transformations.flattenArray\n */\npublic fun <T> Array<out
Array<out T>>.flatten(): List<T> {\n    val result = ArrayList<T>(sumOf { it.size })\n    for (element in this) {\n
result.addAll(element)\n    }\n    return result\n}\n\n/**\n * Returns a pair of lists, where\n * *first* list is built from
the first values of each pair from this array,\n * *second* list is built from the second values of each pair from this
array.\n * @sample samples.collections.Arrays.Transformations.unzipArray\n */\npublic fun <T, R> Array<out
Pair<T, R>>.unzip(): Pair<List<T>, List<R>> {\n    val listT = ArrayList<T>(size)\n    val listR =
ArrayList<R>(size)\n    for (pair in this) {\n        listT.add(pair.first)\n        listR.add(pair.second)\n    }\n    return
listT to listR\n}\n\n/**\n * Returns `true` if this nullable array is either null or empty.\n * @sample

```

```

samples.collections.Arrays.Usage.arrayIsNullOrEmpty\n
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline fun Array<*>?.isEmpty(): Boolean
{\n contract {\n returns(false) implies (this@isEmpty != null)\n }\n\n return this == null ||
this.isEmpty()\n}\n\n**\n * Returns this array if it's not empty\n * or the result of calling [defaultValue] function if
the array is empty.\n *\n * @sample samples.collections.Arrays.Usage.arrayIfEmpty\n
*\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\n@Suppress("UPPER_BOUND_CANNOT_BE_ARRAY\
")\npublic inline fun <C, R> C.ifEmpty(defaultValue: () -> R): R where C : Array<*>, C : R =\n if (isEmpty())
defaultValue() else
this\n\n\n@OptIn(ExperimentalUnsignedTypes::class)\n@SinceKotlin("1.3")\n@PublishedApi\n@kotlin.jvm.Jvm
Name("contentDeepEquals")\n@kotlin.js.JsName("contentDeepEqualsImpl")\ninternal fun <T> Array<out
T>?.contentDeepEqualsImpl(other: Array<out T>?): Boolean {\n if (this === other) return true\n if (this == null
|| other == null || this.size != other.size) return false\n\n for (i in indices) {\n val v1 = this[i]\n val v2 =
other[i]\n\n if (v1 === v2) {\n continue\n } else if (v1 == null || v2 == null) {\n return false\n
}\n\n when {\n v1 is Array<*> && v2 is Array<*> -> if (!v1.contentDeepEquals(v2)) return
false\n v1 is ByteArray && v2 is ByteArray -> if (!v1.contentEquals(v2)) return false\n v1 is
ShortArray && v2 is ShortArray -> if (!v1.contentEquals(v2)) return false\n v1 is IntArray && v2 is
IntArray -> if (!v1.contentEquals(v2)) return false\n v1 is LongArray && v2 is LongArray -> if
(!v1.contentEquals(v2)) return false\n v1 is FloatArray && v2 is FloatArray -> if (!v1.contentEquals(v2))
return false\n v1 is DoubleArray && v2 is DoubleArray -> if (!v1.contentEquals(v2)) return false\n
v1 is CharArray && v2 is CharArray -> if (!v1.contentEquals(v2)) return false\n v1 is BooleanArray &&
v2 is BooleanArray -> if (!v1.contentEquals(v2)) return false\n\n v1 is UByteArray && v2 is UByteArray
-> if (!v1.contentEquals(v2)) return false\n v1 is UShortArray && v2 is UShortArray -> if
(!v1.contentEquals(v2)) return false\n v1 is UIntArray && v2 is UIntArray -> if (!v1.contentEquals(v2))
return false\n v1 is ULongArray && v2 is ULongArray -> if (!v1.contentEquals(v2)) return false\n\n
else -> if (v1 != v2) return false\n }\n\n }\n return
true\n}\n\n@SinceKotlin("1.3")\n@PublishedApi\n@kotlin.jvm.JvmName("contentDeepToString")\n@kotlin.js.
JsName("contentDeepToStringImpl")\ninternal fun <T> Array<out T>?.contentDeepToStringImpl(): String {\n
if (this == null) return "null"\n val length = size.coerceAtMost((Int.MAX_VALUE - 2) / 5) * 5 + 2 // in order not
to overflow Int.MAX_VALUE\n return buildString(length) {\n contentDeepToStringInternal(this,
mutableListOf())\n }\n}\n\n@OptIn(ExperimentalUnsignedTypes::class)\nprivate fun <T> Array<out
T>.contentDeepToStringInternal(result: StringBuilder, processed: MutableList<Array<*>>) {\n if (this in
processed) {\n result.append("[...]")\n return\n }\n processed.add(this)\n result.append('[')\n\n for (i
in indices) {\n if (i != 0) {\n result.append(", ")\n }\n val element = this[i]\n when
(element) {\n null -> result.append("null")\n is Array<*> ->
element.contentDeepToStringInternal(result, processed)\n is ByteArray ->
result.append(element.contentToString())\n is ShortArray -> result.append(element.contentToString())\n
is IntArray -> result.append(element.contentToString())\n is LongArray ->
result.append(element.contentToString())\n is FloatArray -> result.append(element.contentToString())\n
is DoubleArray -> result.append(element.contentToString())\n is CharArray ->
result.append(element.contentToString())\n is BooleanArray -> result.append(element.contentToString())\n\n
is UByteArray -> result.append(element.contentToString())\n is UShortArray ->
result.append(element.contentToString())\n is UIntArray -> result.append(element.contentToString())\n
is ULongArray -> result.append(element.contentToString())\n\n else ->
result.append(element.toString())\n }\n }\n\n result.append(',')\n
processed.removeAt(processed.lastIndex)\n}"/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\npackage kotlin.collections\n\n** Returns true if the brittle contains optimization
is enabled. See KT-45438. *\n\ninternal expect fun brittleContainsOptimizationEnabled(): Boolean\n\n**\n *

```

Returns true if [brittleContainsOptimizationEnabled] is true and it's safe to convert this collection to a set without changing contains method behavior.

```

private fun <T> Collection<T>.safeToConvertToSet() =
    brittleContainsOptimizationEnabled() && size > 2 && this is ArrayList
    When
    [brittleContainsOptimizationEnabled] is true:
    - Converts this [Iterable] to a set if it is not a [Collection].
    - Converts this [Collection] to a set, when it's worth so and it doesn't change contains method behavior.
    - Otherwise returns this.
    When [brittleContainsOptimizationEnabled] is false:
    - Converts this [Iterable] to a list if it is not a [Collection].
    - Otherwise returns this.
internal fun <T>
Iterable<T>.convertToSetForSetOperationWith(source: Iterable<T>): Collection<T> =
    when (this) {
        is Set -> this
        is Collection -> when {
            source is Collection && source.size < 2 -> this
            else -> if (this.safeToConvertToSet()) toHashSet() else this
        }
        else -> if
            (brittleContainsOptimizationEnabled()) toHashSet() else toList()
    }
    When
    [brittleContainsOptimizationEnabled] is true:
    - Converts this [Iterable] to a set if it is not a [Collection].
    - Converts this [Collection] to a set, when it's worth so and it doesn't change contains method behavior.
    - Otherwise returns this.
    When [brittleContainsOptimizationEnabled] is false:
    - Converts this [Iterable] to a list if it is not a [Collection].
    - Otherwise returns this.
internal fun <T>
Iterable<T>.convertToSetForSetOperation(): Collection<T> =
    when (this) {
        is Set -> this
        is Collection -> if (this.safeToConvertToSet()) toHashSet() else this
        else -> if
            (brittleContainsOptimizationEnabled()) toHashSet() else toList()
    }
    Converts this sequence to a set if
    [brittleContainsOptimizationEnabled] is true, otherwise converts it to a list.
internal fun <T>
Sequence<T>.convertToSetForSetOperation(): Collection<T> =
    if (brittleContainsOptimizationEnabled())
        toHashSet() else toList()
    Converts this array to a set if [brittleContainsOptimizationEnabled] is true, otherwise converts it to a list.
internal fun <T> Array<T>.convertToSetForSetOperation(): Collection<T> =
    if (brittleContainsOptimizationEnabled()) toHashSet() else asList()
    Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
    Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
package kotlin.collections
Data class
representing a value from a collection or sequence, along with its index in that collection or sequence.
@property value the underlying value.
@property index the index of the value in the collection or sequence.
public data class IndexedValue<out T>(public val index: Int, public val value: T)
    Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.
    Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
@file:kotlin.jvm.JvmName("MapAccessorsKt")
package kotlin.collections
import
kotlin.reflect.KProperty
import kotlin.internal.Exact
    Returns the value of the property for the given object from this read-only map.
    @param thisRef the object for which the value is requested (not used).
    @param property the metadata for the property, used to get the name of property and lookup the value corresponding to this name in the map.
    @return the property value.
    @throws NoSuchElementException when the map doesn't contain value for the property name and doesn't provide an implicit default (see [withDefault]).
    @kotlin.internal.InlineOnly
    public inline operator fun <V, V1 : V> Map<in String, @Exact V>.getValue(thisRef: Any?, property: KProperty<*>): V1 =
        @Suppress("UNCHECKED_CAST")
        (getOrNull(property.name) as V1)
    Returns the value of the property for the given object from this mutable map.
    @param thisRef the object for which the value is requested (not used).
    @param property the metadata for the property, used to get the name of property and lookup the value corresponding to this name in the map.
    @return the property value.
    @throws NoSuchElementException when the map doesn't contain value for the property name and doesn't provide an implicit default (see [withDefault]).
    @kotlin.jvm.JvmName("getVar")
    @kotlin.internal.InlineOnly
    public inline operator fun <V, V1 : V> MutableMap<in String, out @Exact V>.getValue(thisRef: Any?, property: KProperty<*>): V1 =
        @Suppress("UNCHECKED_CAST")
        (getOrNull(property.name) as V1)
    Stores the value of the property for the given object in this mutable map.
    @param thisRef the object for which the value is requested (not used).
    @param property the metadata for the property, used to get the name of property and store

```

the value associated with that name in the map.\n * @param value the value to set.\n

```

*@\n@kotlin.internal.InlineOnly\npublic inline operator fun <V> MutableMap<in String, in V>.setValue(thisRef: Any?, property: KProperty<*>, value: V) {\n    this.put(property.name, value)\n}\n"/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("\nMapsKt\n")\n\npackage kotlin.collections\n\n/**\n * Returns the value for the given key, or the implicit default value for this map.\n * By default no implicit value is provided for maps and a [NoSuchElementException] is thrown.\n * To create a map with implicit default value use [withDefault] method.\n * @throws NoSuchElementException when the map doesn't contain a value for the specified key and no implicit default was provided for that map.\n
*\n@kotlin.jvm.JvmName("\ngetOrNull\n")\n@PublishedApi\ninternal fun <K, V> Map<K, V>.getOrNull(key: K): V {\n    if (this is MapWithDefault)\n        return this.getOrNull(key)\n    return getOrNull(key, { throw NoSuchElementException("\nKey $key is missing in the map.\n") })\n}\n\n/**\n * Returns a wrapper of this read-only map, having the implicit default value provided with the specified function [defaultValue].\n * This implicit default value is used when the original map doesn't contain a value for the key specified\n * and a value is obtained with [Map.getValue] function, for example when properties are delegated to the map.\n * When this map already has an implicit default value provided with a former call to [withDefault], it is being replaced by this call.\n
*\npublic fun <K, V> Map<K, V>.withDefault(defaultValue: (key: K) -> V): Map<K, V> =\n    when (this) {\n        is MapWithDefault -> this.map.withDefault(defaultValue)\n        else -> MapWithDefaultImpl(this, defaultValue)\n    }\n\n/**\n * Returns a wrapper of this mutable map, having the implicit default value provided with the specified function [defaultValue].\n * This implicit default value is used when the original map doesn't contain a value for the key specified\n * and a value is obtained with [Map.getValue] function, for example when properties are delegated to the map.\n * When this map already has an implicit default value provided with a former call to [withDefault], it is being replaced by this call.\n
*\n@kotlin.jvm.JvmName("\nwithDefaultMutable\n")\npublic fun <K, V> MutableMap<K, V>.withDefault(defaultValue: (key: K) -> V): MutableMap<K, V> =\n    when (this) {\n        is MutableMapWithDefault -> this.map.withDefault(defaultValue)\n        else -> MutableMapWithDefaultImpl(this, defaultValue)\n    }\n\nprivate interface MapWithDefault<K, out V> : Map<K, V> {\n    public val map: Map<K, V>\n    public fun getOrNull(key: K): V\n}\n\nprivate interface MutableMapWithDefault<K, V> : MutableMap<K, V>, MapWithDefault<K, V> {\n    public override val map: MutableMap<K, V>\n}\n\nprivate class MapWithDefaultImpl<K, out V>(public override val map: Map<K, V>, private val default: (key: K) -> V) : MapWithDefault<K, V> {\n    override fun equals(other: Any?): Boolean = map.equals(other)\n    override fun hashCode(): Int = map.hashCode()\n    override fun toString(): String = map.toString()\n    override val size: Int get() = map.size\n    override fun isEmpty(): Boolean = map.isEmpty()\n    override fun containsKey(key: K): Boolean = map.containsKey(key)\n    override fun containsValue(value: @UnsafeVariance V): Boolean = map.containsValue(value)\n    override fun get(key: K): V? = map.get(key)\n    override val keys: Set<K> get() = map.keys\n    override val values: Collection<V> get() = map.values\n    override val entries: Set<Map.Entry<K, V>> get() = map.entries\n}\n\nprivate class MutableMapWithDefaultImpl<K, V>(public override val map: MutableMap<K, V>, private val default: (key: K) -> V) : MutableMapWithDefault<K, V> {\n    override fun equals(other: Any?): Boolean = map.equals(other)\n    override fun hashCode(): Int = map.hashCode()\n    override fun toString(): String = map.toString()\n    override val size: Int get() = map.size\n    override fun isEmpty(): Boolean = map.isEmpty()\n    override fun containsKey(key: K): Boolean = map.containsKey(key)\n    override fun containsValue(value: @UnsafeVariance V): Boolean = map.containsValue(value)\n    override fun get(key: K): V? = map.get(key)\n    override val keys: MutableSet<K> get() = map.keys\n    override val values: MutableCollection<V> get() = map.values\n    override val entries: MutableSet<MutableMap.MutableEntry<K, V>> get() = map.entries\n    override fun put(key: K, value: V): V? = map.put(key, value)\n    override fun remove(key: K): V? = map.remove(key)\n    override fun putAll(from: Map<out K, V>) = map.putAll(from)\n    override fun clear() =

```

```

map.clear()\n\n override fun getOrImplicitDefault(key: K): V = map.getOrElseNullable(key, { default(key)
})\n}\n\n", "/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of
this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("CollectionsKt")\n\npackage
kotlin.collections\n\nimport kotlin.random.Random\n\n/**\n * Removes a single instance of the specified element
from this\n * collection, if it is present.\n * \n * Allows to overcome type-safety restriction of `remove` that requires
to pass an element of type `E`.\n * \n * @return `true` if the element has been successfully removed; `false` if it was
not present in the collection.\n */\n\n@kotlin.internal.InlineOnly\n\npublic inline fun <@kotlin.internal.OnlyInputTypes
T> MutableCollection<out T>.remove(element: T): Boolean =\n @Suppress("UNCHECKED_CAST") (this as
MutableCollection<T>).remove(element)\n\n/**\n * Removes all of this collection's elements that are also
contained in the specified collection.\n * \n * Allows to overcome type-safety restriction of `removeAll` that requires
to pass a collection of type `Collection<E>`.\n * \n * @return `true` if any of the specified elements was removed
from the collection, `false` if the collection was not modified.\n */\n\n@kotlin.internal.InlineOnly\n\npublic inline fun
<@kotlin.internal.OnlyInputTypes T> MutableCollection<out T>.removeAll(elements: Collection<T>): Boolean
=\n @Suppress("UNCHECKED_CAST") (this as MutableCollection<T>).removeAll(elements)\n\n/**\n * Retains only the
elements in this collection that are contained in the specified collection.\n * \n * Allows to
overcome type-safety restriction of `retainAll` that requires to pass a collection of type `Collection<E>`.\n * \n *
@return `true` if any element was removed from the collection, `false` if the collection was not modified.\n
*/\n\n@kotlin.internal.InlineOnly\n\npublic inline fun <@kotlin.internal.OnlyInputTypes T> MutableCollection<out
T>.retainAll(elements: Collection<T>): Boolean =\n @Suppress("UNCHECKED_CAST") (this as
MutableCollection<T>).retainAll(elements)\n\n/**\n * Adds the specified [element] to this mutable collection.\n
*/\n\n@kotlin.internal.InlineOnly\n\npublic inline operator fun <T> MutableCollection<in T>.plusAssign(element: T)
{\n this.add(element)\n}\n\n/**\n * Adds all elements of the given [elements] collection to this mutable
collection.\n */\n\n@kotlin.internal.InlineOnly\n\npublic inline operator fun <T> MutableCollection<in
T>.plusAssign(elements: Iterable<T>) {\n this.addAll(elements)\n}\n\n/**\n * Adds all elements of the given
[elements] array to this mutable collection.\n */\n\n@kotlin.internal.InlineOnly\n\npublic inline operator fun <T>
MutableCollection<in T>.plusAssign(elements: Array<T>) {\n this.addAll(elements)\n}\n\n/**\n * Adds all
elements of the given [elements] sequence to this mutable collection.\n */\n\n@kotlin.internal.InlineOnly\n\npublic
inline operator fun <T> MutableCollection<in T>.plusAssign(elements: Sequence<T>) {\n
this.addAll(elements)\n}\n\n/**\n * Removes a single instance of the specified [element] from this mutable
collection.\n */\n\n@kotlin.internal.InlineOnly\n\npublic inline operator fun <T> MutableCollection<in
T>.minusAssign(element: T) {\n this.remove(element)\n}\n\n/**\n * Removes all elements contained in the given
[elements] collection from this mutable collection.\n */\n\n@kotlin.internal.InlineOnly\n\npublic inline operator fun
<T> MutableCollection<in T>.minusAssign(elements: Iterable<T>) {\n this.removeAll(elements)\n}\n\n/**\n *
Removes all elements contained in the given [elements] array from this mutable collection.\n
*/\n\n@kotlin.internal.InlineOnly\n\npublic inline operator fun <T> MutableCollection<in T>.minusAssign(elements:
Array<T>) {\n this.removeAll(elements)\n}\n\n/**\n * Removes all elements contained in the given [elements]
sequence from this mutable collection.\n */\n\n@kotlin.internal.InlineOnly\n\npublic inline operator fun <T>
MutableCollection<in T>.minusAssign(elements: Sequence<T>) {\n this.removeAll(elements)\n}\n\n/**\n * Adds
all elements of the given [elements] collection to this [MutableCollection].\n */\n\npublic fun <T>
MutableCollection<in T>.addAll(elements: Iterable<T>): Boolean {\n when (elements) {\n is Collection ->
return addAll(elements)\n else -> {\n var result: Boolean = false\n for (item in elements)\n
if (add(item)) result = true\n return result\n } }\n}\n\n/**\n * Adds all elements of the given
[elements] sequence to this [MutableCollection].\n */\n\npublic fun <T> MutableCollection<in T>.addAll(elements:
Sequence<T>): Boolean {\n var result: Boolean = false\n for (item in elements) {\n if (add(item)) result =
true\n }\n return result\n}\n\n/**\n * Adds all elements of the given [elements] array to this
[MutableCollection].\n */\n\npublic fun <T> MutableCollection<in T>.addAll(elements: Array<out T>): Boolean {\n
return addAll(elements.asList())\n}\n\n/**\n * Removes all elements from this [MutableCollection] that are also

```

```

contained in the given [elements] collection.\n */\npublic fun <T> MutableCollection<in T>.removeAll(elements:
Iterable<T>): Boolean {\n    return removeAll(elements.convertToSetForSetOperationWith(this))\n}\n\n/**\n *
Removes all elements from this [MutableCollection] that are also contained in the given [elements] sequence.\n
*/\npublic fun <T> MutableCollection<in T>.removeAll(elements: Sequence<T>): Boolean {\n    val set =
elements.convertToSetForSetOperation()\n    return set.isNotEmpty() && removeAll(set)\n}\n\n/**\n * Removes all
elements from this [MutableCollection] that are also contained in the given [elements] array.\n */\npublic fun <T>
MutableCollection<in T>.removeAll(elements: Array<out T>): Boolean {\n    return elements.isNotEmpty() &&
removeAll(elements.convertToSetForSetOperation())\n}\n\n/**\n * Retains only elements of this
[MutableCollection] that are contained in the given [elements] collection.\n */\npublic fun <T>
MutableCollection<in T>.retainAll(elements: Iterable<T>): Boolean {\n    return
retainAll(elements.convertToSetForSetOperationWith(this))\n}\n\n/**\n * Retains only elements of this
[MutableCollection] that are contained in the given [elements] array.\n */\npublic fun <T> MutableCollection<in
T>.retainAll(elements: Array<out T>): Boolean {\n    if (elements.isNotEmpty())\n        return
retainAll(elements.convertToSetForSetOperation())\n    else\n        return retainNothing()\n}\n\n/**\n * Retains only
elements of this [MutableCollection] that are contained in the given [elements] sequence.\n */\npublic fun <T>
MutableCollection<in T>.retainAll(elements: Sequence<T>): Boolean {\n    val set =
elements.convertToSetForSetOperation()\n    if (set.isNotEmpty())\n        return retainAll(set)\n    else\n        return
retainNothing()\n}\n\nprivate fun MutableCollection<*>.retainNothing(): Boolean {\n    val result = isEmpty()\n    clear()\n    return result\}\n\n\n/**\n * Removes all elements from this [MutableIterable] that match the given
[predicate].\n */\n * @return `true` if any element was removed from this collection, or `false` when no elements
were removed and collection was not modified.\n */\npublic fun <T> MutableIterable<T>.removeAll(predicate: (T)
-> Boolean): Boolean = filterInPlace(predicate, true)\n\n\n/**\n * Retains only elements of this [MutableIterable] that
match the given [predicate].\n */\n * @return `true` if any element was removed from this collection, or `false` when
all elements were retained and collection was not modified.\n */\npublic fun <T>
MutableIterable<T>.retainAll(predicate: (T) -> Boolean): Boolean = filterInPlace(predicate, false)\n\nprivate fun
<T> MutableIterable<T>.filterInPlace(predicate: (T) -> Boolean, predicateResultToRemove: Boolean): Boolean {\n
    var result = false\n    with(iterator()) {\n        while (hasNext())\n            if (predicate(next()) ==
predicateResultToRemove) {\n                remove()\n                result = true\n            }\n    }\n    return
result\}\n\n\n/**\n * Removes the element at the specified [index] from this list.\n * In Kotlin one should use the
[MutableList.removeAt] function instead.\n */\n@Deprecated("Use removeAt(index) instead.")
ReplaceWith("removeAt(index)"), level = DeprecationLevel.ERROR)\n@kotlin.internal.InlineOnly\npublic inline
fun <T> MutableList<T>.remove(index: Int): T = removeAt(index)\n\n\n/**\n * Removes the first element from this
mutable list and returns that removed element, or throws [NoSuchElementException] if this list is empty.\n
*/\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun <T>
MutableList<T>.removeFirst(): T = if (isEmpty()) throw NoSuchElementException("List is empty.") else
removeAt(0)\n\n\n/**\n * Removes the first element from this mutable list and returns that removed element, or
returns `null` if this list is empty.\n
*/\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun <T>
MutableList<T>.removeFirstOrNull(): T? = if (isEmpty()) null else removeAt(0)\n\n\n/**\n * Removes the last
element from this mutable list and returns that removed element, or throws [NoSuchElementException] if this list is
empty.\n */\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun <T>
MutableList<T>.removeLast(): T = if (isEmpty()) throw NoSuchElementException("List is empty.") else
removeAt(lastIndex)\n\n\n/**\n * Removes the last element from this mutable list and returns that removed element,
or returns `null` if this list is empty.\n
*/\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun <T>
MutableList<T>.removeLastOrNull(): T? = if (isEmpty()) null else removeAt(lastIndex)\n\n\n/**\n * Removes all
elements from this [MutableList] that match the given [predicate].\n */\n * @return `true` if any element was
removed from this collection, or `false` when no elements were removed and collection was not modified.\n

```

```

*^public fun <T> MutableList<T>.removeAll(predicate: (T) -> Boolean): Boolean = filterInPlace(predicate,
true)\n\n/**\n * Retains only elements of this [MutableList] that match the given [predicate].\n *\n * @return `true`
if any element was removed from this collection, or `false` when all elements were retained and collection was not
modified.\n */\npublic fun <T> MutableList<T>.retainAll(predicate: (T) -> Boolean): Boolean =
filterInPlace(predicate, false)\n\nprivate fun <T> MutableList<T>.filterInPlace(predicate: (T) -> Boolean,
predicateResultToRemove: Boolean): Boolean {\n    if (this !is RandomAccess)\n        return (this as
MutableIterable<T>).filterInPlace(predicate, predicateResultToRemove)\n\n    var writeIndex: Int = 0\n    for
(readIndex in 0..lastIndex) {\n        val element = this[readIndex]\n        if (predicate(element) ==
predicateResultToRemove)\n            continue\n\n        if (writeIndex != readIndex)\n            this[writeIndex] =
element\n\n        writeIndex++\n    }\n    if (writeIndex < size) {\n        for (removeIndex in lastIndex downTo
writeIndex)\n            removeAt(removeIndex)\n\n        return true\n    } else {\n        return false\n    }\n}\n"/\n\n *
Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is
governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("CollectionsKt")\n\npackage
kotlin.collections\n\nprivate open class ReversedListReadOnly<out T>(private val delegate: List<T>) :
AbstractList<T>() {\n    override val size: Int get() = delegate.size\n    override fun get(index: Int): T =
delegate[reverseElementIndex(index)]\n}\n\nprivate class ReversedList<T>(private val delegate: MutableList<T>) :
AbstractMutableList<T>() {\n    override val size: Int get() = delegate.size\n    override fun get(index: Int): T =
delegate[reverseElementIndex(index)]\n\n    override fun clear() = delegate.clear()\n    override fun removeAt(index:
Int): T = delegate.removeAt(reverseElementIndex(index))\n\n    override fun set(index: Int, element: T): T =
delegate.set(reverseElementIndex(index), element)\n\n    override fun add(index: Int, element: T) {\n        delegate.add(reversePositionIndex(index), element)\n    }\n}\n\nprivate fun List<*>.reverseElementIndex(index:
Int) =\n    if (index in 0..lastIndex) lastIndex - index else throw IndexOutOfBoundsException("Element index
$index must be in range [0..lastIndex].")\n\nprivate fun List<*>.reversePositionIndex(index: Int) =\n    if (index
in 0..size) size - index else throw IndexOutOfBoundsException("Position index $index must be in range
[0..size].")\n\n/**\n * Returns a reversed read-only view of the original List.\n * All changes made in the
original list will be reflected in the reversed one.\n * @sample samples.collections.ReversedViews.asReversedList\n
*/\n\npublic fun <T> List<T>.asReversed(): List<T> = ReversedListReadOnly(this)\n\n/**\n * Returns a reversed
mutable view of the original mutable List.\n * All changes made in the original list will be reflected in the reversed
one and vice versa.\n * @sample samples.collections.ReversedViews.asReversedMutableList\n
*/\n\n@kotlin.jvm.JvmName("asReversedMutable")\n\npublic fun <T> MutableList<T>.asReversed():
MutableList<T> = ReversedList(this)\n\n"/\n\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n
*/\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("SequencesKt")\n@file:OptIn(ExperimentalTypeInference::class)\n\npackage kotlin.sequences\n\nimport kotlin.coroutines.*\nimport
kotlin.coroutines.intrinsics.*\nimport kotlin.experimental.ExperimentalTypeInference\n\n/**\n * Builds a
[Sequence] lazily yielding values one by one.\n *\n * @see kotlin.sequences.generateSequence\n *\n * @sample
samples.collections.Sequences.Building.buildSequenceYieldAll\n * @sample
samples.collections.Sequences.Building.buildFibonacciSequence\n */\n\n@SinceKotlin("1.3")\n\npublic fun <T>
sequence(@BuilderInference block: suspend SequenceScope<T>().() -> Unit): Sequence<T> = Sequence {
iterator(block) }\n\n@SinceKotlin("1.3")\n\n@Deprecated("Use 'sequence { }' function instead.",
ReplaceWith("sequence(builderAction)"), level =
DeprecationLevel.ERROR)\n\n@kotlin.internal.InlineOnly\n\npublic inline fun <T> buildSequence(@BuilderInference
noinline builderAction: suspend SequenceScope<T>().() -> Unit): Sequence<T> = Sequence { iterator(builderAction)
}\n\n/**\n * Builds an [Iterator] lazily yielding values one by one.\n *\n * @sample
samples.collections.Sequences.Building.buildIterator\n * @sample samples.collections.Iterables.Building.iterable\n
*/\n\n@SinceKotlin("1.3")\n\npublic fun <T> iterator(@BuilderInference block: suspend SequenceScope<T>().() ->

```

```

Unit): Iterator<T> { \n    val iterator = SequenceBuilderIterator<T>()\n    iterator.nextStep =
block.createCoroutineUnintercepted(receiver = iterator, completion = iterator)\n    return
iterator\n}\n\n@SinceKotlin("1.3")\n@Deprecated("Use 'iterator { }' function instead.",
ReplaceWith("iterator(builderAction)"), level = DeprecationLevel.ERROR)\n@kotlin.internal.InlineOnly\npublic
inline fun <T> buildIterator(@BuilderInference noline builderAction: suspend SequenceScope<T>().-> Unit):
Iterator<T> = iterator(builderAction)\n\n/**\n * The scope for yielding values of a [Sequence] or an [Iterator],
provides [yield] and [yieldAll] suspension functions.\n * \n * @see sequence\n * @see iterator\n * \n * @sample
samples.collections.Sequences.Building.buildSequenceYieldAll\n * @sample
samples.collections.Sequences.Building.buildFibonacciSequence\n
*\n*\n@RestrictsSuspension\n@SinceKotlin("1.3")\npublic abstract class SequenceScope<in T> internal
constructor() {\n    /**\n     * Yields a value to the [Iterator] being built and suspends\n     * until the next value is
requested.\n     * \n     * @sample samples.collections.Sequences.Building.buildSequenceYieldAll\n     * @sample
samples.collections.Sequences.Building.buildFibonacciSequence\n     * \n     * public abstract suspend fun yield(value:
T)\n     * \n     * /**\n     * Yields all values from the `iterator` to the [Iterator] being built\n     * and suspends until all these
values are iterated and the next one is requested.\n     * \n     * The sequence of values returned by the given iterator
can be potentially infinite.\n     * \n     * @sample samples.collections.Sequences.Building.buildSequenceYieldAll\n
*\n     * public abstract suspend fun yieldAll(iterator: Iterator<T>)\n     * \n     * /**\n     * Yields a collections of values to
the [Iterator] being built\n     * and suspends until all these values are iterated and the next one is requested.\n     * \n
     * \n     * @sample samples.collections.Sequences.Building.buildSequenceYieldAll\n     * \n     * public suspend fun
yieldAll(elements: Iterable<T>) {\n     *     if (elements is Collection && elements.isEmpty()) return\n     *     return
yieldAll(elements.iterator())\n     * }\n     * \n     * /**\n     * Yields potentially infinite sequence of values to the [Iterator]
being built\n     * and suspends until all these values are iterated and the next one is requested.\n     * \n     * The
sequence can be potentially infinite.\n     * \n     * @sample
samples.collections.Sequences.Building.buildSequenceYieldAll\n     * \n     * public suspend fun yieldAll(sequence:
Sequence<T>) = yieldAll(sequence.iterator())\n}\n\n@Deprecated("Use SequenceScope class instead.",
ReplaceWith("SequenceScope<T>"), level = DeprecationLevel.ERROR)\npublic typealias SequenceBuilder<T> =
SequenceScope<T>\n\nprivate typealias State = Int\n\nprivate const val State_NotReady: State = 0\n\nprivate const
val State_ManyNotReady: State = 1\n\nprivate const val State_ManyReady: State = 2\n\nprivate const val State_Ready:
State = 3\n\nprivate const val State_Done: State = 4\n\nprivate const val State_Failed: State = 5\n\nprivate class
SequenceBuilderIterator<T> : SequenceScope<T>(), Iterator<T>, Continuation<Unit> {\n    private var state =
State_NotReady\n    private var nextValue: T? = null\n    private var nextIterator: Iterator<T>? = null\n    var
nextStep: Continuation<Unit>? = null\n\n    override fun hasNext(): Boolean {\n        while (true) {\n            when
(state) {\n                State_NotReady -> {}\n                State_ManyNotReady ->\n                    if
(nextIterator!!.hasNext()) {\n                        state = State_ManyReady\n                        return true\n                    }
else {\n                        nextIterator = null\n                    }\n                State_Done -> return false\n                State_Ready, State_ManyReady -> return true\n            }\n            else -> throw exceptionalState()\n        }\n        state
= State_Failed\n        val step = nextStep!!\n        nextStep = null\n        step.resume(Unit)\n    }\n}\n\noverride fun next(): T {\n    when (state) {\n        State_NotReady, State_ManyNotReady -> return
nextNotReady()\n        State_ManyReady -> {\n            state = State_ManyNotReady\n            return
nextIterator!!.next()\n        }\n        State_Ready -> {\n            state = State_NotReady\n            @Suppress("UNCHECKED_CAST")\n            val result = nextValue as T\n            nextValue = null\n            return result\n        }\n        else -> throw exceptionalState()\n    }\n}\n\nprivate fun nextNotReady(): T
{\n    if (!hasNext()) throw NoSuchElementException() else return next()\n}\n\nprivate fun
exceptionalState(): Throwable = when (state) {\n    State_Done -> NoSuchElementException()\n    State_Failed
-> IllegalStateException("Iterator has failed.")\n    else -> IllegalStateException("Unexpected state of the
iterator: $state")\n}\n\n\n    override suspend fun yield(value: T) {\n        nextValue = value\n        state =
State_Ready\n        return suspendCoroutineUninterceptedOrReturn { c ->\n            nextStep = c\n            COROUTINE_SUSPENDED\n        }\n    }\n}\n\n    override suspend fun yieldAll(iterator: Iterator<T>) {\n        if

```

```

(iterator.hasNext()) return\n    nextIterator = iterator\n    state = State_ManyReady\n    return
suspendCoroutineUninterceptedOrReturn { c ->\n    nextStep = c\n    COROUTINE_SUSPENDED\n}
}\n\n // Completion continuation implementation\n override fun resumeWith(result: Result<Unit>) {\n
result.getOrThrow() // just rethrow exception if it is there\n    state = State_Done\n    }\n\n override val context:
CoroutineContext\n    get() = EmptyCoroutineContext\n}\n"/**\n * Copyright 2010-2018 JetBrains s.r.o. and
Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.collections\n\ninternal fun
checkWindowSizeStep(size: Int, step: Int) {\n    require(size > 0 && step > 0) {\n        if (size != step)\n            \"Both size $size and step $step must be greater than zero.\"\n        else\n            \"size $size must be greater than
zero.\"\n    }\n}\n\ninternal fun <T> Sequence<T>.windowedSequence(size: Int, step: Int, partialWindows: Boolean,
reuseBuffer: Boolean): Sequence<List<T>> {\n    checkWindowSizeStep(size, step)\n    return Sequence {\n
windowedIterator(iterator(), size, step, partialWindows, reuseBuffer) }\n}\n\ninternal fun <T>
windowedIterator(iterator: Iterator<T>, size: Int, step: Int, partialWindows: Boolean, reuseBuffer: Boolean):
Iterator<List<T>> {\n    if (!iterator.hasNext()) return EmptyIterator\n    return iterator<List<T>> {\n        val
bufferInitialCapacity = size.coerceAtMost(1024)\n        val gap = step - size\n        if (gap >= 0) {\n            var buffer
= ArrayList<T>(bufferInitialCapacity)\n            var skip = 0\n            for (e in iterator) {\n                if (skip > 0) {\n
skip -= 1; continue }\n                buffer.add(e)\n                if (buffer.size == size) {\n                    yield(buffer)\n
                    if (reuseBuffer) buffer.clear() else buffer = ArrayList(size)\n                    skip = gap\n                }\n            }\n            if (buffer.isNotEmpty()) {\n                if (partialWindows || buffer.size == size) yield(buffer)\n            }\n        }\n        else {\n            var buffer = RingBuffer<T>(bufferInitialCapacity)\n            for (e in iterator) {\n
buffer.add(e)\n                if (buffer.isFull()) {\n                    if (buffer.size < size) {\n                        buffer =
buffer.expanded(maxCapacity = size); continue }\n                    yield(if (reuseBuffer) buffer else
ArrayList(buffer))\n                    buffer.removeFirst(step)\n                }\n            }\n            if (partialWindows) {\n
while (buffer.size > step) {\n                yield(if (reuseBuffer) buffer else ArrayList(buffer))\n                buffer.removeFirst(step)\n            }\n            if (buffer.isNotEmpty()) yield(buffer)\n        }\n    }\n}\n\ninternal class MovingSubList<out E>(private val list: List<E>) : AbstractList<E>(), RandomAccess {\n
private var fromIndex: Int = 0\n    private var _size: Int = 0\n\n    fun move(fromIndex: Int, toIndex: Int) {\n
checkRangeIndexes(fromIndex, toIndex, list.size)\n        this.fromIndex = fromIndex\n        this._size = toIndex -
fromIndex\n    }\n\n    override fun get(index: Int): E {\n        checkElementIndex(index, _size)\n        return
list[fromIndex + index]\n    }\n\n    override val size: Int get() = _size\n}\n\n/**\n * Provides ring buffer
implementation.\n * Buffer overflow is not allowed so [add] doesn't overwrite tail but raises an exception.\n
*/\n\nprivate class RingBuffer<T>(private val buffer: Array<Any?>, filledSize: Int) : AbstractList<T>(),
RandomAccess {\n    init {\n        require(filledSize >= 0) { \"ring buffer filled size should not be negative but it is
$filledSize\" }\n        require(filledSize <= buffer.size) { \"ring buffer filled size: $filledSize cannot be larger than
the buffer size: ${buffer.size}\" }\n    }\n\n    constructor(capacity: Int) : this(arrayOfNulls<Any?>(capacity), 0)\n\n    private val capacity = buffer.size\n    private var startIndex: Int = 0\n    override var size: Int = filledSize\n\n    private set\n\n    override fun get(index: Int): T {\n        checkElementIndex(index, size)\n
@Suppress(\"UNCHECKED_CAST\")\n        return buffer[startIndex.forward(index)] as T\n    }\n\n    fun isFull() =
size == capacity\n\n    override fun iterator(): Iterator<T> = object : AbstractIterator<T>() {\n        private var count
= size\n        private var index = startIndex\n\n        override fun computeNext() {\n            if (count == 0) {\n
done()\n            } else {\n                @Suppress(\"UNCHECKED_CAST\")\n                setNext(buffer[index] as
T)\n                index = index.forward(1)\n                count--\n            }\n        }\n    }\n}\n\n@Suppress(\"UNCHECKED_CAST\")\n override fun <T> toArray(array: Array<T>): Array<T> {\n    val
result: Array<T?> =\n        if (array.size < this.size) array.copyOf(this.size) else array as Array<T?>\n\n    val
size = this.size\n\n    var widx = 0\n    var idx = startIndex\n\n    while (widx < size && idx < capacity) {\n
result[widx] = buffer[idx] as T\n        widx++\n        idx++\n    }\n\n    idx = 0\n    while (widx <
size) {\n        result[widx] = buffer[idx] as T\n        widx++\n        idx++\n    }\n\n    if (result.size >
this.size) result[this.size] = null\n\n    return result as Array<T>\n }\n\n override fun toArray(): Array<Any?>

```


Function<R>, kind: InvocationKind = InvocationKind.UNKNOWN): CallsInPlace\n\n\n/**\n * Specifies how many times a function invokes its function parameter in place.\n * See [ContractBuilder.callsInPlace] for the details of the call-in-place function contract.\n

```

*\n@ContractsDsl\n@ExperimentalContracts\n@SinceKotlin("1.3")\npublic enum class InvocationKind {\n
/**\n * A function parameter will be invoked one time or not invoked at all.\n * // @sample
samples.contracts.callsInPlaceAtMostOnceContract\n @ContractsDsl AT_MOST_ONCE,\n /**\n * A function parameter will be invoked one or more times.\n *\n * // @sample
samples.contracts.callsInPlaceAtLeastOnceContract\n @ContractsDsl AT_LEAST_ONCE,\n /**\n * A function parameter will be invoked exactly one time.\n *\n * // @sample
samples.contracts.callsInPlaceExactlyOnceContract\n @ContractsDsl EXACTLY_ONCE,\n /**\n * A function parameter is called in place, but it's unknown how many times it can be called.\n *\n * // @sample
samples.contracts.callsInPlaceUnknownContract\n @ContractsDsl UNKNOWN\n}\n\n/**\n * Specifies the contract of a function.\n *\n * The contract description must be at the beginning of a function and have at least one effect.\n *\n * Only the top-level functions can have a contract for now.\n *\n * @param builder the lambda where the contract of a function is described with the help of the [ContractBuilder] members.\n *\n */\n * @sample
samples.contracts.returnsContract\n * @sample samples.contracts.returnsTrueContract\n * @sample
samples.contracts.returnsFalseContract\n * @sample samples.contracts.returnsNullContract\n * @sample
samples.contracts.returnsNotNullContract\n * @sample samples.contracts.callsInPlaceAtMostOnceContract\n * @sample
samples.contracts.callsInPlaceAtLeastOnceContract\n * @sample
samples.contracts.callsInPlaceExactlyOnceContract\n * @sample
samples.contracts.callsInPlaceUnknownContract\n*\n@ContractsDsl\n@ExperimentalContracts\n@InlineOnly\n@SinceKotlin("1.3")\n@Suppress("UNUSED_PARAMETER")\npublic inline fun contract(builder: ContractBuilder.() -> Unit) { } \n", "/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\npackage kotlin.coroutines\n\n/**\n * Marks coroutine context element that intercepts coroutine continuations.\n * The coroutines framework uses [ContinuationInterceptor.Key] to retrieve the interceptor and\n * intercepts all coroutine continuations with [interceptContinuation] invocations.\n *\n * [ContinuationInterceptor] behaves like a [polymorphic element][AbstractCoroutineContextKey], meaning that\n * its implementation delegates [get][CoroutineContext.Element.get] and [minusKey][CoroutineContext.Element.minusKey]\n * to [getPolymorphicElement] and [minusPolymorphicKey] respectively.\n * [ContinuationInterceptor] subtypes can be extracted from the coroutine context using either [ContinuationInterceptor.Key]\n * or subtype key if it extends [AbstractCoroutineContextKey].\n */\n@SinceKotlin("1.3")\npublic interface ContinuationInterceptor : CoroutineContext.Element {\n /**\n * The key that defines *the* context interceptor.\n * companion object Key :
CoroutineContext.Key<ContinuationInterceptor>\n /**\n * Returns continuation that wraps the original [continuation], thus intercepting all resumptions.\n * This function is invoked by coroutines framework when needed and the resulting continuations are\n * cached internally per each instance of the original [continuation].\n *\n * This function may simply return original [continuation] if it does not want to intercept this particular continuation.\n *\n * When the original [continuation] completes, coroutine framework invokes [releaseInterceptedContinuation]\n * with the resulting continuation if it was intercepted, that is if `interceptContinuation` had previously\n * returned a different continuation instance.\n */\n public fun <T> interceptContinuation(continuation: Continuation<T>): Continuation<T>\n /**\n * Invoked for the continuation instance returned by [interceptContinuation] when the original\n * continuation completes and will not be used anymore. This function is invoked only if [interceptContinuation]\n * had returned a different continuation instance from the one it was invoked with.\n *\n * Default implementation does nothing.\n *\n * @param continuation Continuation instance returned by this interceptor's [interceptContinuation] invocation.\n */\n public fun releaseInterceptedContinuation(continuation: Continuation<*>) {\n /* do nothing by default
*\n */\n } \n\n public override operator fun <E : CoroutineContext.Element> get(key: CoroutineContext.Key<E>):

```

```

E? {\n    // getPolymorphicKey specialized for ContinuationInterceptor key\n
@OptIn(ExperimentalStdlibApi::class)\n    if (key is AbstractCoroutineContextKey<*, *>) {\n
@Suppress("UNCHECKED_CAST")\n        return if (key.isSubKey(this.key)) key.tryCast(this) as? E else
null\n    }\n    @Suppress("UNCHECKED_CAST")\n        return if (ContinuationInterceptor === key) this as
E else null\n    }\n\n    public override fun minusKey(key: CoroutineContext.Key<*>): CoroutineContext {\n
// minusPolymorphicKey specialized for ContinuationInterceptor key\n
@OptIn(ExperimentalStdlibApi::class)\n    if (key is AbstractCoroutineContextKey<*, *>) {\n        return if
(key.isSubKey(this.key) && key.tryCast(this) != null) EmptyCoroutineContext else this\n    }\n    return if
(ContinuationInterceptor === key) EmptyCoroutineContext else this\n    }\n}\n"/*\n * Copyright 2010-2018
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.coroutines\n\n/**\n *
Persistent context for the coroutine. It is an indexed set of [Element] instances.\n * An indexed set is a mix between
a set and a map.\n * Every element in this set has a unique [Key].\n */\n@SinceKotlin("1.3")\npublic interface
CoroutineContext {\n    /**\n     * Returns the element with the given [key] from this context or `null`.\n     */\n    public operator fun <E : Element> get(key: Key<E>): E?\n\n    /**\n     * Accumulates entries of this context
starting with [initial] value and applying [operation]\n     * from left to right to current accumulator value and each
element of this context.\n     */\n    public fun <R> fold(initial: R, operation: (R, Element) -> R): R\n\n    /**\n     *
Returns a context containing elements from this context and elements from other [context].\n     * The elements
from this context with the same key as in the other one are dropped.\n     */\n    public operator fun plus(context:
CoroutineContext): CoroutineContext =\n        if (context === EmptyCoroutineContext) this else // fast path -- avoid
lambda creation\n            context.fold(this) { acc, element ->\n                val removed =
acc.minusKey(element.key)\n                if (removed === EmptyCoroutineContext) element else {\n                    //
make sure interceptor is always last in the context (and thus is fast to get when present)\n                    val interceptor
= removed[ContinuationInterceptor]\n                    if (interceptor == null) CombinedContext(removed, element) else
{\n                        val left = removed.minusKey(ContinuationInterceptor)\n                        if (left ===
EmptyCoroutineContext) CombinedContext(element, interceptor) else\n                            CombinedContext(CombinedContext(left, element), interceptor)\n                    }\n                }\n            }\n\n    /**\n     * Returns a context containing elements from this context, but without an element with\n     * the specified [key].\n     */\n    public fun minusKey(key: Key<*>): CoroutineContext\n\n    /**\n     * Key for the elements of
[CoroutineContext]. [E] is a type of element with this key.\n     */\n    public interface Key<E : Element>\n\n    /**\n     * An element of the [CoroutineContext]. An element of the coroutine context is a singleton context by itself.\n     */\n    public interface Element : CoroutineContext {\n        /**\n         * A key of this coroutine context element.\n         */\n        public val key: Key<*>\n\n        public override operator fun <E : Element> get(key: Key<E>): E? =\n            @Suppress("UNCHECKED_CAST")\n                if (this.key == key) this as E else null\n\n        public override
fun <R> fold(initial: R, operation: (R, Element) -> R): R =\n            operation(initial, this)\n\n        public override
fun minusKey(key: Key<*>): CoroutineContext =\n            if (this.key == key) EmptyCoroutineContext else this\n    }\n}\n"/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of
this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.coroutines\n\nimport kotlin.coroutines.CoroutineContext.Element\nimport
kotlin.coroutines.CoroutineContext.Key\n\n/**\n * Base class for [CoroutineContext.Element] implementations.\n */\n@SinceKotlin("1.3")\npublic abstract class AbstractCoroutineContextElement(public override val key:
Key<*>) : Element\n\n/**\n * Base class for [CoroutineContext.Key] associated with polymorphic
[CoroutineContext.Element] implementation.\n * Polymorphic element implementation implies delegating its
[get][Element.get] and [minusKey][Element.minusKey]\n * to [getPolymorphicElement] and
[minusPolymorphicKey] respectively.\n */\n\n/**\n * Polymorphic elements can be extracted from the coroutine context
using both element key and its supertype key.\n * Example of polymorphic elements:\n */\n\n/**\n * open class
BaseElement : CoroutineContext.Element {\n *     companion object Key : CoroutineContext.Key<BaseElement>\n *
    override val key: CoroutineContext.Key<*> get() = Key\n *     // It is important to use getPolymorphicKey and

```

```

minusPolymorphicKey\n * override fun <E : CoroutineContext.Element> get(key: CoroutineContext.Key<E>):
E? = getPolymorphicElement(key)\n * override fun minusKey(key: CoroutineContext.Key<*>):
CoroutineContext = minusPolymorphicKey(key)\n * }\n *\n * class DerivedElement : BaseElement() {\n *
companion object Key : AbstractCoroutineContextKey<BaseElement, DerivedElement>(BaseElement, { it as?
DerivedElement })\n * }\n * // Now it is possible to query both `BaseElement` and `DerivedElement`\n *
someContext[BaseElement] // Returns BaseElement?, non-null both for BaseElement and DerivedElement
instances\n * someContext[DerivedElement] // Returns DerivedElement?, non-null only for DerivedElement
instance\n * ```\n * @param B base class of a polymorphic element\n * @param baseKey an instance of base key\n
* @param E element type associated with the current key\n * @param safeCast a function that can safely cast
abstract [CoroutineContext.Element] to the concrete [E] type\n * and return the element if it is a subtype
of [E] or `null` otherwise.\n * @SinceKotlin("1.3")\n * @ExperimentalStdlibApi\n * public abstract class
AbstractCoroutineContextKey<B : Element, E : B>(\n * baseKey: Key<B>,\n * private val safeCast: (element:
Element) -> E?)\n * ) : Key<E> {\n * private val topmostKey: Key<*> = if (baseKey is
AbstractCoroutineContextKey<*, *>) baseKey.topmostKey else baseKey\n *\n * internal fun tryCast(element:
Element): E? = safeCast(element)\n * internal fun isSubKey(key: Key<*>): Boolean = key === this || topmostKey
=== key\n *\n * Returns the current element if it is associated with the given [key] in a polymorphic manner
or `null` otherwise.\n * This method returns non-null value if either [Element.key] is equal to the given [key] or if
the [key] is associated\n * with [Element.key] via [AbstractCoroutineContextKey].\n * See
[AbstractCoroutineContextKey] for the example of usage.\n
*\n * @SinceKotlin("1.3")\n * @ExperimentalStdlibApi\n * public fun <E : Element>
Element.getPolymorphicElement(key: Key<E>): E? {\n * if (key is AbstractCoroutineContextKey<*, *>) {\n
* @Suppress("UNCHECKED_CAST")\n * return if (key.isSubKey(this.key)) key.tryCast(this) as? E else null\n
* }\n * @Suppress("UNCHECKED_CAST")\n * return if (this.key === key) this as E else null\n *\n * Returns empty coroutine context if the element is associated with the given [key] in a polymorphic manner\n * or
`null` otherwise.\n * This method returns empty context if either [Element.key] is equal to the given [key] or if the
[key] is associated\n * with [Element.key] via [AbstractCoroutineContextKey].\n * See
[AbstractCoroutineContextKey] for the example of usage.\n
*\n * @SinceKotlin("1.3")\n * @ExperimentalStdlibApi\n * public fun Element.minusPolymorphicKey(key: Key<*>):
CoroutineContext {\n * if (key is AbstractCoroutineContextKey<*, *>) {\n * return if (key.isSubKey(this.key)
&& key.tryCast(this) != null) EmptyCoroutineContext else this\n * }\n * return if (this.key === key)
EmptyCoroutineContext else this\n *\n * An empty coroutine context.\n * @SinceKotlin("1.3")\n * public
object EmptyCoroutineContext : CoroutineContext, Serializable {\n * private const val serialVersionUID: Long =
0\n * private fun readResolve(): Any = EmptyCoroutineContext\n *\n * public override fun <E : Element> get(key:
Key<E>): E? = null\n * public override fun <R> fold(initial: R, operation: (R, Element) -> R): R = initial\n *
public override fun plus(context: CoroutineContext): CoroutineContext = context\n * public override fun minusKey(key:
Key<*>): CoroutineContext = this\n * public override fun hashCode(): Int = 0\n * public override fun toString():
String = "EmptyCoroutineContext"\n *\n * ----- internal impl ----- \n *\n * this class is not
exposed, but is hidden inside implementations\n *\n * this is a left-biased list, so that `plus` works
naturally\n * @SinceKotlin("1.3")\n * internal class CombinedContext(\n * private val left: CoroutineContext,\n
* private val element: Element\n * ) : CoroutineContext, Serializable {\n *\n * override fun <E : Element> get(key:
Key<E>): E? {\n * var cur = this\n * while (true) {\n * cur.element[key]?.let { return it }\n * val next
= cur.left\n * if (next is CombinedContext) {\n * cur = next\n * } else {\n * return
next[key]\n * }\n * }\n *\n * public override fun <R> fold(initial: R, operation: (R, Element) -> R): R =\n
* operation(left.fold(initial, operation), element)\n *\n * public override fun minusKey(key: Key<*>):
CoroutineContext {\n * element[key]?.let { return left }\n * val newLeft = left.minusKey(key)\n * return
when {\n * newLeft === left -> this\n * newLeft === EmptyCoroutineContext -> element\n * else ->
CombinedContext(newLeft, element)\n * }\n *\n * private fun size(): Int {\n * var cur = this\n * var size
= 2\n * while (true) {\n * cur = cur.left as? CombinedContext ?: return size\n * size++\n * }\n * }\n
}

```

```

private fun contains(element: Element): Boolean =\n    get(element.key) == element\n\n private fun
containsAll(context: CombinedContext): Boolean {\n    var cur = context\n    while (true) {\n        if
(!contains(cur.element)) return false\n        val next = cur.left\n        if (next is CombinedContext) {\n
cur = next\n        } else {\n            return contains(next as Element)\n        }\n    }\n\n override fun
equals(other: Any?): Boolean =\n    this === other || other is CombinedContext && other.size() == size() &&
other.containsAll(this)\n\n override fun hashCode(): Int = left.hashCode() + element.hashCode()\n\n override
fun toString(): String =\n    "[" + fold("") { acc, element ->\n        if (acc.isEmpty()) element.toString() else
"$acc, $element"\n    } + "]\n\n private fun writeReplace(): Any {\n    val n = size()\n    val elements =
arrayOfNulls<CoroutineContext>(n)\n    var index = 0\n    fold(Unit) { _, element -> elements[index++] =
element }\n    check(index == n)\n    @Suppress("UNCHECKED_CAST")\n    return Serialized(elements
as Array<CoroutineContext>)\n }\n\n private class Serialized(val elements: Array<CoroutineContext>) :
Serializable {\n    companion object {\n        private const val serialVersionUID: Long = 0L\n    }\n\n private fun readResolve(): Any = elements.fold(EmptyCoroutineContext, CoroutineContext::plus)\n }\n\n", /*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code
is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\n@file:kotlin.jvm.JvmName("IntrinsicsKt")\n@file:kotlin.jvm.JvmMultifileClass\n\npackage
kotlin.coroutines.intrinsics\n\nimport kotlin.contracts.*\nimport kotlin.coroutines.*\nimport
kotlin.internal.InlineOnly\n\n/**\n * Obtains the current continuation instance inside suspend functions and either
suspends\n * currently running coroutine or returns result immediately without suspension.\n * If the [block]
returns the special [COROUTINE_SUSPENDED] value, it means that suspend function did suspend the execution
and will\n * not return any result immediately. In this case, the [Continuation] provided to the [block] shall be\n *
resumed by invoking [Continuation.resumeWith] at some moment in the\n * future when the result becomes
available to resume the computation.\n * Otherwise, the return value of the [block] must have a type assignable
to [T] and represents the result of this suspend function.\n * It means that the execution was not suspended and the
[Continuation] provided to the [block] shall not be invoked.\n * As the result type of the [block] is declared as
`Any?` and cannot be correctly type-checked,\n * its proper return type remains on the conscience of the suspend
function's author.\n * Invocation of [Continuation.resumeWith] resumes coroutine directly in the invoker's
thread without going through the\n * [ContinuationInterceptor] that might be present in the coroutine's
[CoroutineContext].\n * It is the invoker's responsibility to ensure that a proper invocation context is established.\n *
[Continuation.intercepted] can be used to acquire the intercepted continuation.\n * Note that it is not
recommended to call either [Continuation.resume] nor [Continuation.resumeWithException] functions
synchronously\n * in the same stackframe where suspension function is run. Use [suspendCoroutine] as a safer way
to obtain current\n * continuation instance.\n
*/\n\n@SinceKotlin("1.3")\n@InlineOnly\n@Suppress("UNUSED_PARAMETER",
"RedundantSuspendModifier")\npublic suspend inline fun <T>
suspendCoroutineUninterceptedOrReturn(crossinline block: (Continuation<T>) -> Any?): T {\n    contract {
callsInPlace(block, InvocationKind.EXACTLY_ONCE) }\n    throw NotImplementedError("Implementation of
suspendCoroutineUninterceptedOrReturn is intrinsic")\n}\n\n/**\n * This value is used as a return value of
[suspendCoroutineUninterceptedOrReturn] `block` argument to state that\n * the execution was suspended and will
not return any result immediately.\n * Note: this value should not be used in general code. Using it outside
of the context of\n * `suspendCoroutineUninterceptedOrReturn` function return value (including, but not limited
to,\n * storing this value in other properties, returning it from other functions, etc)\n * can lead to unspecified
behavior of the code.\n */\n\n// It is implemented as property with getter to avoid ProGuard <clinit> problem with
multifile IntrinsicsKt class\n@SinceKotlin("1.3")\npublic val COROUTINE_SUSPENDED: Any get() =
CoroutineSingletons.COROUTINE_SUSPENDED\n\n// Using enum here ensures two important properties:\n// 1.
It makes SafeContinuation serializable with all kinds of serialization frameworks (since all of them natively support
enums)\n// 2. It improves debugging experience, since you clearly see toString() value of those objects and what
package they come from\n@SinceKotlin("1.3")\n@PublishedApi // This class is Published API via serialized

```

representation of SafeContinuation, don't rename/move\ninternal enum class CoroutineSingletons {
COROUTINE_SUSPENDED, UNDECIDED, RESUMED } \n", "/*\n * Copyright 2010-2018 JetBrains s.r.o. and
Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.experimental\n\n/** Performs a bitwise AND
operation between the two values. */\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline infix fun
Byte.and(other: Byte): Byte = (this.toInt() and other.toInt()).toByte()\n\n/** Performs a bitwise OR operation
between the two values. */\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline infix fun
Byte.or(other: Byte): Byte = (this.toInt() or other.toInt()).toByte()\n\n/** Performs a bitwise XOR operation
between the two values. */\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline infix fun
Byte.xor(other: Byte): Byte = (this.toInt() xor other.toInt()).toByte()\n\n/** Inverts the bits in this value.
*/\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline fun Byte.inv(): Byte =
(this.toInt().inv()).toByte()\n\n/** Performs a bitwise AND operation between the two values.
*/\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline infix fun Short.and(other: Short): Short =
(this.toInt() and other.toInt()).toShort()\n\n/** Performs a bitwise OR operation between the two values.
*/\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline infix fun Short.or(other: Short): Short =
(this.toInt() or other.toInt()).toShort()\n\n/** Performs a bitwise XOR operation between the two values.
*/\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline infix fun Short.xor(other: Short): Short =
(this.toInt() xor other.toInt()).toShort()\n\n/** Inverts the bits in this value.
*/\n@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline fun Short.inv(): Short =
(this.toInt().inv()).toShort()\n\n", "/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\npackage kotlin.experimental\n\n/**\n * The experimental marker for type
inference augmenting annotations.\n * Any usage of a declaration annotated with
`@ExperimentalTypeInference` must be accepted either by\n * annotating that usage with the [OptIn] annotation,
e.g. `@OptIn(ExperimentalTypeInference::class)`,\n * or by using the compiler argument `-opt-
in=kotlin.experimental.ExperimentalTypeInference`.\n */\n@RequiresOptIn(level =
RequiresOptIn.Level.ERROR)\n@MustBeDocumented\n@Retention(AnnotationRetention.BINARY)\n@Target(A
nnotationTarget.ANNOTATION_CLASS)\n@SinceKotlin("1.3")\npublic annotation class
ExperimentalTypeInference\n", "/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\npackage kotlin.internal\n\n/**\n * Specifies that the corresponding type should be
ignored during type inference.\n */\n@Target(AnnotationTarget.TYPE)\n@Retention(AnnotationRetention.BINARY)\ninternal annotation class
NoInfer\n\n/**\n * Specifies that the constraint built for the type during type inference should be an equality one.\n */\n@Target(AnnotationTarget.TYPE)\n@Retention(AnnotationRetention.BINARY)\ninternal annotation class
Exact\n\n/**\n * Specifies that a corresponding member has the lowest priority in overload resolution.\n */\n@Target(AnnotationTarget.FUNCTION, AnnotationTarget.PROPERTY,
AnnotationTarget.CONSTRUCTOR)\n@Retention(AnnotationRetention.BINARY)\ninternal annotation class
LowPriorityInOverloadResolution\n\n/**\n * Specifies that the corresponding member has the highest priority in
overload resolution. Effectively this means that\n * an extension annotated with this annotation will win in overload
resolution over a member with the same signature.\n */\n@Target(AnnotationTarget.FUNCTION,
AnnotationTarget.PROPERTY)\n@Retention(AnnotationRetention.BINARY)\ninternal annotation class
HidesMembers\n\n/**\n * The value of this type parameter should be mentioned in input types (argument types,
receiver type or expected type).\n */\n@Target(AnnotationTarget.TYPE_PARAMETER)\n@Retention(AnnotationRetention.BINARY)\ninternal
annotation class OnlyInputTypes\n\n/**\n * Specifies that this function should not be called directly without
inlining\n */\n@Target(AnnotationTarget.FUNCTION, AnnotationTarget.PROPERTY,
AnnotationTarget.PROPERTY_GETTER,

```

AnnotationTarget.PROPERTY_SETTER)\n@Retention(AnnotationRetention.BINARY)\ninternal annotation class
InlineOnly\n\n/**\n * Specifies that this declaration can have dynamic receiver type.\n
*/\n@Target(AnnotationTarget.FUNCTION,
AnnotationTarget.PROPERTY)\n@Retention(AnnotationRetention.BINARY)\ninternal annotation class
DynamicExtension\n\n/**\n * The value of this parameter should be a property reference expression (`this::foo`),
referencing a `lateinit` property,\n * the backing field of which is accessible at the point where the corresponding
argument is passed.\n
*/\n@Target(AnnotationTarget.VALUE_PARAMETER)\n@Retention(AnnotationRetention.BINARY)\n@SinceK
otlin("1.2")\ninternal annotation class AccessibleLateinitPropertyLiteral\n\n/**\n * Specifies that this declaration is
only completely supported since the specified version.\n * The Kotlin compiler of an earlier version is going to
report a diagnostic on usages of this declaration.\n * The diagnostic message can be specified with [message], or via
[errorCode] (takes less space, but might not be immediately clear\n * to the user). The diagnostic severity can be
specified with [level]: WARNING/ERROR mean that either a warning or an error\n * is going to be reported,
HIDDEN means that the declaration is going to be removed from resolution completely.\n * [versionKind]
specifies which version should be compared with the [version] value, when compiling the usage of the annotated
declaration.\n * Note that prior to 1.2, only [RequireKotlinVersionKind.LANGUAGE_VERSION] was supported,
so the Kotlin compiler before 1.2 is going to\n * treat any [RequireKotlin] as if it requires the language version.
Since 1.2, the Kotlin compiler supports\n * [RequireKotlinVersionKind.LANGUAGE_VERSION],
[RequireKotlinVersionKind.COMPILER_VERSION] and [RequireKotlinVersionKind.API_VERSION].\n * If the
actual value of [versionKind] is something different (e.g. a new version kind, added in future versions of Kotlin),\n *
Kotlin 1.2 is going to ignore this [RequireKotlin] altogether, where as Kotlin before 1.2 is going to treat this as a
requirement\n * on the language version.\n * This annotation is erased at compile time; its arguments are stored
in a more compact form in the Kotlin metadata.\n */\n@Target(AnnotationTarget.CLASS,
AnnotationTarget.FUNCTION, AnnotationTarget.PROPERTY, AnnotationTarget.CONSTRUCTOR,
AnnotationTarget.TYPEALIAS)\n@Retention(AnnotationRetention.SOURCE)\n@Repeatable\n@SinceKotlin("1.
2")\ninternal annotation class RequireKotlin(\n    val version: String,\n    val message: String = "",\n    val level:
DeprecationLevel = DeprecationLevel.ERROR,\n    val versionKind: RequireKotlinVersionKind =
RequireKotlinVersionKind.LANGUAGE_VERSION,\n    val errorCode: Int = -1)\n\n/**\n * The kind of the
version that is required by [RequireKotlin].\n */\n@SinceKotlin("1.2")\ninternal enum class
RequireKotlinVersionKind {\n    LANGUAGE_VERSION,\n    COMPILER_VERSION,\n
API_VERSION,\n}\n\n/**\n * Specifies that this declaration is a part of special DSL, used for constructing
function's contract.\n */\n@Retention(AnnotationRetention.BINARY)\n@SinceKotlin("1.2")\ninternal annotation
class ContractsDsl\n\n/**\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\npackage kotlin.properties\n\nimport kotlin.reflect.KProperty\n\n/**\n * Standard
property delegates.\n */\npublic object Delegates {\n    /**\n     * Returns a property delegate for a read/write
property with a non-`null` value that is initialized not during\n     * object construction time but at a later time.
Trying to read the property before the initial value has been\n     * assigned results in an exception.\n     *
*/\n     * @sample samples.properties.Delegates.notNullDelegate\n     */\n     public fun <T : Any> notNull():
ReadWriteProperty<Any?, T> = NotNullVar()\n\n    /**\n     * Returns a property delegate for a read/write property
that calls a specified callback function when changed.\n     * @param initialValue the initial value of the property.\n
     * @param onChange the callback which is called after the change of the property is made. The value of the
property\n     * has already been changed when this callback is invoked.\n     */\n     * @sample
samples.properties.Delegates.observableDelegate\n     */\n     public inline fun <T> observable(initialValue: T,
crossinline onChange: (property: KProperty<*>, oldValue: T, newValue: T) -> Unit):\n
ReadWriteProperty<Any?, T> =\n     object : ObservableProperty<T>(initialValue) {\n         override fun
afterChange(property: KProperty<*>, oldValue: T, newValue: T) = onChange(property, oldValue, newValue)\n
     }\n\n    /**\n     * Returns a property delegate for a read/write property that calls a specified callback function when

```



```

/**\n * The callback which is called before a change to the property value is attempted.\n * The value of the
property hasn't been changed yet, when this callback is invoked.\n * If the callback returns `true` the value of the
property is being set to the new value,\n * and if the callback returns `false` the new value is discarded and the
property remains its old value.\n *^\n protected open fun beforeChange(property: KProperty<*>, oldValue: V,
newValue: V): Boolean = true\n\n /**\n * The callback which is called after the change of the property is made.
The value of the property\n * has already been changed when this callback is invoked.\n *^\n protected open
fun afterChange(property: KProperty<*>, oldValue: V, newValue: V): Unit { }\n\n public override fun
getValue(thisRef: Any?, property: KProperty<*>): V {\n return value\n }\n\n public override fun
setValue(thisRef: Any?, property: KProperty<*>, value: V) {\n val oldValue = this.value\n if
(!beforeChange(property, oldValue, value)) {\n return\n }\n this.value = value\n
afterChange(property, oldValue, value)\n }\n}", /*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n *^\n\n@file:Suppress("PackageDirectoryMismatch")\npackage
kotlin\n\nimport kotlin.reflect.*\n\n/**\n * An extension operator that allows delegating a read-only property of type
[V]\n * to a property reference to a property of type [V] or its subtype.\n *^\n * @receiver A property reference to a
read-only or mutable property of type [V] or its subtype.\n * The reference is without a receiver, i.e. it either
references a top-level property or\n * has the receiver bound to it.\n *^\n * Example:\n *^\n * class Login(val
username: String)\n * val defaultLogin = Login("Admin")\n * val defaultUsername by defaultLogin::username\n *
// equivalent to\n * val defaultUserName get() = defaultLogin.username\n *^\n\n *^\n\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline operator fun <V>
KProperty0<V>.getValue(thisRef: Any?, property: KProperty<*>): V {\n return get()\n}\n\n/**\n * An extension
operator that allows delegating a mutable property of type [V]\n * to a property reference to a mutable property of
the same type [V].\n *^\n * @receiver A property reference to a mutable property of type [V].\n * The reference is
without a receiver, i.e. it either references a top-level property or\n * has the receiver bound to it.\n *^\n * Example:\n
*\n * class Login(val username: String, var incorrectAttemptCounter: Int = 0)\n * val defaultLogin =
Login("Admin")\n * var defaultLoginAttempts by defaultLogin::incorrectAttemptCounter\n * // equivalent to\n *
var defaultLoginAttempts: Int\n * get() = defaultLogin.incorrectAttemptCounter\n * set(value) {\n
defaultLogin.incorrectAttemptCounter = value }\n *^\n\n *^\n\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline operator fun <V>
KMutableProperty0<V>.setValue(thisRef: Any?, property: KProperty<*>, value: V) {\n set(value)\n}\n\n/**\n
* An extension operator that allows delegating a read-only member or extension property of type [V]\n * to a
property reference to a member or extension property of type [V] or its subtype.\n *^\n * @receiver A property
reference to a read-only or mutable property of type [V] or its subtype.\n * The reference has an unbound receiver of
type [T].\n *^\n * Example:\n *^\n * class Login(val username: String)\n * val Login.user by
Login::username\n * // equivalent to\n * val Login.user get() = this.username\n *^\n\n *^\n\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline operator fun <T, V> KProperty1<T,
V>.getValue(thisRef: T, property: KProperty<*>): V {\n return get(thisRef)\n}\n\n/**\n * An extension operator
that allows delegating a mutable member or extension property of type [V]\n * to a property reference to a member
or extension mutable property of the same type [V].\n *^\n * @receiver A property reference to a read-only or
mutable property of type [V] or its subtype.\n * The reference has an unbound receiver of type [T].\n *^\n *
Example:\n *^\n * class Login(val username: String, var incorrectAttemptCounter: Int)\n * var Login.attempts
by Login::incorrectAttemptCounter\n * // equivalent to\n * var Login.attempts: Int\n * get() =
this.incorrectAttemptCounter\n * set(value) { this.incorrectAttemptCounter = value }\n *^\n\n *^\n\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline operator fun <T, V> KMutableProperty1<T,
V>.setValue(thisRef: T, property: KProperty<*>, value: V) {\n set(thisRef, value)\n}", /*\n * Copyright 2010-
2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n *^\n\npackage kotlin.random\n\nimport
kotlin.math.nextDown\n\n/**\n * An abstract class that is implemented by random number generator algorithms.\n

```

```

*\n * The companion object [Random.Default] is the default instance of [Random].\n *\n * To get a seeded instance
of random generator use [Random] function.\n *\n * @sample samples.random.Randoms.defaultRandom\n
*/\n@SinceKotlin("1.3")\npublic abstract class Random {\n\n    /**\n     * Gets the next random [bitCount] number
of bits.\n     *\n     * Generates an `Int` whose lower [bitCount] bits are filled with random values and the remaining
upper bits are zero.\n     *\n     * @param bitCount number of bits to generate, must be in range 0..32, otherwise the
behavior is unspecified.\n     *\n     * @sample samples.random.Randoms.nextBits\n     */\n    public abstract fun
nextBits(bitCount: Int): Int\n\n    /**\n     * Gets the next random `Int` from the random number generator.\n     *\n     * Generates an `Int` random value uniformly distributed between `Int.MIN_VALUE` and `Int.MAX_VALUE`
(inclusive).\n     *\n     * @sample samples.random.Randoms.nextInt\n     */\n    public open fun nextInt(): Int =
nextBits(32)\n\n    /**\n     * Gets the next random non-negative `Int` from the random number generator less than
the specified [until] bound.\n     *\n     * Generates an `Int` random value uniformly distributed between `0`
(inclusive) and the specified [until] bound (exclusive).\n     *\n     * @param until must be positive.\n     *\n     *
@throws IllegalArgumentException if [until] is negative or zero.\n     *\n     * @sample
samples.random.Randoms.nextIntFromUntil\n     */\n    public open fun nextInt(until: Int): Int = nextInt(0, until)\n\n    /**\n     * Gets the next random `Int` from the random number generator in the specified range.\n     *\n     *
Generates an `Int` random value uniformly distributed between the specified [from] (inclusive) and [until]
(exclusive) bounds.\n     *\n     * @throws IllegalArgumentException if [from] is greater than or equal to [until].\n     *\n     * @sample samples.random.Randoms.nextIntFromUntil\n     */\n    public open fun nextInt(from: Int, until:
Int): Int {\n        checkRangeBounds(from, until)\n        val n = until - from\n        if (n > 0 || n == Int.MIN_VALUE)\n        {\n            val rnd = if (n and -n == n) {\n                val bitCount = fastLog2(n)\n                nextBits(bitCount)\n
            } else {\n                var v: Int\n                do {\n                    val bits = nextInt().ushr(1)\n                    v = bits % n\n
                } while (bits - v + (n - 1) < 0)\n                v\n            }\n            return from + rnd\n        } else {\n            while
(true) {\n                val rnd = nextInt()\n                if (rnd in from until until) return rnd\n            }\n        }\n    }\n\n    /**\n     * Gets the next random `Long` from the random number generator.\n     *\n     * Generates a `Long` random
value uniformly distributed between `Long.MIN_VALUE` and `Long.MAX_VALUE` (inclusive).\n     *\n     *
@sample samples.random.Randoms.nextLong\n     */\n    public open fun nextLong(): Long =
nextInt().toLong().shl(32) + nextInt()\n\n    /**\n     * Gets the next random non-negative `Long` from the random
number generator less than the specified [until] bound.\n     *\n     * Generates a `Long` random value uniformly
distributed between `0` (inclusive) and the specified [until] bound (exclusive).\n     *\n     * @param until must be
positive.\n     *\n     * @throws IllegalArgumentException if [until] is negative or zero.\n     *\n     * @sample
samples.random.Randoms.nextLongFromUntil\n     */\n    public open fun nextLong(until: Long): Long =
nextLong(0, until)\n\n    /**\n     * Gets the next random `Long` from the random number generator in the specified
range.\n     *\n     * Generates a `Long` random value uniformly distributed between the specified [from] (inclusive)
and [until] (exclusive) bounds.\n     *\n     * @throws IllegalArgumentException if [from] is greater than or equal to
[until].\n     *\n     * @sample samples.random.Randoms.nextLongFromUntil\n     */\n    public open fun
nextLong(from: Long, until: Long): Long {\n        checkRangeBounds(from, until)\n        val n = until - from\n
if (n > 0) {\n            val rnd: Long\n            if (n and -n == n) {\n                val nLow = n.toInt()\n                val nHigh
= (n ushr 32).toInt()\n                rnd = when {\n                    nLow != 0 -> {\n                        val bitCount =
fastLog2(nLow)\n                        // toUInt().toLong()\n                        nextBits(bitCount).toLong() and
0xFFFF_FFFF\n                    }\n                    nHigh == 1 ->{\n                        // toUInt().toLong()\n
                        nextInt().toLong() and 0xFFFF_FFFF\n                    }\n                    else -> {\n                        val bitCount = fastLog2(nHigh)\n
                        nextBits(bitCount).toLong().shl(32) + (nextInt().toLong() and 0xFFFF_FFFF)\n                    }\n                }\n            } else {\n                var v: Long\n                do {\n                    val bits = nextLong().ushr(1)\n
v = bits % n\n                } while (bits - v + (n - 1) < 0)\n                rnd = v\n            }\n            return from + rnd\n        } else {\n            while (true) {\n                val rnd = nextLong()\n                if (rnd in from until until) return rnd\n
            }\n        }\n    }\n\n    /**\n     * Gets the next random [Boolean] value.\n     *\n     * @sample
samples.random.Randoms.nextBoolean\n     */\n    public open fun nextBoolean(): Boolean = nextBits(1) != 0\n\n    /**\n     * Gets the next random [Double] value uniformly distributed between 0 (inclusive) and 1 (exclusive).\n
}

```

```

*\n * @sample samples.random.Randoms.nextDouble\n *\n public open fun nextDouble(): Double =
doubleFromParts(nextBits(26), nextBits(27))\n\n /**\n * Gets the next random non-negative `Double` from the
random number generator less than the specified [until] bound.\n *\n * Generates a `Double` random value
uniformly distributed between 0 (inclusive) and [until] (exclusive).\n *\n * @throws IllegalArgumentException
if [until] is negative or zero.\n *\n * @sample samples.random.Randoms.nextDoubleFromUntil\n *\n
public open fun nextDouble(until: Double): Double = nextDouble(0.0, until)\n\n /**\n * Gets the next random
`Double` from the random number generator in the specified range.\n *\n * Generates a `Double` random value
uniformly distributed between the specified [from] (inclusive) and [until] (exclusive) bounds.\n *\n * [from]
and [until] must be finite otherwise the behavior is unspecified.\n *\n * @throws IllegalArgumentException if
[from] is greater than or equal to [until].\n *\n * @sample samples.random.Randoms.nextDoubleFromUntil\n
*\n public open fun nextDouble(from: Double, until: Double): Double {\n checkRangeBounds(from, until)\n
val size = until - from\n val r = if (size.isInfinite() && from.isFinite() && until.isFinite()) {\n val r1 =
nextDouble() * (until / 2 - from / 2)\n from + r1 + r1\n } else {\n from + nextDouble() * size\n
}\n return if (r >= until) until.nextDown() else r\n }\n\n /**\n * Gets the next random [Float] value
uniformly distributed between 0 (inclusive) and 1 (exclusive).\n *\n * @sample
samples.random.Randoms.nextFloat\n *\n public open fun nextFloat(): Float = nextBits(24) / (1 shl
24).toFloat()\n\n /**\n * Fills a subrange of the specified byte [array] starting from [fromIndex] inclusive and
ending [toIndex] exclusive\n * with random bytes.\n *\n * @return [array] with the subrange filled with
random bytes.\n *\n * @sample samples.random.Randoms.nextBytes\n *\n public open fun
nextBytes(array: ByteArray, fromIndex: Int = 0, toIndex: Int = array.size): ByteArray {\n require(fromIndex in
0..array.size && toIndex in 0..array.size) { "\nfromIndex ($fromIndex) or toIndex ($toIndex) are out of range:
0..${array.size}.\n" }\n require(fromIndex <= toIndex) { "\nfromIndex ($fromIndex) must be not greater than
toIndex ($toIndex).\n" }\n val steps = (toIndex - fromIndex) / 4\n val position = fromIndex\n repeat(steps) {\n val v = nextInt()\n array[position] = v.toByte()\n array[position + 1] =
v.ushr(8).toByte()\n array[position + 2] = v.ushr(16).toByte()\n array[position + 3] =
v.ushr(24).toByte()\n position += 4\n }\n val remainder = toIndex - position\n val vr =
nextBits(remainder * 8)\n for (i in 0 until remainder) {\n array[position + i] = vr.ushr(i * 8).toByte()\n
}\n return array\n }\n\n /**\n * Fills the specified byte [array] with random bytes and returns it.\n *\n
* @return [array] filled with random bytes.\n *\n * @sample samples.random.Randoms.nextBytes\n *\n
public open fun nextBytes(array: ByteArray): ByteArray = nextBytes(array, 0, array.size)\n\n /**\n * Creates a
byte array of the specified [size], filled with random bytes.\n *\n * @sample
samples.random.Randoms.nextBytes\n *\n public open fun nextBytes(size: Int): ByteArray =
nextBytes(ByteArray(size))\n\n\n /**\n * The default random number generator.\n *\n * On JVM this
generator is thread-safe, its methods can be invoked from multiple threads.\n *\n * @sample
samples.random.Randoms.defaultRandom\n *\n companion object Default : Random(), Serializable {\n
private val defaultRandom: Random = defaultPlatformRandom()\n private object Serialized : Serializable {\n
private const val serialVersionUID = 0L\n private fun readResolve(): Any = Random\n }\n\n private fun writeReplace(): Any = Serialized\n\n override fun nextBits(bitCount: Int): Int =
defaultRandom.nextBits(bitCount)\n override fun nextInt(): Int = defaultRandom.nextInt()\n override fun
nextInt(until: Int): Int = defaultRandom.nextInt(until)\n override fun nextInt(from: Int, until: Int): Int =
defaultRandom.nextInt(from, until)\n\n override fun nextLong(): Long = defaultRandom.nextLong()\n
override fun nextLong(until: Long): Long = defaultRandom.nextLong(until)\n override fun nextLong(from:
Long, until: Long): Long = defaultRandom.nextLong(from, until)\n\n override fun nextBoolean(): Boolean =
defaultRandom.nextBoolean()\n\n override fun nextDouble(): Double = defaultRandom.nextDouble()\n
override fun nextDouble(until: Double): Double = defaultRandom.nextDouble(until)\n override fun
nextDouble(from: Double, until: Double): Double = defaultRandom.nextDouble(from, until)\n\n override fun
nextFloat(): Float = defaultRandom.nextFloat()\n\n override fun nextBytes(array: ByteArray): ByteArray =
defaultRandom.nextBytes(array)\n\n override fun nextBytes(size: Int): ByteArray =

```



```

greater than or equal to [until].\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
Random.nextIntUInt(from: UInt, until: UInt): UInt { \n    checkUIntRangeBounds(from, until)\n    val signedFrom =
from.toInt() xor Int.MIN_VALUE\n    val signedUntil = until.toInt() xor Int.MIN_VALUE\n    val signedResult =
nextInt(signedFrom, signedUntil) xor Int.MIN_VALUE\n    return signedResult.toUInt()\n}\n\n/**\n * Gets the next
random [UInt] from the random number generator in the specified [range].\n * \n * Generates a [UInt] random value
uniformly distributed in the specified [range]:\n * from `range.start` inclusive to `range.endInclusive` inclusive.\n
*\n * @throws IllegalArgumentException if [range] is empty.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
Random.nextIntUInt(range: UIntRange): UInt = when { \n    range.isEmpty() -> throw
IllegalArgumentException("Cannot get random in empty range: $range")\n    range.last < UInt.MAX_VALUE ->
nextIntUInt(range.first, range.last + 1u)\n    range.first > UInt.MIN_VALUE -> nextUInt(range.first - 1u, range.last) +
1u\n    else -> nextUInt()\n}\n\n/**\n * Gets the next random [ULong] from the random number generator.\n * \n *
Generates a [ULong] random value uniformly distributed between [ULong.MIN_VALUE] and
[ULong.MAX_VALUE] (inclusive).\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
Random.nextULong(): ULong = nextLong().toULong()\n\n/**\n * Gets the next random [ULong] from the random
number generator less than the specified [until] bound.\n * \n * Generates a [ULong] random value uniformly
distributed between `0` (inclusive) and the specified [until] bound (exclusive).\n * \n * @throws
IllegalArgumentException if [until] is zero.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
Random.nextULong(until: ULong): ULong = nextULong(0uL, until)\n\n/**\n * Gets the next random [ULong] from
the random number generator in the specified range.\n * \n * Generates a [ULong] random value uniformly
distributed between the specified [from] (inclusive) and [until] (exclusive) bounds.\n * \n * @throws
IllegalArgumentException if [from] is greater than or equal to [until].\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
Random.nextULong(from: ULong, until: ULong): ULong { \n    checkULongRangeBounds(from, until)\n    val
signedFrom = from.toLong() xor Long.MIN_VALUE\n    val signedUntil = until.toLong() xor
Long.MIN_VALUE\n    val signedResult = nextLong(signedFrom, signedUntil) xor Long.MIN_VALUE\n    return
signedResult.toULong()\n}\n\n/**\n * Gets the next random [ULong] from the random number generator in
the specified [range].\n * \n * Generates a [ULong] random value uniformly distributed in the specified [range]:\n *
from `range.start` inclusive to `range.endInclusive` inclusive.\n * \n * @throws IllegalArgumentException if [range]
is empty.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
Random.nextULong(range: ULongRange): ULong = when { \n    range.isEmpty() -> throw
IllegalArgumentException("Cannot get random in empty range: $range")\n    range.last < ULong.MAX_VALUE -
> nextULong(range.first, range.last + 1u)\n    range.first > ULong.MIN_VALUE -> nextULong(range.first - 1u,
range.last) + 1u\n    else -> nextULong()\n}\n\n/**\n * Fills the specified unsigned byte [array] with random bytes
and returns it.\n * \n * @return [array] filled with random bytes.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun Random.nextUBytes(array: UByteArray):
UByteArray { \n    nextBytes(array.asByteArray())\n    return array\n}\n\n/**\n * Creates an unsigned byte array of
the specified [size], filled with random bytes.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic
fun Random.nextUBytes(size: Int): UByteArray = nextBytes(size).asUByteArray()\n\n/**\n * Fills a subrange of
the specified `UByte` [array] starting from [fromIndex] inclusive and ending [toIndex] exclusive with random
UBytes.\n * \n * @return [array] with the subrange filled with random bytes.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun Random.nextUBytes(array: UByteArray,
fromIndex: Int = 0, toIndex: Int = array.size): UByteArray { \n    nextBytes(array.asByteArray(), fromIndex,
toIndex)\n    return array\n}\n\ninternal fun checkUIntRangeBounds(from: UInt, until: UInt) = require(until >
from) { boundsErrorMessage(from, until) }\ninternal fun checkULongRangeBounds(from: ULong, until: ULong) =

```

```

require(until > from) { boundsErrorMessage(from, until) }n", /*\n * Copyright 2010-2018 JetBrains s.r.o. and
Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.random\n\n/**\n * Random number generator,
using Marsaglia's "xorwow" algorithm\n * Cycles after 2^192 - 2^32 repetitions.\n * For more details, see
Marsaglia, George (July 2003). "Xorshift RNGs". Journal of Statistical Software. 8 (14).
doi:10.18637/jss.v008.i14\n * Available at https://www.jstatsoft.org/v08/i14/paper\n */\n\ninternal class
XorWowRandom internal constructor(\n private var x: Int,\n private var y: Int,\n private var z: Int,\n private
var w: Int,\n private var v: Int,\n private var addend: Int\n) : Random(), Serializable {\n\n internal
constructor(seed1: Int, seed2: Int) :n this(seed1, seed2, 0, 0, seed1.inv(), (seed1 shl 10) xor (seed2 ushr
4))\n\n init {\n require((x or y or z or w or v) != 0) { "Initial state must have at least one non-zero element.\n"
}\n\n // some trivial seeds can produce several values with zeroes in upper bits, so we discard first 64\n
repeat(64) { nextInt() }\n }\n\n override fun nextInt(): Int {\n // Equivalent to the xorwow algorithm\n //
From Marsaglia, G. 2003. Xorshift RNGs. J. Statis. Soft. 8, 14, p. 5\n var t = x\n t = t xor (t ushr 2)\n x
= y\n y = z\n z = w\n val v0 = v\n w = v0\n t = (t xor (t shl 1)) xor v0 xor (v0 shl 4)\n v =
t\n addend += 362437\n return t + addend\n }\n\n override fun nextBits(bitCount: Int): Int =\n
nextInt().takeUpperBits(bitCount)\n\n private companion object {\n private const val serialVersionUID: Long
= 0L\n }\n}\n", /*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n *
Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("RangesKt")\n\npackage
kotlin.ranges\n\n/**\n * Represents a range of [Comparable] values.\n */\n\nprivate open class ComparableRange<T> :
Comparable<T>>(\n override val start: T,\n override val endInclusive: T\n) : ClosedRange<T> {\n\n override
fun equals(other: Any?): Boolean {\n return other is ComparableRange<*> && (isEmpty() && other.isEmpty())
||\n start == other.start && endInclusive == other.endInclusive\n }\n\n override fun hashCode(): Int
{\n return if (isEmpty()) -1 else 31 * start.hashCode() + endInclusive.hashCode()\n }\n\n override fun
toString(): String = "$start..$endInclusive"\n}\n\n/**\n * Creates a range from this [Comparable] value to the
specified [that] value.\n * This value needs to be smaller than or equal to [that] value, otherwise the returned
range will be empty.\n * @sample samples.ranges.Ranges.rangeFromComparable\n */\n\npublic operator fun <T> :
Comparable<T>> T.rangeTo(that: T): ClosedRange<T> = ComparableRange(this, that)\n\n\n/**\n * Represents a
range of floating point numbers.\n * Extends [ClosedRange] interface providing custom operation
[lessThanOrEquals] for comparing values of range domain type.\n * This interface is implemented by floating
point ranges returned by [Float.rangeTo] and [Double.rangeTo] operators to\n * achieve IEEE-754 comparison order
instead of total order of floating point numbers.\n */\n\n@SinceKotlin("1.1")\n\npublic interface
ClosedFloatingPointRange<T> : Comparable<T>> : ClosedRange<T> {\n override fun contains(value: T): Boolean
= lessThanOrEquals(start, value) && lessThanOrEquals(value, endInclusive)\n override fun isEmpty(): Boolean =
!lessThanOrEquals(start, endInclusive)\n\n /**\n * Compares two values of range domain type and returns true
if first is less than or equal to second.\n */\n fun lessThanOrEquals(a: T, b: T): Boolean\n}\n\n\n/**\n * A
closed range of values of type `Double`.\n * Numbers are compared with the ends of this range according to
IEEE-754.\n */\n\nprivate class ClosedDoubleRange(\n start: Double,\n endInclusive: Double\n) :
ClosedFloatingPointRange<Double> {\n private val _start = start\n private val _endInclusive = endInclusive\n
override val start: Double get() = _start\n override val endInclusive: Double get() = _endInclusive\n\n override
fun lessThanOrEquals(a: Double, b: Double): Boolean = a <= b\n\n override fun contains(value: Double): Boolean
= value >= _start && value <= _endInclusive\n override fun isEmpty(): Boolean = !(_start <= _endInclusive)\n\n
override fun equals(other: Any?): Boolean {\n return other is ClosedDoubleRange && (isEmpty() &&
other.isEmpty()) ||\n _start == other._start && _endInclusive == other._endInclusive\n }\n\n override
fun hashCode(): Int {\n return if (isEmpty()) -1 else 31 * _start.hashCode() + _endInclusive.hashCode()\n
}\n\n override fun toString(): String = "$_start..$_endInclusive"\n}\n\n\n/**\n * Creates a range from this [Double]
value to the specified [that] value.\n * Numbers are compared with the ends of this range according to IEEE-
754.\n * @sample samples.ranges.Ranges.rangeFromDouble\n */\n\n@SinceKotlin("1.1")\n\npublic operator fun

```

```

Double.rangeTo(that: Double): ClosedFloatingPointRange<Double> = ClosedDoubleRange(this, that)\n\n/**\n *
A closed range of values of type `Float`.\n *\n * Numbers are compared with the ends of this range according to
IEEE-754.\n */\nprivate class ClosedFloatRange(\n    start: Float,\n    endInclusive: Float\n) :
ClosedFloatingPointRange<Float> {\n    private val _start = start\n    private val _endInclusive = endInclusive\n
override val start: Float get() = _start\n    override val endInclusive: Float get() = _endInclusive\n\n    override fun
lessThanOrEquals(a: Float, b: Float): Boolean = a <= b\n\n    override fun contains(value: Float): Boolean = value
>= _start && value <= _endInclusive\n    override fun isEmpty(): Boolean = !(_start <= _endInclusive)\n\n
override fun equals(other: Any?): Boolean {\n        return other is ClosedFloatRange && (isEmpty() &&
other.isEmpty()) ||\n            _start == other._start && _endInclusive == other._endInclusive\n    }\n\n    override
fun hashCode(): Int {\n        return if (isEmpty()) -1 else 31 * _start.hashCode() + _endInclusive.hashCode()\n    }\n\n    override fun toString(): String = "$_start..$_endInclusive"\n}\n\n/**\n * Creates a range from this [Float]
value to the specified [that] value.\n *\n * Numbers are compared with the ends of this range according to IEEE-
754.\n * @sample samples.ranges.Ranges.rangeFromFloat\n */\n@SinceKotlin("1.1")\npublic operator fun
Float.rangeTo(that: Float): ClosedFloatingPointRange<Float> = ClosedFloatRange(this, that)\n\n/**\n * Returns
`true` if this iterable range contains the specified [element].\n *\n * Always returns `false` if the [element] is `null`.\n */\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline operator fun <T, R> R.contains(element: T?):
Boolean where T : Any, R : Iterable<T>, R : ClosedRange<T> =\n    element != null &&
contains(element)\n\ninternal fun checkStepIsPositive(isPositive: Boolean, step: Number) {\n    if (!isPositive)
throw IllegalArgumentException("Step must be positive, was: $step.")\n}\n\n/*\n * Copyright 2010-2019
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\n@file:kotlin.jvm.JvmName("KClasses")\n@file:Suppress("UNCHECKED_CAST")\n\npackage
kotlin.reflect\n\nimport kotlin.internal.LowPriorityInOverloadResolution\n\n/**\n * Casts the given [value] to the
class represented by this [KClass] object.\n * Throws an exception if the value is `null` or if it is not an instance of
this class.\n *\n * This is an experimental function that behaves as a similar function from kotlin.reflect.full on
JVM.\n *\n * @see [KClass.isInstance]\n * @see [KClass.safeCast]\n
*/\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@LowPriorityInOverloadResoluti
on\nfun <T : Any> KClass<T>.cast(value: Any?): T {\n    if (!isInstance(value)) throw ClassCastException("Value
cannot be cast to $qualifiedOrSimpleName")\n    return value as T\n}\n\n// TODO: replace with qualifiedName
when it is fully supported in K/JS\n\ninternal expect val KClass<*>.qualifiedOrSimpleName: String?\n\n/**\n * Casts
the given [value] to the class represented by this [KClass] object.\n * Returns `null` if the value is `null` or if it is not
an instance of this class.\n *\n * This is an experimental function that behaves as a similar function from
kotlin.reflect.full on JVM.\n *\n * @see [KClass.isInstance]\n * @see [KClass.cast]\n
*/\n\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@LowPriorityInOverloadResoluti
on\nfun <T : Any> KClass<T>.safeCast(value: Any?): T? {\n    return if (isInstance(value)) value as T else
null\n}\n\n/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of
this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\npackage kotlin.reflect\n\nimport kotlin.jvm.JvmField\nimport kotlin.jvm.JvmStatic\n\n/**\n * Represents
a type projection. Type projection is usually the argument to another type in a type usage.\n * For example, in the
type `Array<out Number>`, `out Number` is the covariant projection of the type represented by the class
`Number`.\n *\n * Type projection is either the star projection, or an entity consisting of a specific type plus optional
variance.\n *\n * See the [Kotlin language documentation](https://kotlinlang.org/docs/reference/generics.html#type-
projections)\n * for more information.\n */\n\n@SinceKotlin("1.1")\npublic data class KTypeProjection
constructor(\n    /**\n     * The use-site variance specified in the projection, or `null` if this is a star projection.\n
*/\n    variance: KVariance?,\n    /**\n     * The type specified in the projection, or `null` if this is a star
projection.\n     */\n    type: KType?\n) {\n    init {\n        require((variance == null) == (type == null))\n
\n        if (variance == null)\n            "Star projection must have no type specified." else\n
\n            "The projection variance $variance requires type to be specified." }\n    }\n\n    override fun toString():

```



```

subsequence to append.\n * @param endIndex the end (exclusive) of the subsequence to append.\n * \n *
@throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of
range of the [value] character sequence indices or when `startIndex > endIndex`. \n * \n fun append(value:
CharSequence?, startIndex: Int, endIndex: Int): Appendable\n}\n\n/** \n * Appends a subsequence of the specified
character sequence [value] to this Appendable and returns this instance.\n * \n * @param value the character
sequence from which a subsequence is appended.\n * @param startIndex the beginning (inclusive) of the
subsequence to append.\n * @param endIndex the end (exclusive) of the subsequence to append.\n * \n * @throws
IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of the
[value] character sequence indices or when `startIndex > endIndex`. \n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun <T : Appendable>
T.appendRange(value: CharSequence, startIndex: Int, endIndex: Int): T {\n
@Suppress("UNCHECKED_CAST")\n return append(value, startIndex, endIndex) as T\n}\n\n/** \n * Appends
all arguments to the given [Appendable].\n *\npublic fun <T : Appendable> T.append(vararg value:
CharSequence?): T {\n for (item in value)\n append(item)\n return this\n}\n\n/** Appends a line feed
character (`\n`) to this Appendable. *\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun
Appendable.appendLine(): Appendable = append("\n")\n\n/** Appends value to the given Appendable and a line
feed character (`\n`) after it. *\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun
Appendable.appendLine(value: CharSequence?): Appendable = append(value).appendLine()\n\n/** Appends value
to the given Appendable and a line feed character (`\n`) after it.
*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun Appendable.appendLine(value: Char):
Appendable = append(value).appendLine()\n\n\ninternal fun <T> Appendable.appendElement(element: T,
transform: ((T) -> CharSequence)?) {\n when {\n transform != null -> append(transform(element))\n
element is CharSequence? -> append(element)\n element is Char -> append(element)\n else ->
append(element.toString())\n }\n}\n\n"/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n
*\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("StringsKt")\n\npackage
kotlin.text\n\n/** \n * Trims leading whitespace characters followed by [marginPrefix] from every line of a source
string and removes\n * the first and the last lines if they are blank (notice difference blank vs empty).\n * \n * Doesn't
affect a line if it doesn't contain [marginPrefix] except the first and the last blank lines.\n * \n * Doesn't preserve the
original line endings.\n * \n * @param marginPrefix non-blank string, which is used as a margin delimiter. Default is
`|` (pipe character).\n * \n * @sample samples.text.Strings.trimMargin\n * @see trimIndent\n * @see
kotlin.text.isWhitespace\n *\npublic fun String.trimMargin(marginPrefix: String = "|"): String =\n
replaceIndentByMargin("|", marginPrefix)\n\n\n/** \n * Detects indent by [marginPrefix] as it does [trimMargin] and
replace it with [newIndent].\n * \n * @param marginPrefix non-blank string, which is used as a margin delimiter.
Default is `|` (pipe character).\n *\npublic fun String.replaceIndentByMargin(newIndent: String = "|",
marginPrefix: String = "|"): String {\n require(marginPrefix.isNotBlank()) { "marginPrefix must be non-blank
string." }\n val lines = lines()\n return lines.reindent(length + newIndent.length * lines.size,
getIndentFunction(newIndent), { line ->\n val firstNonWhitespaceIndex = line.indexOfFirst { !it.isWhitespace()
}\n\n when {\n firstNonWhitespaceIndex == -1 -> null\n line.startsWith(marginPrefix,
firstNonWhitespaceIndex) -> line.substring(firstNonWhitespaceIndex + marginPrefix.length)\n else -> null\n
}\n })\n}\n\n\n/** \n * Detects a common minimal indent of all the input lines, removes it from every line and
also removes the first and the last\n * lines if they are blank (notice difference blank vs empty).\n * \n * Note that
blank lines do not affect the detected indent level.\n * \n * In case if there are non-blank lines with no leading
whitespace characters (no indent at all) then the\n * common indent is 0, and therefore this function doesn't change
the indentation.\n * \n * Doesn't preserve the original line endings.\n * \n * @sample
samples.text.Strings.trimIndent\n * @see trimMargin\n * @see kotlin.text.isBlank\n *\npublic fun
String.trimIndent(): String = replaceIndent("|")\n\n\n/** \n * Detects a common minimal indent like it does

```

```

[trimIndent] and replaces it with the specified [newIndent].\n *^public fun String.replaceIndent(newIndent: String
= "\\"): String {\n    val lines = lines()\n    val minCommonIndent = lines\n        .filter(String::isNotBlank)\n
.map(String::indentWidth)\n        .minOrNull() ?: 0\n    return lines.reindent(length + newIndent.length *
lines.size, getIndentFunction(newIndent), { line -> line.drop(minCommonIndent) })\n}\n\n/**\n * Prepends [indent]
to every line of the original string.\n * Doesn't preserve the original line endings.\n *^public fun
String.prependIndent(indent: String = "\\"): String =\n    lineSequence()\n        .map {\n            when {\n
it.isBlank() -> {\n                when {\n                    it.length < indent.length -> indent\n                        else -> it\n
                }\n            }\n            else -> indent + it\n        }\n    }.joinToString("\\n")\n\nprivate fun
String.indentWidth(): Int = indexOfFirst { !it.isWhitespace() }.let { if (it == -1) length else it }\n\nprivate fun
getIndentFunction(indent: String) = when {\n    indent.isEmpty() -> { line: String -> line }\n    else -> { line: String -
> indent + line }\n}\n\nprivate inline fun List<String>.reindent(\n    resultSizeEstimate: Int,\n    indentAddFunction:
(String) -> String,\n    indentCutFunction: (String) -> String?): String {\n    val lastIndex = lastIndex\n    return
mapIndexedNotNull { index, value ->\n        if ((index == 0 || index == lastIndex) && value.isBlank())\n            null\n        else\n            indentCutFunction(value)?.let(indentAddFunction) ?: value\n    }\n
}.joinTo(StringBuilder(resultSizeEstimate), "\\n")\n    .toString()\n}\n","/*\n * Copyright 2010-2018 JetBrains
s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0
license that can be found in the license/LICENSE.txt file.\n *^package kotlin.text\n\n/**\n * Defines names for
Unicode symbols used in proper Typography.\n *^public object Typography {\n    /** The character &#x22;\n
\u2013 quotation mark *\n    public const val quote: Char = "\u0022"\n    /** The character &#x24;\n
\u2013 dollar sign *\n    public const val dollar: Char = "\u0024"\n    /** The character &#x26;\n
\u2013 ampersand *\n    public const val amp: Char = "\u0026"\n    /** The character &#x3C;\n
\u2013 less-than sign *\n    public const val less: Char = "\u003C"\n    /** The character &#x3E;\n
\u2013 greater-than sign *\n    public const val greater: Char = "\u003E"\n    /** The non-breaking space character *\n
    public const val nbsp: Char = "\u00A0"\n    /** The character &#xD7;\n
    public const val times: Char = "\u00D7"\n    /** The character &#xA2;\n
    public const val cent: Char = "\u00A2"\n    /** The character &#xA3;\n
    public const val pound: Char = "\u00A3"\n    /** The character &#xA7;\n
    public const val section: Char = "\u00A7"\n    /** The character &#xA9;\n
    public const val copyright: Char = "\u00A9"\n    /** The character &#xAB;\n
    @SinceKotlin("1.6")\n    public const val leftGuillemet: Char = "\u00AB"\n    /** The character &#xBB;\n
    @SinceKotlin("1.6")\n    public const val rightGuillemet: Char = "\u00BB"\n    /** The character &#xAE;\n
    public const val registered: Char = "\u00AE"\n    /** The character &#xB0;\n
    public const val degree: Char = "\u00B0"\n    /** The character &#xB1;\n
    public const val plusMinus: Char = "\u00B1"\n    /** The character &#xB6;\n
    public const val paragraph: Char = "\u00B6"\n    /** The character &#xB7;\n
    public const val middleDot: Char = "\u00B7"\n    /** The character &#xBD;\n
    public const val half: Char = "\u00BD"\n    /** The character &#x2013;\n
    public const val ndash: Char = "\u2013"\n    /** The character &#x2014;\n
    public const val mdash: Char = "\u2014"\n    /** The character &#x2018;\n
    public const val leftSingleQuote: Char = "\u2018"\n    /** The character &#x2019;\n
    public const val rightSingleQuote: Char = "\u2019"\n    /** The character &#x201A;\n
    public const val lowSingleQuote: Char = "\u201A"\n    /** The character &#x201C;\n
    public const val leftDoubleQuote: Char = "\u201C"\n    /** The character &#x201D;\n
    public const val rightDoubleQuote: Char = "\u201D"\n    /** The character &#x201E;\n
    public const val lowDoubleQuote: Char = "\u201E"\n    /** The character &#x2020;\n
    public const val dagger: Char = "\u2020"\n    /** The character &#x2021;\n
    public const val doubleDagger: Char = "\u2021"\n    /** The character &#x2022;\n
    public const val bullet: Char = "\u2022"\n    /** The character &#x2026;\n
    public const val ellipsis: Char = "\u2026"\n    /** The character &#x2032;\n
    public const val prime: Char = "\u2032"\n    /** The character &#x2033;\n
    public const val doublePrime: Char = "\u2033"\n    /** The character &#x20AC;\n
    public const val euro: Char = "\u20AC"\n    /** The character &#x2122;\n
    public const val tm: Char = "\u2122"\n    /** The character &#x2248;\n
    public const val almostEqual: Char = "\u2248"\n    /** The character &#x2260;\n
    public const val notEqual: Char = "\u2260"\n    /** The character &#x2264;\n
    public const val lessOrEqual: Char = "\u2264"\n    /** The character &#x2265;\n
    public const val greaterOrEqual: Char = "\u2265"\n}\n\n/** The character &#xAB; *\n

```



```

component4(): String = match.groupValues[4]\n    @kotlin.internal.InlineOnly\n    public operator inline fun
component5(): String = match.groupValues[5]\n    @kotlin.internal.InlineOnly\n    public operator inline fun
component6(): String = match.groupValues[6]\n    @kotlin.internal.InlineOnly\n    public operator inline fun
component7(): String = match.groupValues[7]\n    @kotlin.internal.InlineOnly\n    public operator inline fun
component8(): String = match.groupValues[8]\n    @kotlin.internal.InlineOnly\n    public operator inline fun
component9(): String = match.groupValues[9]\n    @kotlin.internal.InlineOnly\n    public operator inline fun
component10(): String = match.groupValues[10]\n\n    /**\n     * Returns destructured group values as a list of
strings.\n     * First value in the returned list corresponds to the value of the first group, and so on.\n     */\n
* @sample samples.text.Regexps.matchDestructuringToGroupValues\n    */\n    public fun toList():
List<String> = match.groupValues.subList(1, match.groupValues.size)\n    }\n}","/*\n * Copyright 2010-2021
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\n@file:kotlin.jvm.JvmMultifileClass()\n@file:kotlin.jvm.JvmName("DurationUnitKt")\n\npackage
kotlin.time\n\n/**\n * The list of possible time measurement units, in which a duration can be expressed.\n */\n *
The smallest time unit is [NANOSECONDS] and the largest is [DAYS], which corresponds to exactly 24
[HOURS].\n */\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalTime::class)\npublic expect enum class
DurationUnit {\n    /**\n     * Time unit representing one nanosecond, which is 1/1000 of a microsecond.\n     */\n    NANOSECONDS,\n    /**\n     * Time unit representing one microsecond, which is 1/1000 of a millisecond.\n     */\n    MICROSECONDS,\n    /**\n     * Time unit representing one millisecond, which is 1/1000 of a second.\n     */\n    MILLISECONDS,\n    /**\n     * Time unit representing one second.\n     */\n    SECONDS,\n    /**\n     *
Time unit representing one minute.\n     */\n    MINUTES,\n    /**\n     * Time unit representing one hour.\n     */\n    HOURS,\n    /**\n     * Time unit representing one day, which is always equal to 24 hours.\n     */\n    DAYS;\n}\n\n/**\n * Converts the given time duration [value] expressed in the specified [sourceUnit] into the specified
[targetUnit].\n */\n@SinceKotlin("1.3")\n\ninternal expect fun convertDurationUnit(value: Double, sourceUnit:
DurationUnit, targetUnit: DurationUnit): Double\n\n// overflown result is
unspecified\n@SinceKotlin("1.5")\n\ninternal expect fun convertDurationUnitOverflow(value: Long, sourceUnit:
DurationUnit, targetUnit: DurationUnit): Long\n\n// overflown result is coerced in the Long range
boundaries\n@SinceKotlin("1.5")\n\ninternal expect fun convertDurationUnit(value: Long, sourceUnit:
DurationUnit, targetUnit: DurationUnit):
Long\n\n\n@SinceKotlin("1.3")\n\n@Suppress("REDUNDANT_ELSE_IN_WHEN")\n\ninternal fun
DurationUnit.shortName(): String = when (this) {\n    DurationUnit.NANOSECONDS -> "ns"\n    DurationUnit.MICROSECONDS -> "us"\n    DurationUnit.MILLISECONDS -> "ms"\n    DurationUnit.SECONDS -> "s"\n    DurationUnit.MINUTES -> "m"\n    DurationUnit.HOURS -> "h"\n    DurationUnit.DAYS -> "d"\n    else -> error("Unknown unit: $this")\n}\n\n@SinceKotlin("1.5")\n\ninternal fun
durationUnitByShortName(shortName: String): DurationUnit = when (shortName) {\n    "ns" ->
DurationUnit.NANOSECONDS\n    "us" -> DurationUnit.MICROSECONDS\n    "ms" ->
DurationUnit.MILLISECONDS\n    "s" -> DurationUnit.SECONDS\n    "m" -> DurationUnit.MINUTES\n    "h" -> DurationUnit.HOURS\n    "d" -> DurationUnit.DAYS\n    else -> throw
IllegalArgumentException("Unknown duration unit short name:
$shortName")\n}\n\n@SinceKotlin("1.5")\n\ninternal fun durationUnitByIsoChar(isoChar: Char,
isTimeComponent: Boolean): DurationUnit =\n    when {\n        !isTimeComponent -> {\n            when (isoChar)
{\n                'D' -> DurationUnit.DAYS\n                else -> throw IllegalArgumentException("Invalid or
unsupported duration ISO non-time unit: $isoChar")\n            }\n        }\n        else -> {\n            when (isoChar) {\n                'H' -> DurationUnit.HOURS\n                'M' -> DurationUnit.MINUTES\n                'S' ->
DurationUnit.SECONDS\n                else -> throw IllegalArgumentException("Invalid duration ISO time unit:
$isoChar")\n            }\n        }\n    }\n}","/*\n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\npackage kotlin.time\n\nimport kotlin.annotation.AnnotationTarget.*\n\n/**\n *

```

This annotation marks the experimental preview of the standard library API for measuring time and working with durations.

Note that this API is in a preview state and has a very high chance of being changed in the future. Do not use it if you develop a library since your library will become binary incompatible with the future versions of the standard library. Any usage of a declaration annotated with `@ExperimentalTime` must be accepted either by annotating that usage with the `[OptIn]` annotation, e.g. `@OptIn(ExperimentalTime::class)`, or by using the compiler argument `-opt-in=kotlin.time.ExperimentalTime`.

`@RequiresOptIn(level = RequiresOptIn.Level.ERROR)`
`@MustBeDocumented`
`@Retention(AnnotationRetention.BINARY)`
`@Target(Class, AnnotationClass, Property, Field, LocalVariable, ValueParameter, Constructor, Function, PropertyGetter, PropertySetter, TypeAlias)`

public annotation class ExperimentalTime {
 /* Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.
 Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file. */
 A source of time for measuring time intervals.
 The only operation provided by the time source is `[markNow]`. It returns a `[TimeMark]`, which can be used to query the elapsed time later.
 @see `[measureTime]`
 @see `[measureTimedValue]`
 @SinceKotlin("1.3")
 @ExperimentalTime
 public interface TimeSource {
 /* Marks a point in time on this time source.
 The returned `[TimeMark]` instance encapsulates the captured time point and allows querying the duration of time interval `[elapsed]` `[TimeMark.elapsedNow]` from that point.
 public fun markNow(): TimeMark
 /* The most precise time source available in the platform.
 This time source returns its readings from a source of monotonic time when it is available in a target platform, and resorts to a non-monotonic time source otherwise.
 public object Monotonic : TimeSource by MonotonicTimeSource {
 override fun toString(): String = MonotonicTimeSource.toString()
 }
 public companion object {
 /* Represents a time point notched on a particular `[TimeSource]`. Remains bound to the time source it was taken from and allows querying for the duration of time elapsed from that point (see the function `[elapsedNow]`).
 @SinceKotlin("1.3")
 @ExperimentalTime
 public abstract class TimeMark {
 /* Returns the amount of time passed from this mark measured with the time source from which this mark was taken.
 Note that the value returned by this function can change on subsequent invocations.
 public abstract fun elapsedNow(): Duration
 /* Returns a time mark on the same time source that is ahead of this time mark by the specified `[duration]`.
 The returned time mark is more `_late_` when the `[duration]` is positive, and more `_early_` when the `[duration]` is negative.
 public open operator fun plus(duration: Duration): TimeMark = AdjustedTimeMark(this, duration)
 /* Returns a time mark on the same time source that is behind this time mark by the specified `[duration]`.
 The returned time mark is more `_early_` when the `[duration]` is positive, and more `_late_` when the `[duration]` is negative.
 public open operator fun minus(duration: Duration): TimeMark = plus(-duration)
 /* Returns true if this time mark has passed according to the time source from which this mark was taken.
 Note that the value returned by this function can change on subsequent invocations.
 If the time source is monotonic, it can change only from ``false`` to ``true``, namely, when the time mark becomes behind the current point of the time source.
 public fun hasPassedNow(): Boolean = !elapsedNow().isNegative()
 /* Returns false if this time mark has not passed according to the time source from which this mark was taken.
 Note that the value returned by this function can change on subsequent invocations.
 If the time source is monotonic, it can change only from ``true`` to ``false``, namely, when the time mark becomes behind the current point of the time source.
 public fun hasNotPassedNow(): Boolean = elapsedNow().isNegative()
 }
 @ExperimentalTime
 @SinceKotlin("1.3")
 @kotlin.internal.InlineOnly
 @Deprecated("Subtracting one TimeMark from another is not a well defined operation because these time marks could have been obtained from the different time sources.", level = DeprecationLevel.ERROR)
 @Suppress("UNUSED_PARAMETER")
 public inline operator fun TimeMark.minus(other: TimeMark): Duration = throw Error("Operation is")

```

disallowed.\n\n@ExperimentalTime\n@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly\n@Deprecated(\n
"Comparing one TimeMark to another is not a well defined operation because these time marks could have been
obtained from the different time sources.\",\n    level =
DeprecationLevel.ERROR\n)\n\n@Suppress("UNUSED_PARAMETER")\npublic inline operator fun
TimeMark.compareTo(other: TimeMark): Int = throw Error("Operation is
disallowed.\")\n\n\n@ExperimentalTime\nprivate class AdjustedTimeMark(val mark: TimeMark, val adjustment:
Duration) : TimeMark() {\n    override fun elapsedNow(): Duration = mark.elapsedNow() - adjustment\n\n
override fun plus(duration: Duration): TimeMark = AdjustedTimeMark(mark, adjustment + duration)\n}\n"/*\n *
Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is
governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage
kotlin.time\n\n@SinceKotlin("1.3")\n@ExperimentalTime\ninternal expect object MonotonicTimeSource :
TimeSource\n\n/**\n * An abstract class used to implement time sources that return their readings as [Long] values
in the specified [unit].\n * @property unit The unit in which this time source's readings are expressed.\n
*/\n\n@SinceKotlin("1.3")\n@ExperimentalTime\npublic abstract class AbstractLongTimeSource(protected val
unit: DurationUnit) : TimeSource {\n    /**\n     * This protected method should be overridden to return the current
reading of the time source expressed as a [Long] number\n     * in the unit specified by the [unit] property.\n
*/\n\n    protected abstract fun read(): Long\n\n    private class LongTimeMark(private val startedAt: Long, private val
timeSource: AbstractLongTimeSource, private val offset: Duration) : TimeMark() {\n        override fun
elapsedNow(): Duration = (timeSource.read() - startedAt).toDuration(timeSource.unit) - offset\n        override fun
plus(duration: Duration): TimeMark = LongTimeMark(startedAt, timeSource, offset + duration)\n    }\n\n    override
fun markNow(): TimeMark = LongTimeMark(read(), this, Duration.ZERO)\n}\n\n/**\n * An abstract class used to
implement time sources that return their readings as [Double] values in the specified [unit].\n * @property unit
The unit in which this time source's readings are expressed.\n
*/\n\n@SinceKotlin("1.3")\n@ExperimentalTime\npublic abstract class AbstractDoubleTimeSource(protected val
unit: DurationUnit) : TimeSource {\n    /**\n     * This protected method should be overridden to return the current
reading of the time source expressed as a [Double] number\n     * in the unit specified by the [unit] property.\n
*/\n\n    protected abstract fun read(): Double\n\n    private class DoubleTimeMark(private val startedAt: Double,
private val timeSource: AbstractDoubleTimeSource, private val offset: Duration) : TimeMark() {\n        override fun
elapsedNow(): Duration = (timeSource.read() - startedAt).toDuration(timeSource.unit) - offset\n        override fun
plus(duration: Duration): TimeMark = DoubleTimeMark(startedAt, timeSource, offset + duration)\n    }\n\n    override
fun markNow(): TimeMark = DoubleTimeMark(read(), this, Duration.ZERO)\n}\n\n/**\n * A time source
that has programmatically updatable readings. It is useful as a predictable source of time in tests.\n * The current
reading value can be advanced by the specified duration amount with the operator [plusAssign]:\n * val
timeSource = TestTimeSource()\n * timeSource += 10.seconds\n * Implementation note: the current
reading value is stored as a [Long] number of nanoseconds,\n * thus it's capable to represent a time range of
approximately 1292 years.\n * Should the reading value overflow as the result of [plusAssign] operation, an
[IllegalStateException] is thrown.\n */\n\n@SinceKotlin("1.3")\n@ExperimentalTime\npublic class TestTimeSource
: AbstractLongTimeSource(unit = DurationUnit.NANOSECONDS) {\n    private var reading: Long = 0L\n\n    override fun
read(): Long = reading\n\n    /**\n     * Advances the current reading value of this time source by the
specified [duration].\n     * [duration] value is rounded down towards zero when converting it to a [Long]
number of nanoseconds.\n     * For example, if the duration being added is `0.6.nanoseconds`, the reading doesn't
advance because\n     * the duration value is rounded to zero nanoseconds.\n     */\n\n    @throws
IllegalStateException when the reading value overflows as the result of this operation.\n    public operator fun
plusAssign(duration: Duration) {\n        val longDelta = duration.toLong(unit)\n        reading = if (longDelta !=
Long.MIN_VALUE && longDelta != Long.MAX_VALUE) {\n            // when delta fits in long, add it as long\n
            val newReading = reading + longDelta\n            if (reading xor longDelta >= 0 && reading xor newReading < 0)\n                overflow(duration)\n            newReading\n        } else {\n            val delta = duration.toDouble(unit)\n            // when
delta is greater than long, add it as double\n            val newReading = reading + delta\n            if (newReading >

```

```

Long.MAX_VALUE || newReading < Long.MIN_VALUE) overflow(duration)\n      newReading.toLong()\n }\n }\n\n private fun overflow(duration: Duration) {\n      throw IllegalStateException("TestTimeSource will\n overflow if its reading ${reading}ns is advanced by $duration.")\n }\n }\n\n", /*\n * Copyright 2010-2020 JetBrains\n s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0\n license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.time\n\nimport\nkotlin.contracts.*\n\n/**\n * Executes the given function [block] and returns the duration of elapsed time interval.\n *\n * The elapsed time is measured with [TimeSource.Monotonic].\n *\n */\n\n@SinceKotlin("1.3")\n@ExperimentalTime\npublic inline fun measureTime(block: () -> Unit): Duration {\n\n    contract {\n        callsInPlace(block, InvocationKind.EXACTLY_ONCE)\n    }\n\n    return\n    TimeSource.Monotonic.measureTime(block)\n }\n\n", /*\n * Executes the given function [block] and returns the\n duration of elapsed time interval.\n *\n * The elapsed time is measured with the specified `this` [TimeSource]\n instance.\n *\n */\n\n@SinceKotlin("1.3")\n@ExperimentalTime\npublic inline fun TimeSource.measureTime(block: ()\n-> Unit): Duration {\n\n    contract {\n        callsInPlace(block, InvocationKind.EXACTLY_ONCE)\n    }\n\n    val\n    mark = markNow()\n    block()\n    return mark.elapsedNow()\n }\n\n", /*\n * Data class representing a result of\n executing an action, along with the duration of elapsed time interval.\n *\n * @property value the result of the\n action.\n * @property duration the time elapsed to execute the action.\n *\n */\n\n@SinceKotlin("1.3")\n@ExperimentalTime\npublic data class TimedValue<T>(val value: T, val duration:\nDuration)\n\n", /*\n * Executes the given function [block] and returns an instance of [TimedValue] class, containing\n both\n * the result of the function execution and the duration of elapsed time interval.\n *\n * The elapsed time is\n measured with [TimeSource.Monotonic].\n *\n */\n\n@SinceKotlin("1.3")\n@ExperimentalTime\npublic inline fun <T>\nmeasureTimedValue(block: () -> T): TimedValue<T> {\n\n    contract {\n        callsInPlace(block,\n    InvocationKind.EXACTLY_ONCE)\n    }\n\n    return\n    TimeSource.Monotonic.measureTimedValue(block)\n }\n\n", /*\n * Executes the given [block] and returns an\n instance of [TimedValue] class, containing both\n * the result of function execution and the duration of elapsed time\n interval.\n *\n * The elapsed time is measured with the specified `this` [TimeSource] instance.\n *\n */\n\n@SinceKotlin("1.3")\n@ExperimentalTime\npublic inline fun <T> TimeSource.measureTimedValue(block: ()\n-> T): TimedValue<T> {\n\n    contract {\n        callsInPlace(block, InvocationKind.EXACTLY_ONCE)\n    }\n\n    val\n    mark = markNow()\n    val result = block()\n    return TimedValue(result, mark.elapsedNow())\n }\n\n", /*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is\n governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage\nkotlin\n\nimport kotlin.coroutines.*\nimport kotlin.coroutines.intrinsics.*\nimport\nkotlin.native.concurrent.SharedImmutable\n\n", /*\n * Defines deep recursive function that keeps its stack on the\n heap,\n * which allows very deep recursive computations that do not use the actual call stack.\n * To initiate a call to\n this deep recursive function use its [invoke] function.\n * As a rule of thumb, it should be used if recursion goes\n deeper than a thousand calls.\n *\n * The [DeepRecursiveFunction] takes one parameter of type [T] and returns a\n result of type [R].\n * The [block] of code defines the body of a recursive function. In this block\n * [callRecursive][DeepRecursiveScope.callRecursive] function can be used to make a recursive call\n * to the\n declared function. Other instances of [DeepRecursiveFunction] can be called\n * in this scope with `callRecursive`\n extension, too.\n *\n * For example, take a look at the following recursive tree class and a deeply\n * recursive\n instance of this tree with 100K nodes:\n *\n * ```\n * class Tree(val left: Tree? = null, val right: Tree? = null)\n * val\n    deepTree = generateSequence(Tree()) { Tree(it) }.take(100_000).last()\n * ```\n *\n * A regular recursive function\n can be defined to compute a depth of a tree:\n *\n * ```\n * fun depth(t: Tree?): Int =\n    if (t == null) 0 else\n    max(depth(t.left), depth(t.right)) + 1\n * println(depth(deepTree)) // StackOverflowError\n * ```\n *\n * If this\n `depth` function is called for a `deepTree` it produces [StackOverflowError] because of deep recursion.\n *\n * However, the `depth` function can be rewritten using `DeepRecursiveFunction` in the following way, and then\n * it\n successfully computes [depth(deepTree)][DeepRecursiveFunction.invoke] expression:\n *\n * ```\n * val depth =\n    DeepRecursiveFunction<Tree?, Int> { t ->\n        if (t == null) 0 else max(callRecursive(t.left),\n    callRecursive(t.right)) + 1\n * }\n * println(depth(deepTree)) // Ok\n * ```\n *\n * Deep recursive functions can also

```

mutually call each other using a heap for the stack via `callRecursive` [DeepRecursiveScope.callRecursive] extension. For example, the following pair of mutually recursive functions computes the number of tree nodes at even depth in the tree.

```

val mutualRecursion = object {
    val even: DeepRecursiveFunction<Tree?, Int> = DeepRecursiveFunction { t ->
        if (t == null) 0 else
        odd.callRecursive(t.left) + odd.callRecursive(t.right) + 1
    }
    val odd: DeepRecursiveFunction<Tree?, Int> = DeepRecursiveFunction { t ->
        if (t == null) 0 else even.callRecursive(t.left) +
        even.callRecursive(t.right)
    }
}

```

the function parameter type. `@param [T]` the function parameter type. `@param [R]` the function result type. `@param block` the function body.

```

@SinceKotlin("1.4")
@ExperimentalStdlibApi
public class DeepRecursiveFunction<T, R> {
    internal val block: suspend DeepRecursiveScope<T, R>.(T) -> R
}

```

Initiates a call to this deep recursive function, forming a root of the call tree. This operator should not be used from inside of `DeepRecursiveScope` as it uses the call stack slot for initial recursive invocation. From inside of `DeepRecursiveScope` use `callRecursive` [DeepRecursiveScope.callRecursive].

```

@SinceKotlin("1.4")
@ExperimentalStdlibApi
public operator fun <T, R> DeepRecursiveFunction<T, R>.invoke(value: T): R =
    DeepRecursiveScopeImpl<T, R>(block, value).runCallLoop()

```

A scope class for `DeepRecursiveFunction` function declaration that defines `callRecursive` methods to recursively call this function or another `DeepRecursiveFunction` putting the call activation frame on the heap. `@param [T]` function parameter type. `@param [R]` function result type.

```

@RestrictsSuspension
@SinceKotlin("1.4")
@ExperimentalStdlibApi
public sealed class DeepRecursiveScope<T, R> {
    /**
     * Makes recursive call to this [DeepRecursiveFunction] function putting
     * the call activation frame on the heap, as opposed to the actual call stack that is used by a regular recursive
     * call.
     */
    public abstract suspend fun callRecursive(value: T): R

    /**
     * Makes call to the specified [DeepRecursiveFunction] function putting the call activation frame on the heap,
     * as opposed to the actual call stack that is used by a regular call.
     */
    public abstract suspend fun <U, S> DeepRecursiveFunction<U, S>.callRecursive(value: U): S

    @Deprecated(
        level = DeprecationLevel.ERROR,
        message = "'invoke' should not be called from DeepRecursiveScope. Use 'callRecursive' to do recursion in the heap instead of the call stack."
    )
    replaceWith = ReplaceWith("this.callRecursive(value)")
    @Suppress("UNUSED_PARAMETER")
    public operator fun DeepRecursiveFunction<*, *>.invoke(value: Any?): Nothing =
        throw UnsupportedOperationException("Should not be called from DeepRecursiveScope")
}

```

Implementation

```

@ExperimentalStdlibApi
private typealias DeepRecursiveFunctionBlock = suspend DeepRecursiveScope<*, *>.(Any?) -> Any?
@SharedImmutable
private val UNDEFINED_RESULT = Result.success(COROUTINE_SUSPENDED)
@ExperimentalStdlibApi
private class DeepRecursiveScopeImpl<T, R> {
    block: suspend DeepRecursiveScope<T, R>.(T) -> R,
    value: T
} : DeepRecursiveScope<T, R> {
    Continuation<R> {
        // Active function block
        private var function: DeepRecursiveFunctionBlock = block as DeepRecursiveFunctionBlock
        // Value to call function with
        private var value: Any? = value
        // Continuation of the current call
        private var cont: Continuation<Any?>? = this as Continuation<Any?>
        // Completion result (completion of the whole call stack)
        private var result: Result<Any?> = UNDEFINED_RESULT
        override val context: CoroutineContext {
            get() = EmptyCoroutineContext
        }
        override fun resumeWith(result: Result<R>) {
            this.cont = null
            this.result = result
        }
        override suspend fun callRecursive(value: T): R =
            suspendCoroutineUninterceptedOrReturn { cont ->
                // calling the same function that is currently active
                this.cont = cont as Continuation<Any?>
                this.value = value
                COROUTINE_SUSPENDED
            }
        override suspend fun <U, S> DeepRecursiveFunction<U, S>.callRecursive(value: U): S =
            suspendCoroutineUninterceptedOrReturn { cont ->
                // calling another recursive function
                val function = block as DeepRecursiveFunctionBlock
                with(this@DeepRecursiveScopeImpl) {
                    val currentFunction = this.function
                    if (function !== currentFunction) {
                        // calling a different function -- create a trampoline to restore function ref
                        this.function = function
                        this.cont =

```

```

crossFunctionCompletion(currentFunction, cont as Continuation<Any?>)\n      } else {\n          // calling the
same function -- direct\n          this.cont = cont as Continuation<Any?>\n      }\n      this.value = value\n  }\n  COROUTINE_SUSPENDED\n  }\n\n  private fun crossFunctionCompletion(\n      currentFunction:
DeepRecursiveFunctionBlock,\n      cont: Continuation<Any?>\n  ): Continuation<Any?> =
Continuation(EmptyCoroutineContext) {\n      this.function = currentFunction\n      // When going back from a
trampoline we cannot just call cont.resume (stack usage!)\n      // We delegate the cont.resumeWith(it) call to
runCallLoop\n      this.cont = cont\n      this.result = it\n  }\n\n  @Suppress(\"UNCHECKED_CAST\")\n  fun
runCallLoop(): R {\n      while (true) {\n          // Note: cont is set to null in DeepRecursiveScopeImpl.resumeWith
when the whole computation completes\n          val result = this.result\n          val cont = this.cont\n          ?:
return (result as Result<R>).getOrThrow() // done -- final result\n          // The order of comparison is important
here for that case of rogue class with broken equals\n          if (UNDEFINED_RESULT == result) {\n          //
call \"function\" with \"value\" using \"cont\" as completion\n          val r = try {\n          // This is
block.startCoroutine(this, value, cont)\n          function.startCoroutineUninterceptedOrReturn(this, value,
cont)\n          } catch (e: Throwable) {\n          cont.resumeWithException(e)\n          continue\n
          }\n          // If the function returns without suspension -- calls its continuation immediately\n          if (r !==
COROUTINE_SUSPENDED)\n          cont.resume(r as R)\n          } else {\n          // we returned from a
crossFunctionCompletion trampoline -- call resume here\n          this.result = UNDEFINED_RESULT // reset
result back\n          cont.resumeWith(result)\n          }\n      }\n  }\n}\n\n} * Copyright 2010-2022
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n * Auto-generated file. DO NOT
EDIT!\n\n@file:kotlin.jvm.JvmName(\"NumbersKt\")\n@file:kotlin.jvm.JvmMultifileClass\npackage
kotlin\n\nimport kotlin.math.sign\n\n/** Divides this value by the other value, flooring the result to an integer that is
closer to negative infinity. *\n\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\npublic inline fun
Byte.floorDiv(other: Byte): Int = \n      this.toInt().floorDiv(other.toInt())\n\n/**\n * Calculates the remainder of
flooring division of this value by the other value.\n * \n * The result is either zero or has the same sign as the
_divisor_ and has the absolute value less than the absolute value of the divisor.\n\n
*\n\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\npublic inline fun Byte.mod(other: Byte): Byte = \n
this.toInt().mod(other.toInt()).toByte()\n\n/** Divides this value by the other value, flooring the result to an integer
that is closer to negative infinity. *\n\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\npublic inline fun
Byte.floorDiv(other: Short): Int = \n      this.toInt().floorDiv(other.toInt())\n\n/**\n * Calculates the remainder of
flooring division of this value by the other value.\n * \n * The result is either zero or has the same sign as the
_divisor_ and has the absolute value less than the absolute value of the divisor.\n\n
*\n\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\npublic inline fun Byte.mod(other: Short): Short = \n
this.toInt().mod(other.toInt()).toShort()\n\n/** Divides this value by the other value, flooring the result to an integer
that is closer to negative infinity. *\n\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\npublic inline fun
Byte.floorDiv(other: Int): Int = \n      this.toInt().floorDiv(other)\n\n/**\n * Calculates the remainder of flooring
division of this value by the other value.\n * \n * The result is either zero or has the same sign as the _divisor_ and
has the absolute value less than the absolute value of the divisor.\n\n
*\n\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\npublic inline fun Byte.mod(other: Int): Int = \n
this.toInt().mod(other)\n\n/** Divides this value by the other value, flooring the result to an integer that is closer to
negative infinity. *\n\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\npublic inline fun Byte.floorDiv(other:
Long): Long = \n      this.toLong().floorDiv(other)\n\n/**\n * Calculates the remainder of flooring division of this
value by the other value.\n * \n * The result is either zero or has the same sign as the _divisor_ and has the absolute
value less than the absolute value of the divisor.\n\n
*\n\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\npublic
inline fun Byte.mod(other: Long): Long = \n      this.toLong().mod(other)\n\n/** Divides this value by the other
value, flooring the result to an integer that is closer to negative infinity. *\n\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\npublic inline fun Short.floorDiv(other: Byte): Int = \n
this.toInt().floorDiv(other.toInt())\n\n/**\n * Calculates the remainder of flooring division of this value by the other

```

value.
The result is either zero or has the same sign as the `_divisor_` and has the absolute value less than the absolute value of the divisor.
`@SinceKotlin("1.5")@kotlin.internal.InlineOnly\npublic inline fun Short.mod(other: Byte): Byte = this.toInt().mod(other.toInt()).toByte()`
Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.

`@SinceKotlin("1.5")@kotlin.internal.InlineOnly\npublic inline fun Short.floorDiv(other: Short): Int = this.toInt().floorDiv(other.toInt())`
Calculates the remainder of flooring division of this value by the other value.
The result is either zero or has the same sign as the `_divisor_` and has the absolute value less than the absolute value of the divisor.
`@SinceKotlin("1.5")@kotlin.internal.InlineOnly\npublic inline fun Short.mod(other: Short): Short = this.toInt().mod(other.toInt()).toShort()`
Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.

`@SinceKotlin("1.5")@kotlin.internal.InlineOnly\npublic inline fun Short.floorDiv(other: Int): Int = this.toInt().floorDiv(other)`
Calculates the remainder of flooring division of this value by the other value.
The result is either zero or has the same sign as the `_divisor_` and has the absolute value less than the absolute value of the divisor.
`@SinceKotlin("1.5")@kotlin.internal.InlineOnly\npublic inline fun Short.mod(other: Int): Int = this.toInt().mod(other)`
Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.

`@SinceKotlin("1.5")@kotlin.internal.InlineOnly\npublic inline fun Short.floorDiv(other: Long): Long = this.toLong().floorDiv(other)`
Calculates the remainder of flooring division of this value by the other value.
The result is either zero or has the same sign as the `_divisor_` and has the absolute value less than the absolute value of the divisor.
`@SinceKotlin("1.5")@kotlin.internal.InlineOnly\npublic inline fun Short.mod(other: Long): Long = this.toLong().mod(other)`
Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.

`@SinceKotlin("1.5")@kotlin.internal.InlineOnly\npublic inline fun Int.floorDiv(other: Byte): Int = this.floorDiv(other.toInt())`
Calculates the remainder of flooring division of this value by the other value.
The result is either zero or has the same sign as the `_divisor_` and has the absolute value less than the absolute value of the divisor.
`@SinceKotlin("1.5")@kotlin.internal.InlineOnly\npublic inline fun Int.mod(other: Byte): Byte = this.mod(other.toInt()).toByte()`
Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.

`@SinceKotlin("1.5")@kotlin.internal.InlineOnly\npublic inline fun Int.floorDiv(other: Short): Int = this.floorDiv(other.toInt())`
Calculates the remainder of flooring division of this value by the other value.
The result is either zero or has the same sign as the `_divisor_` and has the absolute value less than the absolute value of the divisor.
`@SinceKotlin("1.5")@kotlin.internal.InlineOnly\npublic inline fun Int.mod(other: Short): Short = this.mod(other.toInt()).toShort()`
Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.

`@SinceKotlin("1.5")@kotlin.internal.InlineOnly\npublic inline fun Int.floorDiv(other: Int): Int { val q = this / other\n if (this xor other < 0 && q * other != this) q--\n return q}`
Calculates the remainder of flooring division of this value by the other value.
The result is either zero or has the same sign as the `_divisor_` and has the absolute value less than the absolute value of the divisor.

`@SinceKotlin("1.5")@kotlin.internal.InlineOnly\npublic inline fun Int.mod(other: Int): Int { val r = this % other\n return r + (other and (((r xor other) and (r or -r)) shr 31))}`
Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.

`@SinceKotlin("1.5")@kotlin.internal.InlineOnly\npublic inline fun Int.floorDiv(other: Long): Long = this.toLong().floorDiv(other)`
Calculates the remainder of flooring division of this value by the other value.
The result is either zero or has the same sign as the `_divisor_` and has the absolute value less than the absolute value of the divisor.
`@SinceKotlin("1.5")@kotlin.internal.InlineOnly\npublic inline fun Int.mod(other: Long): Long = this.toLong().mod(other)`
Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.

`@SinceKotlin("1.5")@kotlin.internal.InlineOnly\npublic inline fun Long.floorDiv(other: Byte): Long =`

```

this.floorDiv(other.toLong())\n\n/**\n * Calculates the remainder of flooring division of this value by the other
value.\n * \n * The result is either zero or has the same sign as the _divisor_ and has the absolute value less than the
absolute value of the divisor.\n */\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun
Long.mod(other: Byte): Byte = \n    this.mod(other.toLong()).toByte()\n\n/** Divides this value by the other value,
flooring the result to an integer that is closer to negative infinity.
*/\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun Long.floorDiv(other: Short): Long = \n
this.floorDiv(other.toLong())\n\n/**\n * Calculates the remainder of flooring division of this value by the other
value.\n * \n * The result is either zero or has the same sign as the _divisor_ and has the absolute value less than the
absolute value of the divisor.\n */\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun
Long.mod(other: Short): Short = \n    this.mod(other.toLong()).toShort()\n\n/** Divides this value by the other
value, flooring the result to an integer that is closer to negative infinity.
*/\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun Long.floorDiv(other: Int): Long = \n
this.floorDiv(other.toLong())\n\n/**\n * Calculates the remainder of flooring division of this value by the other
value.\n * \n * The result is either zero or has the same sign as the _divisor_ and has the absolute value less than the
absolute value of the divisor.\n */\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun
Long.mod(other: Int): Int = \n    this.mod(other.toLong()).toInt()\n\n/** Divides this value by the other value,
flooring the result to an integer that is closer to negative infinity.
*/\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun Long.floorDiv(other: Long): Long {\n
var q = this / other\n    if (this xor other < 0 && q * other != this) q--\n    return q\n}\n\n/**\n * Calculates the
remainder of flooring division of this value by the other value.\n * \n * The result is either zero or has the same sign
as the _divisor_ and has the absolute value less than the absolute value of the divisor.\n
*/\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun Long.mod(other: Long): Long {\n    val r
= this % other\n    return r + (other and (((r xor other) and (r or -r)) shr 63))\n}\n\n/**\n * Calculates the remainder
of flooring division of this value by the other value.\n * \n * The result is either zero or has the same sign as the
_divisor_ and has the absolute value less than the absolute value of the divisor.\n * \n * If the result cannot be
represented exactly, it is rounded to the nearest representable number. In this case the absolute value of the result
can be less than or _equal to_ the absolute value of the divisor.\n
*/\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun Float.mod(other: Float): Float {\n    val r
= this % other\n    return if (r != 0.0.toFloat() && r.sign != other.sign) r + other else r\n}\n\n/**\n * Calculates the
remainder of flooring division of this value by the other value.\n * \n * The result is either zero or has the same sign
as the _divisor_ and has the absolute value less than the absolute value of the divisor.\n * \n * If the result cannot be
represented exactly, it is rounded to the nearest representable number. In this case the absolute value of the result
can be less than or _equal to_ the absolute value of the divisor.\n
*/\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun Float.mod(other: Double): Double = \n
this.toDouble().mod(other)\n\n/**\n * Calculates the remainder of flooring division of this value by the other
value.\n * \n * The result is either zero or has the same sign as the _divisor_ and has the absolute value less than the
absolute value of the divisor.\n * \n * If the result cannot be represented exactly, it is rounded to the nearest
representable number. In this case the absolute value of the result can be less than or _equal to_ the absolute value of
the divisor.\n */\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun Double.mod(other: Float):
Double = \n    this.mod(other.toDouble())\n\n/**\n * Calculates the remainder of flooring division of this value by
the other value.\n * \n * The result is either zero or has the same sign as the _divisor_ and has the absolute value less
than the absolute value of the divisor.\n * \n * If the result cannot be represented exactly, it is rounded to the nearest
representable number. In this case the absolute value of the result can be less than or _equal to_ the absolute value of
the divisor.\n */\n@SinceKotlin("1.5")\n@kotlin.internal.InlineOnly\npublic inline fun Double.mod(other:
Double): Double {\n    val r = this % other\n    return if (r != 0.0 && r.sign != other.sign) r + other else
r\n}\n\n"/>\n\n/* Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of
this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\npackage kotlin\n\nimport kotlin.internal.InlineOnly\n\n/**\n * Returns a hash code value for the object or

```

```

zero if the object is `null`.  

@see Any.hashCode  

@SinceKotlin("1.3")  

@InlineOnly  

public inline fun Any?.hashCode(): Int = this?.hashCode() ?: 0  

"/**  

 * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.  

 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.  

 */  

package kotlin  

/**  

 * Represents a version of the Kotlin standard library.  

 * [major], [minor] and [patch] are integer components of a version,  

 * they must be non-negative and not greater than 255 ([MAX_COMPONENT_VALUE]).  

 * @constructor Creates a version from all three components.  

 * @SinceKotlin("1.1")  

public class KotlinVersion(val major: Int, val minor: Int, val patch: Int) : Comparable<KotlinVersion> {  

    /**  

     * Creates a version from [major] and [minor] components, leaving [patch] component zero.  

     */  

    public constructor(major: Int, minor: Int) : this(major, minor, 0)  

    private val version = versionOf(major, minor, patch)  

    private fun versionOf(major: Int, minor: Int, patch: Int): Int {  

        require(major in 0..MAX_COMPONENT_VALUE && minor in 0..MAX_COMPONENT_VALUE && patch in 0..MAX_COMPONENT_VALUE) {  

            "Version components are out of range: $major.$minor.$patch"  

        }  

        return major.shl(16) + minor.shl(8) + patch  

    }  

    /**  

     * Returns the string representation of this version.  

     */  

    override fun toString(): String = "$major.$minor.$patch"  

    override fun equals(other: Any?): Boolean {  

        if (this === other) return true  

        val otherVersion = (other as? KotlinVersion) ?: return false  

        return this.version == otherVersion.version  

    }  

    override fun hashCode(): Int = version  

    override fun compareTo(other: KotlinVersion): Int = version - other.version  

    /**  

     * Returns `true` if this version is not less than the version specified  

     * with the provided [major] and [minor] components.  

     */  

    public fun isAtLeast(major: Int, minor: Int): Boolean = // this.version >= versionOf(major, minor, 0)  

        this.major > major || (this.major == major && this.minor >= minor)  

    /**  

     * Returns `true` if this version is not less than the version specified  

     * with the provided [major], [minor] and [patch] components.  

     */  

    public fun isAtLeast(major: Int, minor: Int, patch: Int): Boolean = // this.version >= versionOf(major, minor, patch)  

        this.major > major || (this.major == major && (this.minor > minor || this.minor == minor && this.patch >= patch))  

    companion object {  

        /**  

         * Maximum value a version component can have, a constant value 255.  

         */  

        // NOTE: Must be placed before CURRENT because its initialization requires this field being initialized in JS  

        public const val MAX_COMPONENT_VALUE = 255  

        /**  

         * Returns the current version of the Kotlin standard library.  

         */  

        @kotlin.jvm.JvmField  

        public val CURRENT: KotlinVersion = KotlinVersionCurrentValue.get()  

    }  

} // this class is ignored during classpath normalization when considering whether to recompile dependencies in Kotlin build  

private object KotlinVersionCurrentValue {  

    @kotlin.jvm.JvmStatic  

    fun get(): KotlinVersion = KotlinVersion(1, 6, 21) // value is written here automatically during build  

}"/**  

 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.  

 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.  

 */  

@file:kotlin.jvm.JvmName("LateinitKt")  

@file:Suppress("unused")  

package kotlin  

import kotlin.internal.InlineOnly  

import kotlin.internal.AccessibleLateinitPropertyLiteral  

import kotlin.reflect.KProperty0  

/**  

 * Returns `true` if this lateinit property has been assigned a value, and `false` otherwise.  

 * Cannot be used in an inline function, to avoid binary compatibility issues.  

 */  

@SinceKotlin("1.2")  

@InlineOnly  

inline val @receiver: AccessibleLateinitPropertyLiteral KProperty0<*>.isInitialized: Boolean  

    get() = throw NotImplementedError("Implementation is intrinsic")  

"/**  

 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.  

 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.  

 */  

@file:kotlin.jvm.JvmName("LazyKt")  

@file:kotlin.jvm.JvmMultifileClass  

package kotlin  

import kotlin.reflect.KProperty  

/**  

 * Represents a value with lazy initialization.  

 * To create an instance of [Lazy] use the [lazy] function.  

 */  

public interface Lazy<out T> {  

    /**  

     * Gets the lazily initialized value of the current Lazy instance.  

     * Once the value was initialized it must not change during the rest of lifetime of this Lazy instance.  

     */  

    public val value: T  

    /**  

     * Returns `true` if a value for this Lazy instance has been already initialized, and `false` otherwise.  

     * Once this function has returned `true` it stays `true` for the rest of lifetime of this Lazy instance.  

     */  

    public fun isInitialized(): Boolean  

} /**  

 * Creates a new instance of

```

```

the [Lazy] that is already initialized with the specified [value].\n *^\npublic fun <T> lazyOf(value: T): Lazy<T> =
InitializedLazyImpl(value)\n\n/**\n * An extension to delegate a read-only property of type [T] to an instance of
[Lazy].\n * This extension allows to use instances of Lazy for property delegation:\n * `val property: String by
lazy { initializer }`\n *^\n@kotlin.internal.InlineOnly\npublic inline operator fun <T> Lazy<T>.getValue(thisRef:
Any?, property: KProperty<*>): T = value\n\n/**\n * Specifies how a [Lazy] instance synchronizes initialization
among multiple threads.\n *^\npublic enum class LazyThreadSafetyMode {\n\n    /**\n     * Locks are used to ensure
that only a single thread can initialize the [Lazy] instance.\n     *^\n    SYNCHRONIZED,\n\n    /**\n     * Initializer
function can be called several times on concurrent access to uninitialized [Lazy] instance value,\n     * but only the
first returned value will be used as the value of [Lazy] instance.\n     *^\n    PUBLICATION,\n\n    /**\n     * No
locks are used to synchronize an access to the [Lazy] instance value; if the instance is accessed from multiple
threads, its behavior is undefined.\n     *\n     * This mode should not be used unless the [Lazy] instance is
guaranteed never to be initialized from more than one thread.\n     *^\n    NONE,\n}\n\ninternal object
UNINITIALIZED_VALUE\n\n// internal to be called from lazy in JS\ninternal class UnsafeLazyImpl<out
T>(initializer: () -> T) : Lazy<T>, Serializable {\n    private var initializer: (() -> T)? = initializer\n    private var
_value: Any? = UNINITIALIZED_VALUE\n\n    override val value: T\n        get() {\n            if (_value ===
UNINITIALIZED_VALUE) {\n                _value = initializer!!()\n                initializer = null\n            }\n
@Suppress("UNCHECKED_CAST")\n            return _value as T\n        }\n\n    override fun isInitialized():
Boolean = _value !== UNINITIALIZED_VALUE\n\n    override fun toString(): String = if (isInitialized())
value.toString() else "Lazy value not initialized yet.\n\n    private fun writeReplace(): Any =
InitializedLazyImpl(value)\n}\n\ninternal class InitializedLazyImpl<out T>(override val value: T) : Lazy<T>,
Serializable {\n    override fun isInitialized(): Boolean = true\n\n    override fun toString(): String =
value.toString()\n}\n\n"/**\n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n
*\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("NumbersKt")\npackage kotlin\n\n/**\n *
Counts the number of set bits in the binary representation of this [Int] number.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
Int.countOneBits(): Int\n\n/**\n * Counts the number of consecutive most significant bits that are zero in the binary
representation of this [Int] number.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
Int.countLeadingZeroBits(): Int\n\n/**\n * Counts the number of consecutive least significant bits that are zero in
the binary representation of this [Int] number.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
Int.countTrailingZeroBits(): Int\n\n/**\n * Returns a number having a single bit set in the position of the most
significant set bit of this [Int] number,\n * or zero, if this number is zero.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
Int.takeHighestOneBit(): Int\n\n/**\n * Returns a number having a single bit set in the position of the least
significant set bit of this [Int] number,\n * or zero, if this number is zero.\n
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
Int.takeLowestOneBit(): Int\n\n/**\n * Rotates the binary representation of this [Int] number left by the specified
[bitCount] number of bits.\n * The most significant bits pushed out from the left side reenter the number as the least
significant bits on the right side.\n * Rotating the number left by a negative bit count is the same as rotating it
right by the negated bit count:\n * `number.rotateLeft(-n) == number.rotateRight(n)`\n * Rotating by a multiple
of [Int.SIZE_BITS] (32) returns the same number, or more generally\n * `number.rotateLeft(n) ==
number.rotateLeft(n % 32)`\n
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun
Int.rotateLeft(bitCount: Int): Int\n\n/**\n * Rotates the binary representation of this [Int] number right by the
specified [bitCount] number of bits.\n * The least significant bits pushed out from the right side reenter the number

```

as the most significant bits on the left side.\n * Rotating the number right by a negative bit count is the same as rotating it left by the negated bit count:\n * `number.rotateRight(-n) == number.rotateLeft(n)`\n * Rotating by a multiple of [Int.SIZE_BITS] (32) returns the same number, or more generally\n * `number.rotateRight(n) == number.rotateRight(n % 32)`\n

*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun Int.rotateRight(bitCount: Int): Int\n\n**\n * Counts the number of set bits in the binary representation of this [Long] number.\n * \n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun Long.countOneBits(): Int\n\n**\n * Counts the number of consecutive most significant bits that are zero in the binary representation of this [Long] number.\n

*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun Long.countLeadingZeroBits(): Int\n\n**\n * Counts the number of consecutive least significant bits that are zero in the binary representation of this [Long] number.\n

*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun Long.countTrailingZeroBits(): Int\n\n**\n * Returns a number having a single bit set in the position of the most significant set bit of this [Long] number,\n * or zero, if this number is zero.\n

*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun Long.takeHighestOneBit(): Long\n\n**\n * Returns a number having a single bit set in the position of the least significant set bit of this [Long] number,\n * or zero, if this number is zero.\n

*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun Long.takeLowestOneBit(): Long\n\n**\n * Rotates the binary representation of this [Long] number left by the specified [bitCount] number of bits.\n * The most significant bits pushed out from the left side reenter the number as the least significant bits on the right side.\n * Rotating the number left by a negative bit count is the same as rotating it right by the negated bit count:\n * `number.rotateLeft(-n) == number.rotateRight(n)`\n * Rotating by a multiple of [Long.SIZE_BITS] (64) returns the same number, or more generally\n * `number.rotateLeft(n) == number.rotateLeft(n % 64)`\n

*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun Long.rotateLeft(bitCount: Int): Long\n\n**\n * Rotates the binary representation of this [Long] number right by the specified [bitCount] number of bits.\n * The least significant bits pushed out from the right side reenter the number as the most significant bits on the left side.\n * Rotating the number right by a negative bit count is the same as rotating it left by the negated bit count:\n * `number.rotateRight(-n) == number.rotateLeft(n)`\n * Rotating by a multiple of [Long.SIZE_BITS] (64) returns the same number, or more generally\n * `number.rotateRight(n) == number.rotateRight(n % 64)`\n

*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun Long.rotateRight(bitCount: Int): Long\n\n**\n * Counts the number of set bits in the binary representation of this [Byte] number.\n

*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun Byte.countOneBits(): Int = (toInt() and 0xFF).countOneBits()\n\n**\n * Counts the number of consecutive most significant bits that are zero in the binary representation of this [Byte] number.\n

*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun Byte.countLeadingZeroBits(): Int = (toInt() and 0xFF).countLeadingZeroBits() - (Int.SIZE_BITS - Byte.SIZE_BITS)\n\n**\n * Counts the number of consecutive least significant bits that are zero in the binary representation of this [Byte] number.\n

*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun Byte.countTrailingZeroBits(): Int = (toInt() or 0x100).countTrailingZeroBits()\n\n**\n * Returns a number having a single bit set in the position of the most significant set bit of this [Byte] number,\n * or zero, if this number is zero.\n

*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun Byte.takeHighestOneBit(): Byte = (toInt() and 0xFF).takeHighestOneBit().toByte()\n\n**\n * Returns a

number having a single bit set in the position of the least significant set bit of this [Byte] number, or zero, if this number is zero.

```
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun Byte.takeLowestOneBit(): Byte = toInt().takeLowestOneBit().toByte()\n\n/**\n * Rotates the binary representation of this [Byte] number left by the specified [bitCount] number of bits.\n * The most significant bits pushed out from the left side reenter the number as the least significant bits on the right side.\n * Rotating the number left by a negative bit count is the same as rotating it right by the negated bit count:\n * `number.rotateLeft(-n) == number.rotateRight(n)`\n * Rotating by a multiple of [Byte.SIZE_BITS] (8) returns the same number, or more generally\n * `number.rotateLeft(n) == number.rotateLeft(n % 8)`\n
```

```
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun Byte.rotateLeft(bitCount: Int): Byte =\n    (toInt().shl(bitCount and 7) or (toInt() and 0xFF).ushr(8 - (bitCount and 7))).toByte()\n\n/**\n * Rotates the binary representation of this [Byte] number right by the specified [bitCount] number of bits.\n * The least significant bits pushed out from the right side reenter the number as the most significant bits on the left side.\n * Rotating the number right by a negative bit count is the same as rotating it left by the negated bit count:\n * `number.rotateRight(-n) == number.rotateLeft(n)`\n * Rotating by a multiple of [Byte.SIZE_BITS] (8) returns the same number, or more generally\n * `number.rotateRight(n) == number.rotateRight(n % 8)`\n
```

```
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun Byte.rotateRight(bitCount: Int): Byte =\n    (toInt().shl(8 - (bitCount and 7)) or (toInt() and 0xFF).ushr(bitCount and 7)).toByte()\n\n/**\n * Counts the number of set bits in the binary representation of this [Short] number.\n
```

```
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun Short.countOneBits(): Int = (toInt() and 0xFFFF).countOneBits()\n\n/**\n * Counts the number of consecutive most significant bits that are zero in the binary representation of this [Short] number.\n
```

```
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun Short.countLeadingZeroBits(): Int =\n    (toInt() and 0xFFFF).countLeadingZeroBits() - (Int.SIZE_BITS - Short.SIZE_BITS)\n\n/**\n * Counts the number of consecutive least significant bits that are zero in the binary representation of this [Short] number.\n
```

```
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun Short.countTrailingZeroBits(): Int = (toInt() or 0x10000).countTrailingZeroBits()\n\n/**\n * Returns a number having a single bit set in the position of the most significant set bit of this [Short] number, or zero, if this number is zero.\n
```

```
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun Short.takeHighestOneBit(): Short = (toInt() and 0xFFFF).takeHighestOneBit().toShort()\n\n/**\n * Returns a number having a single bit set in the position of the least significant set bit of this [Short] number, or zero, if this number is zero.\n
```

```
*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun Short.takeLowestOneBit(): Short = toInt().takeLowestOneBit().toShort()\n\n/**\n * Rotates the binary representation of this [Short] number left by the specified [bitCount] number of bits.\n * The most significant bits pushed out from the left side reenter the number as the least significant bits on the right side.\n * Rotating the number left by a negative bit count is the same as rotating it right by the negated bit count:\n * `number.rotateLeft(-n) == number.rotateRight(n)`\n * Rotating by a multiple of [Short.SIZE_BITS] (16) returns the same number, or more generally\n * `number.rotateLeft(n) == number.rotateLeft(n % 16)`\n
```

```
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun Short.rotateLeft(bitCount: Int): Short =\n    (toInt().shl(bitCount and 15) or (toInt() and 0xFFFF).ushr(16 - (bitCount and 15))).toShort()\n\n/**\n * Rotates the binary representation of this [Short] number right by the specified [bitCount] number of bits.\n * The least significant bits pushed out from the right side reenter the number as the most significant bits on the left side.\n * Rotating the number right by a negative bit count is the same as rotating it left by the negated bit count:\n * `number.rotateRight(-n) == number.rotateLeft(n)`\n * Rotating by a multiple
```

of [Short.SIZE_BITS] (16) returns the same number, or more generally
number.rotateRight(n) ==
number.rotateRight(n % 16)

```
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun  
Short.rotateRight(bitCount: Int): Short =\n    (toInt().shl(16 - (bitCount and 15)) or (toInt() and  
0xFFFF).ushr(bitCount and 15)).toShort()\n"/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming  
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the  
license/LICENSE.txt file.\n */\n\npackage kotlin\nimport kotlin.internal.RequireKotlin\nimport  
kotlin.internal.RequireKotlinVersionKind\n@kotlin.internal.InlineOnly\n@SinceKotlin("1.2")\n@Suppress("IN  
VISIBLE_MEMBER", "INVISIBLE_REFERENCE")\n@RequireKotlin("1.2.30", level =  
DeprecationLevel.HIDDEN, versionKind = RequireKotlinVersionKind.COMPILER_VERSION)\npublic inline fun  
<R> suspend(noinline block: suspend () -> R): suspend () -> R = block\n"/*\n * Copyright 2010-2018 JetBrains  
s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0  
license that can be found in the license/LICENSE.txt file.\n
```

```
*\n\n@file:kotlin.jvm.JvmName("TuplesKt")\n\npackage kotlin\n\n/**\n * Represents a generic pair of two  
values.\n * There is no meaning attached to values in this class, it can be used for any purpose.\n * Pair exhibits  
value semantics, i.e. two pairs are equal if both components are equal.\n * An example of decomposing it into  
values:\n * @sample samples.misc.Tuples.pairDestructuring\n * @param A type of the first value.\n * @param  
B type of the second value.\n * @property first First value.\n * @property second Second value.\n * @constructor  
Creates a new instance of Pair.\n */\npublic data class Pair<out A, out B>(\n    public val first: A,\n    public val  
second: B\n) : Serializable {\n    /**\n     * Returns string representation of the [Pair] including its [first] and  
[second] values.\n     */\n    public override fun toString(): String = "$first, $second"\n}\n\n/**\n * Creates a tuple  
of type [Pair] from this and [that].\n * This can be useful for creating [Map] literals with less noise, for  
example:\n * @sample samples.collections.Maps.instantiation.mapFromPairs\n */\npublic infix fun <A, B>  
A.to(that: B): Pair<A, B> = Pair(this, that)\n\n/**\n * Converts this pair into a list.\n * @sample  
samples.misc.Tuples.pairToList\n */\npublic fun <T> Pair<T, T>.toList(): List<T> = listOf(first, second)\n\n/**\n * Represents a triad of values.\n * There is no meaning attached to values in this class, it can be used for any  
purpose.\n * Triple exhibits value semantics, i.e. two triples are equal if all three components are equal.\n * An  
example of decomposing it into values:\n * @sample samples.misc.Tuples.tripleDestructuring\n * @param A  
type of the first value.\n * @param B type of the second value.\n * @param C type of the third value.\n * @property  
first First value.\n * @property second Second value.\n * @property third Third value.\n */\npublic data class  
Triple<out A, out B, out C>(\n    public val first: A,\n    public val second: B,\n    public val third: C\n) : Serializable  
{\n    /**\n     * Returns string representation of the [Triple] including its [first], [second] and [third] values.\n     */\n    public override fun toString(): String = "$first, $second, $third"\n}\n\n/**\n * Converts this triple into a  
list.\n * @sample samples.misc.Tuples.tripleToList\n */\npublic fun <T> Triple<T, T, T>.toList(): List<T> =  
listOf(first, second, third)\n"/*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language  
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the  
license/LICENSE.txt file.\n */\n\n// Auto-generated file. DO NOT EDIT!\n\npackage kotlin.ranges\n\nimport  
kotlin.internal.*\n\n/**\n * A range of values of type `UInt`.\n
```

```
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic class UIntRange(start:  
UInt, endInclusive: UInt) : UIntProgression(start, endInclusive, 1), ClosedRange<UInt> {\n    override val start:  
UInt get() = first\n    override val endInclusive: UInt get() = last\n\n    override fun contains(value: UInt): Boolean =  
first <= value && value <= last\n\n    /**\n     * Checks if the range is empty.\n     * The range is empty if its  
start value is greater than the end value.\n     */\n    override fun isEmpty(): Boolean = first > last\n\n    override fun  
equals(other: Any?): Boolean =\n        other is UIntRange && (isEmpty() && other.isEmpty()) ||\n        first ==  
other.first && last == other.last\n\n    override fun hashCode(): Int =\n        if (isEmpty()) -1 else (31 * first.toInt()  
+ last.toInt())\n\n    override fun toString(): String = "$first..$last"\n\n    companion object {\n        /** An empty  
range of values of type UInt. */\n        public val EMPTY: UIntRange = UIntRange(UInt.MAX_VALUE,  
UInt.MIN_VALUE)\n    }\n}\n\n/**\n * A progression of values of type `UInt`.\n
```



```

kotlin.ranges\n\nimport kotlin.internal.*\n\n**\n * A range of values of type `ULong`.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic class
ULongRange(start: ULong, endInclusive: ULong) : ULongProgression(start, endInclusive, 1),
ClosedRange<ULong> {\n    override val start: ULong get() = first\n    override val endInclusive: ULong get() =
last\n\n    override fun contains(value: ULong): Boolean = first <= value && value <= last\n\n    /** \n    * Checks
if the range is empty.\n    \n    * The range is empty if its start value is greater than the end value.\n    */\n
    override fun isEmpty(): Boolean = first > last\n\n    override fun equals(other: Any?): Boolean =\n        other is
ULongRange && (isEmpty() && other.isEmpty()) ||\n            first == other.first && last == other.last)\n\n
    override fun hashCode(): Int =\n        if (isEmpty()) -1 else (31 * (first xor (first shr 32)).toInt() + (last xor (last shr
32)).toInt())\n\n    override fun toString(): String = "$first..$last"\n\n    companion object {\n        /** An empty
range of values of type ULong. */\n        public val EMPTY: ULongRange = ULongRange(ULong.MAX_VALUE,
ULong.MIN_VALUE)\n    }\n}\n\n**\n * A progression of values of type `ULong`.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic open class
ULongProgression\n    internal constructor(\n        start: ULong,\n        endInclusive: ULong,\n        step: Long\n    ) :
Iterable<ULong> {\n    init {\n        if (step == 0.toLong()) throw kotlin.IllegalArgumentException("Step must be
non-zero.")\n        if (step == Long.MIN_VALUE) throw kotlin.IllegalArgumentException("Step must be greater
than Long.MIN_VALUE to avoid overflow on negation.")\n    }\n\n    /**\n    * The first element in the
progression.\n    */\n    public val first: ULong = start\n\n    /**\n    * The last element in the progression.\n    */\n
    public val last: ULong = getProgressionLastElement(start, endInclusive, step)\n\n    /**\n    * The step of the
progression.\n    */\n    public val step: Long = step\n\n    final override fun iterator(): Iterator<ULong> =
ULongProgressionIterator(first, last, step)\n\n    /** \n    * Checks if the progression is empty.\n    \n    *
Progression with a positive step is empty if its first element is greater than the last element.\n    * Progression with a
negative step is empty if its first element is less than the last element.\n    */\n    public open fun isEmpty(): Boolean
= if (step > 0) first > last else first < last\n\n    override fun equals(other: Any?): Boolean =\n        other is
ULongProgression && (isEmpty() && other.isEmpty()) ||\n            first == other.first && last == other.last &&
step == other.step)\n\n    override fun hashCode(): Int =\n        if (isEmpty()) -1 else (31 * (31 * (first xor (first shr
32)).toInt() + (last xor (last shr 32)).toInt()) + (step xor (step ushr 32)).toInt())\n\n    override fun toString(): String =
if (step > 0) "$first..$last step $step" else "$first downTo $last step ${-step}"\n\n    companion object {\n
        /**\n        * Creates ULongProgression within the specified bounds of a closed range.\n        \n        * The progression
starts with the [rangeStart] value and goes toward the [rangeEnd] value not excluding it, with the specified [step].\n
        \n        * In order to go backwards the [step] must be negative.\n        \n        * [step] must be greater than
`Long.MIN_VALUE` and not equal to zero.\n        */\n        public fun fromClosedRange(rangeStart: ULong,
rangeEnd: ULong, step: Long): ULongProgression = ULongProgression(rangeStart, rangeEnd, step)\n    }\n}\n\n**\n * An iterator over a progression of values of type `ULong`.\n * @property step the number by which
the value is incremented on each step.\n
*\n@SinceKotlin("1.3")\n@Suppress("DEPRECATION_ERROR")\nprivate class
ULongProgressionIterator(first: ULong, last: ULong, step: Long) : ULongIterator() {\n    private val finalElement =
last\n    private var hasNext: Boolean = if (step > 0) first <= last else first >= last\n    private val step =
step.toULong() // use 2-complement math for negative steps\n    private var next = if (hasNext) first else
finalElement\n\n    override fun hasNext(): Boolean = hasNext\n\n    override fun nextULong(): ULong {\n        val
value = next\n        if (value == finalElement) {\n            if (!hasNext) throw kotlin.NoSuchElementException()\n
            hasNext = false\n        } else {\n            next += step\n        }\n        return value\n    }\n}\n\n**\n * Copyright
2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed
by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.math\n\n**\n *
Returns the smaller of two values.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\n
public inline fun min(a: UInt, b: UInt): UInt {\n    return minOf(a, b)\n}\n\n**\n * Returns the smaller of two
values.\n

```

```

*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun min(a: ULong, b: ULong): ULong {\n    return minOf(a, b)\n}\n\n/**\n * Returns the greater of two values.\n */
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun max(a: UInt, b: UInt): UInt {\n    return maxOf(a, b)\n}\n\n/**\n * Returns the greater of two values.\n */
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun max(a: ULong, b: ULong): ULong {\n    return maxOf(a, b)\n}\n\n"/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */
*\n\n@file:kotlin.jvm.JvmName("UNumbersKt")\n\npackage kotlin\n\n/**\n * Counts the number of set bits in the binary representation of this [UInt] number.\n */
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class, ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UInt.countOneBits(): Int = toInt().countOneBits()\n\n/**\n * Counts the number of consecutive most significant bits that are zero in the binary representation of this [UInt] number.\n */
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class, ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UInt.countLeadingZeroBits(): Int = toInt().countLeadingZeroBits()\n\n/**\n * Counts the number of consecutive least significant bits that are zero in the binary representation of this [UInt] number.\n */
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class, ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UInt.countTrailingZeroBits(): Int = toInt().countTrailingZeroBits()\n\n/**\n * Returns a number having a single bit set in the position of the most significant set bit of this [UInt] number,\n * or zero, if this number is zero.\n */
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class, ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UInt.takeHighestOneBit(): UInt = toInt().takeHighestOneBit().toUInt()\n\n/**\n * Returns a number having a single bit set in the position of the least significant set bit of this [UInt] number,\n * or zero, if this number is zero.\n */
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class, ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UInt.takeLowestOneBit(): UInt = toInt().takeLowestOneBit().toUInt()\n\n/**\n * Rotates the binary representation of this [UInt] number left by the specified [bitCount] number of bits.\n * The most significant bits pushed out from the left side reenter the number as the least significant bits on the right side.\n * Rotating the number left by a negative bit count is the same as rotating it right by the negated bit count:\n * `number.rotateLeft(-n) == number.rotateRight(n)`\n * Rotating by a multiple of [UInt.SIZE_BITS] (32) returns the same number, or more generally\n * `number.rotateLeft(n) == number.rotateLeft(n % 32)`\n */
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class, ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun UInt.rotateLeft(bitCount: Int): UInt = toInt().rotateLeft(bitCount).toUInt()\n\n/**\n * Rotates the binary representation of this [UInt] number right by the specified [bitCount] number of bits.\n * The least significant bits pushed out from the right side reenter the number as the most significant bits on the left side.\n * Rotating the number right by a negative bit count is the same as rotating it left by the negated bit count:\n * `number.rotateRight(-n) == number.rotateLeft(n)`\n * Rotating by a multiple of [UInt.SIZE_BITS] (32) returns the same number, or more generally\n * `number.rotateRight(n) == number.rotateRight(n % 32)`\n */
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class, ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun UInt.rotateRight(bitCount: Int): UInt = toInt().rotateRight(bitCount).toUInt()\n\n/**\n * Counts the number of set bits in the binary representation of this [ULong] number.\n */
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class, ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun ULong.countOneBits(): Int =

```

```

toLong().countOneBits()\n\n**\n * Counts the number of consecutive most significant bits that are zero in the
binary representation of this [ULong] number.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun ULong.countLeadingZeroBits(): Int
= toLong().countLeadingZeroBits()\n\n**\n * Counts the number of consecutive least significant bits that are zero
in the binary representation of this [ULong] number.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun ULong.countTrailingZeroBits(): Int
= toLong().countTrailingZeroBits()\n\n**\n * Returns a number having a single bit set in the position of the most
significant set bit of this [ULong] number,\n * or zero, if this number is zero.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun ULong.takeHighestOneBit(): ULong
= toLong().takeHighestOneBit().toULong()\n\n**\n * Returns a number having a single bit set in the position of the
least significant set bit of this [ULong] number,\n * or zero, if this number is zero.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun ULong.takeLowestOneBit(): ULong
= toLong().takeLowestOneBit().toULong()\n\n**\n * Rotates the binary representation of this [ULong] number left
by the specified [bitCount] number of bits.\n * The most significant bits pushed out from the left side reenter the
number as the least significant bits on the right side.\n * Rotating the number left by a negative bit count is the
same as rotating it right by the negated bit count:\n * `number.rotateLeft(-n) == number.rotateRight(n)`\n *
Rotating by a multiple of [ULong.SIZE_BITS] (64) returns the same number, or more generally\n *
`number.rotateLeft(n) == number.rotateLeft(n % 64)`\n
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class,
ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun ULong.rotateLeft(bitCount:
Int): ULong = toLong().rotateLeft(bitCount).toULong()\n\n**\n * Rotates the binary representation of this [ULong]
number right by the specified [bitCount] number of bits.\n * The least significant bits pushed out from the right side
reenter the number as the most significant bits on the left side.\n * Rotating the number right by a negative bit
count is the same as rotating it left by the negated bit count:\n * `number.rotateRight(-n) == number.rotateLeft(n)`\n
*\n * Rotating by a multiple of [ULong.SIZE_BITS] (64) returns the same number, or more generally\n *
`number.rotateRight(n) == number.rotateRight(n % 64)`\n
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class,
ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun ULong.rotateRight(bitCount:
Int): ULong = toLong().rotateRight(bitCount).toULong()\n\n**\n * Counts the number of set bits in the binary
representation of this [UByte] number.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UByte.countOneBits(): Int =
toUInt().countOneBits()\n\n**\n * Counts the number of consecutive most significant bits that are zero in the
binary representation of this [UByte] number.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UByte.countLeadingZeroBits(): Int =
toByte().countLeadingZeroBits()\n\n**\n * Counts the number of consecutive least significant bits that are zero in
the binary representation of this [UByte] number.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UByte.countTrailingZeroBits(): Int =
toByte().countTrailingZeroBits()\n\n**\n * Returns a number having a single bit set in the position of the most
significant set bit of this [UByte] number,\n * or zero, if this number is zero.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UByte.takeHighestOneBit(): UByte

```

```

= toInt().takeHighestOneBit().toUByte()\n\n/**\n * Returns a number having a single bit set in the position of the
least significant set bit of this [UByte] number,\n * or zero, if this number is zero.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UByte.takeLowestOneBit(): UByte =
toInt().takeLowestOneBit().toUByte()\n\n\n/**\n * Rotates the binary representation of this [UByte] number left by
the specified [bitCount] number of bits.\n * The most significant bits pushed out from the left side reenter the
number as the least significant bits on the right side.\n * Rotating the number left by a negative bit count is the
same as rotating it right by the negated bit count:\n * `number.rotateLeft(-n) == number.rotateRight(n)`\n *
Rotating by a multiple of [UByte.SIZE_BITS] (8) returns the same number, or more generally\n *
`number.rotateLeft(n) == number.rotateLeft(n % 8)`\n
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class,
ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun UByte.rotateLeft(bitCount:
Int): UByte = toByte().rotateLeft(bitCount).toUByte()\n\n\n/**\n * Rotates the binary representation of this [UByte]
number right by the specified [bitCount] number of bits.\n * The least significant bits pushed out from the right side
reenter the number as the most significant bits on the left side.\n * Rotating the number right by a negative bit
count is the same as rotating it left by the negated bit count:\n * `number.rotateRight(-n) == number.rotateLeft(n)`\n
*\n * Rotating by a multiple of [UByte.SIZE_BITS] (8) returns the same number, or more generally\n *
`number.rotateRight(n) == number.rotateRight(n % 8)`\n
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class,
ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun UByte.rotateRight(bitCount:
Int): UByte = toByte().rotateRight(bitCount).toUByte()\n\n\n/**\n * Counts the number of set bits in the binary
representation of this [UShort] number.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UShort.countOneBits(): Int =
toUInt().countOneBits()\n\n\n/**\n * Counts the number of consecutive most significant bits that are zero in the
binary representation of this [UShort] number.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UShort.countLeadingZeroBits(): Int
= toShort().countLeadingZeroBits()\n\n\n/**\n * Counts the number of consecutive least significant bits that are zero
in the binary representation of this [UShort] number.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UShort.countTrailingZeroBits(): Int
= toShort().countTrailingZeroBits()\n\n\n/**\n * Returns a number having a single bit set in the position of the most
significant set bit of this [UShort] number,\n * or zero, if this number is zero.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UShort.takeHighestOneBit(): UShort
= toInt().takeHighestOneBit().toUShort()\n\n\n/**\n * Returns a number having a single bit set in the position of the
least significant set bit of this [UShort] number,\n * or zero, if this number is zero.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class,
ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UShort.takeLowestOneBit(): UShort
= toInt().takeLowestOneBit().toUShort()\n\n\n/**\n * Rotates the binary representation of this [UShort] number left
by the specified [bitCount] number of bits.\n * The most significant bits pushed out from the left side reenter the
number as the least significant bits on the right side.\n * Rotating the number left by a negative bit count is the
same as rotating it right by the negated bit count:\n * `number.rotateLeft(-n) == number.rotateRight(n)`\n *
Rotating by a multiple of [UShort.SIZE_BITS] (16) returns the same number, or more generally\n *
`number.rotateLeft(n) == number.rotateLeft(n % 16)`\n
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class,
ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun UShort.rotateLeft(bitCount:

```

```

Int): UShort = toShort().rotateLeft(bitCount).toUShort()\n\n**\n * Rotates the binary representation of this
[UShort] number right by the specified [bitCount] number of bits.\n * The least significant bits pushed out from the
right side reenter the number as the most significant bits on the left side.\n *\n * Rotating the number right by a
negative bit count is the same as rotating it left by the negated bit count:\n * `number.rotateRight(-n) ==
number.rotateLeft(n)`\n *\n * Rotating by a multiple of [UShort.SIZE_BITS] (16) returns the same number, or more
generally\n * `number.rotateRight(n) == number.rotateRight(n % 16)`\n
*\n@SinceKotlin("1.6")\n@WasExperimental(ExperimentalStdlibApi::class,
ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun UShort.rotateRight(bitCount:
Int): UShort = toShort().rotateRight(bitCount).toUShort()\n","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n *\npackage kotlin.internal\n\n// (a - b) mod c\nprivate fun
differenceModulo(a: UInt, b: UInt, c: UInt): UInt {\n    val ac = a % c\n    val bc = b % c\n    return if (ac >= bc) ac -
bc else ac - bc + c\n}\n\nprivate fun differenceModulo(a: ULong, b: ULong, c: ULong): ULong {\n    val ac = a %
c\n    val bc = b % c\n    return if (ac >= bc) ac - bc else ac - bc + c\n}\n\n**\n * Calculates the final element of a
bounded arithmetic progression, i.e. the last element of the progression which is in the range\n * from [start] to [end]
in case of a positive [step], or from [end] to [start] in case of a negative\n * [step].\n *\n * No validation on passed
parameters is performed. The given parameters should satisfy the condition:\n *\n * - either `step > 0` and `start <=
end`,\n *\n * - or `step < 0` and `start >= end`.\n *\n * @param start first element of the progression\n * @param end
ending bound for the progression\n * @param step increment, or difference of successive elements in the
progression\n * @return the final element of the progression\n * @suppress\n
*\n@PublishedApi\n@SinceKotlin("1.3")\ninternal fun getProgressionLastElement(start: UInt, end: UInt, step:
Int): UInt = when {\n    step > 0 -> if (start >= end) end else end - differenceModulo(end, start, step.toUInt())\n
step < 0 -> if (start <= end) end else end + differenceModulo(start, end, (-step).toUInt())\n    else -> throw
kotlin.IllegalArgumentException("Step is zero.")\n}\n\n**\n * Calculates the final element of a bounded
arithmetic progression, i.e. the last element of the progression which is in the range\n * from [start] to [end] in case
of a positive [step], or from [end] to [start] in case of a negative\n * [step].\n *\n * No validation on passed
parameters is performed. The given parameters should satisfy the condition:\n *\n * - either `step > 0` and `start <=
end`,\n *\n * - or `step < 0` and `start >= end`.\n *\n * @param start first element of the progression\n * @param end
ending bound for the progression\n * @param step increment, or difference of successive elements in the
progression\n * @return the final element of the progression\n * @suppress\n
*\n@PublishedApi\n@SinceKotlin("1.3")\ninternal fun getProgressionLastElement(start: ULong, end: ULong,
step: Long): ULong = when {\n    step > 0 -> if (start >= end) end else end - differenceModulo(end, start,
step.toULong())\n    step < 0 -> if (start <= end) end else end + differenceModulo(start, end, (-step).toULong())\n
else -> throw kotlin.IllegalArgumentException("Step is zero.")\n}\n\n","/*\n * Copyright 2010-2021 JetBrains s.r.o.
and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license
that can be found in the license/LICENSE.txt file.\n *\n\n@file:kotlin.jvm.JvmName("UStringsKt") // string
representation of unsigned numbers\n\npackage kotlin.text\n\n**\n * Returns a string representation of this [Byte]
value in the specified [radix].\n *\n * @throws IllegalArgumentException when [radix] is not a valid radix for
number to string conversion.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic /*inline*/ fun UByte.toString(radix: Int): String = this.toInt().toString(radix)\n\n**\n * Returns a string
representation of this [Short] value in the specified [radix].\n *\n * @throws IllegalArgumentException when [radix]
is not a valid radix for number to string conversion.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic /*inline*/ fun UShort.toString(radix: Int): String = this.toInt().toString(radix)\n\n**\n * Returns a string
representation of this [Int] value in the specified [radix].\n *\n * @throws IllegalArgumentException when [radix] is
not a valid radix for number to string conversion.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly

```

```

\npublic /*inline*/ fun UInt.toString(radix: Int): String = this.toLong().toString(radix)\n\n/**\n * Returns a string
representation of this [Long] value in the specified [radix].\n *\n * @throws IllegalArgumentException when [radix]
is not a valid radix for number to string conversion.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
ULong.toString(radix: Int): String = ulongToString(this.toLong(), checkRadix(radix))\n\n/**\n * Parses the string
as a signed [UByte] number and returns the result.\n *\n * @throws NumberFormatException if the string is not a valid
representation of a number.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun String.toUByte():
UByte = toUByteOrNull() ?: numberFormatError(this)\n\n/**\n * Parses the string as a signed [UByte] number and
returns the result.\n *\n * @throws NumberFormatException if the string is not a valid representation of a number.\n
*\n * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
String.toUByte(radix: Int): UByte = toUByteOrNull(radix) ?: numberFormatError(this)\n\n/**\n * Parses the
string as a [UShort] number and returns the result.\n *\n * @throws NumberFormatException if the string is not a valid
representation of a number.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun String.toUShort():
UShort = toUShortOrNull() ?: numberFormatError(this)\n\n/**\n * Parses the string as a [UShort] number and
returns the result.\n *\n * @throws NumberFormatException if the string is not a valid representation of a number.\n
*\n * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
String.toUShort(radix: Int): UShort = toUShortOrNull(radix) ?: numberFormatError(this)\n\n/**\n * Parses the
string as an [UInt] number and returns the result.\n *\n * @throws NumberFormatException if the string is not a valid
representation of a number.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun String.toUInt():
UInt = toUIntOrNull() ?: numberFormatError(this)\n\n/**\n * Parses the string as an [UInt] number and returns the
result.\n *\n * @throws NumberFormatException if the string is not a valid representation of a number.\n *\n * @throws
IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
String.toUInt(radix: Int): UInt = toUIntOrNull(radix) ?: numberFormatError(this)\n\n/**\n * Parses the string as a
[ULong] number and returns the result.\n *\n * @throws NumberFormatException if the string is not a valid
representation of a number.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun String.toULong():
ULong = toULongOrNull() ?: numberFormatError(this)\n\n/**\n * Parses the string as a [ULong] number and
returns the result.\n *\n * @throws NumberFormatException if the string is not a valid representation of a number.\n
*\n * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
String.toULong(radix: Int): ULong = toULongOrNull(radix) ?: numberFormatError(this)\n\n/**\n * Parses
the string as an [UByte] number and returns the result\n *\n * or `null` if the string is not a valid representation of a
number.\n *\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
String.toUByteOrNull(): UByte? = toUByteOrNull(radix = 10)\n\n/**\n * Parses the string as an [UByte] number
and returns the result\n *\n * or `null` if the string is not a valid representation of a number.\n *\n * @throws
IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
String.toUByteOrNull(radix: Int): UByte? {
    val int = this.toUIntOrNull(radix) ?: return null
    if (int >
UByte.MAX_VALUE) return null
    return int.toUByte()\n}\n\n/**\n * Parses the string as an [UShort] number
and returns the result\n *\n * or `null` if the string is not a valid representation of a number.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
String.toUShortOrNull(): UShort? = toUShortOrNull(radix = 10)\n\n/**\n * Parses the string as an [UShort] number

```

```

and returns the result\n * or `null` if the string is not a valid representation of a number.\n *\n * @throws
IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
String.toUShortOrNull(radix: Int): UShort? {\n    val int = this.toUIntOrNull(radix) ?: return null\n    if (int >
UShort.MAX_VALUE) return null\n    return int.toUShort()\n}\n\n/**\n * Parses the string as an [UInt] number and
returns the result\n * or `null` if the string is not a valid representation of a number.\n
*\n * @throws
IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
String.toUIntOrNull(): UInt? = toUIntOrNull(radix = 10)\n\n/**\n * Parses the string as an [UInt] number and
returns the result\n * or `null` if the string is not a valid representation of a number.\n
*\n * @throws
IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
String.toUIntOrNull(radix: Int): UInt? {\n    checkRadix(radix)\n\n    val length = this.length\n    if (length == 0)
return null\n\n    val limit: UInt = UInt.MAX_VALUE\n    val start: Int\n\n    val firstChar = this[0]\n    if (firstChar
< '0') {\n        if (length == 1 || firstChar != '+') return null\n        start = 1\n    } else {\n        start = 0\n    }\n\n    val
limitForMaxRadix = 119304647u // limit / 36\n    var limitBeforeMul = limitForMaxRadix\n    val uradix =
radix.toUInt()\n    var result = 0u\n    for (i in start until length) {\n        val digit = digitOf(this[i], radix)\n\n        if
(digit < 0) return null\n        if (result > limitBeforeMul) {\n            if (limitBeforeMul == limitForMaxRadix) {\n
                limitBeforeMul = limit / uradix\n                if (result > limitBeforeMul) {\n                    return null\n
                }\n            } else {\n                return null\n            }\n        }\n        result *= uradix\n        val beforeAdding =
result\n        result += digit.toUInt()\n        if (result < beforeAdding) return null // overflow has happened\n    }\n\n    return result\n}\n\n/**\n * Parses the string as an [ULong] number and returns the result\n * or `null` if the string is
not a valid representation of a number.\n
*\n * @throws
IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
String.toULongOrNull(): ULong? = toULongOrNull(radix = 10)\n\n/**\n * Parses the string as an [ULong] number
and returns the result\n * or `null` if the string is not a valid representation of a number.\n
*\n * @throws
IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n
*\n@SinceKotlin("1.5")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
String.toULongOrNull(radix: Int): ULong? {\n    checkRadix(radix)\n\n    val length = this.length\n    if (length ==
0) return null\n\n    val limit: ULong = ULong.MAX_VALUE\n    val start: Int\n\n    val firstChar = this[0]\n    if
(firstChar < '0') {\n        if (length == 1 || firstChar != '+') return null\n        start = 1\n    } else {\n        start = 0\n    }\n\n    val
limitForMaxRadix = 512409557603043100uL // limit / 36\n    var limitBeforeMul =
limitForMaxRadix\n    val uradix = radix.toULong()\n    var result = 0uL\n    for (i in start until length) {\n        val
digit = digitOf(this[i], radix)\n\n        if (digit < 0) return null\n        if (result > limitBeforeMul) {\n            if
(limitBeforeMul == limitForMaxRadix) {\n                limitBeforeMul = limit / uradix\n                if (result >
limitBeforeMul) {\n                    return null\n                }\n            } else {\n                return null\n            }\n        }\n        result *= uradix\n        val beforeAdding = result\n        result += digit.toUInt()\n        if (result <
beforeAdding) return null // overflow has happened\n    }\n\n    return result\n}\n\n"/>\n * Copyright 2010-2018
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*\n\n@file:Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")\npackage kotlin\n\nimport
kotlin.annotation.AnnotationTarget.\nimport kotlin.internal.RequireKotlin\nimport
kotlin.internal.RequireKotlinVersionKind\n\n/**\n * Marks the API that is dependent on the experimental unsigned
types, including those types themselves.\n *\n * Usages of such API will be reported as warnings unless an explicit
opt-in with\n * the [OptIn] annotation, e.g. `@OptIn(ExperimentalUnsignedTypes::class)`\n * or with the `opt-
in=kotlin.ExperimentalUnsignedTypes` compiler option is given.\n *\n * It's recommended to propagate the
experimental status to the API that depends on unsigned types by annotating it with this annotation.\n
*\n * @RequiresOptIn(level = RequiresOptIn.Level.WARNING)\n * @MustBeDocumented\n * @Target(CLASS,
ANNOTATION_CLASS, PROPERTY, FIELD, LOCAL_VARIABLE, VALUE_PARAMETER,

```

CONSTRUCTOR, FUNCTION, PROPERTY_GETTER, PROPERTY_SETTER,
 TYPEALIAS)\n@Retention(AnnotationRetention.BINARY)\n@RequireKotlin("1.2.50", versionKind =
 RequireKotlinVersionKind.COMPILER_VERSION)\npublic annotation class ExperimentalUnsignedTypes\n"/*\n
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code
 is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
 */\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("MathKt")\n\npackage
 kotlin.math\n\n\n// constants, can't use them from nativeMath as they are not constants there\n/** Ratio of the
 circumference of a circle to its diameter, approximately 3.14159. */\n@SinceKotlin("1.2")\npublic const val PI:
 Double = 3.141592653589793\n/** Base of the natural logarithms, approximately 2.71828.
 */\n@SinceKotlin("1.2")\npublic const val E: Double = 2.718281828459045\n\n// region =====
 Double Math =====\n\n/** Computes the sine of the angle [x]
 given in radians.\n * \n * Special cases:\n * - `sin(NaN|+Inf|-Inf)` is `NaN`\n */\n@SinceKotlin("1.2")\npublic
 expect fun sin(x: Double): Double\n\n/** Computes the cosine of the angle [x] given in radians.\n * \n * Special
 cases:\n * - `cos(NaN|+Inf|-Inf)` is `NaN`\n */\n@SinceKotlin("1.2")\npublic expect fun cos(x: Double):
 Double\n\n/** Computes the tangent of the angle [x] given in radians.\n * \n * Special cases:\n * - `tan(NaN|+Inf|-
 Inf)` is `NaN`\n */\n@SinceKotlin("1.2")\npublic expect fun tan(x: Double): Double\n\n/** \n * Computes the arc
 sine of the value [x];\n * the returned value is an angle in the range from $-\pi/2$ to $\pi/2$ radians.\n * \n * Special
 cases:\n * - `asin(x)` is `NaN`, when $abs(x) > 1$ or x is `NaN`\n */\n@SinceKotlin("1.2")\npublic expect fun
 asin(x: Double): Double\n\n/** \n * Computes the arc cosine of the value [x];\n * the returned value is an angle in
 the range from 0.0 to π radians.\n * \n * Special cases:\n * - `acos(x)` is `NaN`, when $abs(x) > 1$ or x is
 `NaN`\n */\n@SinceKotlin("1.2")\npublic expect fun acos(x: Double): Double\n\n/** \n * Computes the arc tangent
 of the value [x];\n * the returned value is an angle in the range from $-\pi/2$ to $\pi/2$ radians.\n * \n * Special cases:\n *
 * - `atan(NaN)` is `NaN`\n */\n@SinceKotlin("1.2")\npublic expect fun atan(x: Double): Double\n\n/** \n *
 Returns the angle `theta` of the polar coordinates `(r, theta)` that correspond\n * to the rectangular coordinates `(x,
 y)` by computing the arc tangent of the value y / x ;\n * the returned value is an angle in the range from $-\pi$ to
 π radians.\n * \n * Special cases:\n * - `atan2(0.0, 0.0)` is 0.0 \n * - `atan2(0.0, x)` is 0.0 for $x > 0$ and π
 for $x < 0$ \n * - `atan2(-0.0, x)` is -0.0 for $x > 0$ and $-\pi$ for $x < 0$ \n * - `atan2(y, +Inf)` is 0.0 for $0 < y <$
 $+Inf$ and -0.0 for $-Inf < y < 0$ \n * - `atan2(y, -Inf)` is π for $0 < y < +Inf$ and $-\pi$ for $-Inf < y < 0$ \n * -
 `atan2(y, 0.0)` is $\pi/2$ for $y > 0$ and $-\pi/2$ for $y < 0$ \n * - `atan2(+Inf, x)` is $\pi/2$ for finite `x` y\n * -
 `atan2(-Inf, x)` is $-\pi/2$ for finite `x`\n * - `atan2(NaN, x)` and `atan2(y, NaN)` is `NaN`\n
 */\n@SinceKotlin("1.2")\npublic expect fun atan2(y: Double, x: Double): Double\n\n/** \n * Computes the
 hyperbolic sine of the value [x].\n * \n * Special cases:\n * - `sinh(NaN)` is `NaN`\n * - `sinh(+Inf)` is $+Inf$ \n *
 - `sinh(-Inf)` is $-Inf$ \n */\n@SinceKotlin("1.2")\npublic expect fun sinh(x: Double): Double\n\n/** \n * Computes
 the hyperbolic cosine of the value [x].\n * \n * Special cases:\n * - `cosh(NaN)` is `NaN`\n * - `cosh(+Inf|-Inf)` is
 $+Inf$ \n */\n@SinceKotlin("1.2")\npublic expect fun cosh(x: Double): Double\n\n/** \n * Computes the hyperbolic
 tangent of the value [x].\n * \n * Special cases:\n * - `tanh(NaN)` is `NaN`\n * - `tanh(+Inf)` is 1.0 \n * - `tanh(-
 Inf)` is -1.0 \n */\n@SinceKotlin("1.2")\npublic expect fun tanh(x: Double): Double\n\n/** \n * Computes the
 inverse hyperbolic sine of the value [x].\n * \n * The returned value is `y` such that $\sinh(y) == x$.\n * \n * Special
 cases:\n * - `asinh(NaN)` is `NaN`\n * - `asinh(+Inf)` is $+Inf$ \n * - `asinh(-Inf)` is $-Inf$ \n
 */\n@SinceKotlin("1.2")\npublic expect fun asinh(x: Double): Double\n\n/** \n * Computes the inverse hyperbolic
 cosine of the value [x].\n * \n * The returned value is positive `y` such that $\cosh(y) == x$.\n * \n * Special cases:\n *
 - `acosh(NaN)` is `NaN`\n * - `acosh(x)` is `NaN` when $x < 1$ \n * - `acosh(+Inf)` is $+Inf$ \n
 */\n@SinceKotlin("1.2")\npublic expect fun acosh(x: Double): Double\n\n/** \n * Computes the inverse hyperbolic
 tangent of the value [x].\n * \n * The returned value is `y` such that $\tanh(y) == x$.\n * \n * Special cases:\n * -
 `tanh(NaN)` is `NaN`\n * - `tanh(x)` is `NaN` when $x > 1$ or $x < -1$ \n * - `tanh(1.0)` is $+Inf$ \n * - `tanh(-
 1.0)` is $-Inf$ \n */\n@SinceKotlin("1.2")\npublic expect fun atanh(x: Double): Double\n\n/** \n * Computes
 $\sqrt{x^2 + y^2}$ without intermediate overflow or underflow.\n * \n * Special cases:\n * - returns $+Inf$ if any of
 arguments is infinite\n * - returns `NaN` if any of arguments is `NaN` and the other is not infinite\n
 */

`public expect fun hypot(x: Double, y: Double): Double` * Computes the positive square root of the value [x]. * Special cases: `sqrt(x)` is `NaN` when `x < 0` or `x` is `NaN`

`public expect fun sqrt(x: Double): Double` * Computes Euler's number `e` raised to the power of the value [x]. * Special cases: `exp(NaN)` is `NaN` * `exp(+Inf)` is `+Inf` * `exp(-Inf)` is `0.0`

`public expect fun exp(x: Double): Double` * Computes `exp(x) - 1`. * This function can be implemented to produce more precise result for [x] near zero. * Special cases: `expm1(NaN)` is `NaN` * `expm1(+Inf)` is `+Inf` * `expm1(-Inf)` is `-1.0` * @see [exp] function.

`public expect fun expm1(x: Double): Double` * Computes the logarithm of the value [x] to the given [base]. * Special cases: `log(x, b)` is `NaN` if either `x` or `b` are `NaN` * `log(x, b)` is `NaN` when `x < 0` or `b <= 0` or `b == 1.0` * `log(+Inf, +Inf)` is `NaN` * `log(+Inf, b)` is `+Inf` for `b > 1` and `-Inf` for `b < 1` * `log(0.0, b)` is `-Inf` for `b > 1` and `+Inf` for `b < 1` * See also logarithm functions for common fixed bases: [ln], [log10] and [log2].

`public expect fun log(x: Double, base: Double): Double` * Computes the natural logarithm (base `E`) of the value [x]. * Special cases: `ln(NaN)` is `NaN` * `ln(x)` is `NaN` when `x < 0.0` * `ln(+Inf)` is `+Inf` * `ln(0.0)` is `-Inf`

`public expect fun ln(x: Double): Double` * Computes the common logarithm (base 10) of the value [x]. * @see [ln] function for special cases.

`public expect fun log10(x: Double): Double` * Computes the binary logarithm (base 2) of the value [x]. * @see [ln] function for special cases.

`public expect fun log2(x: Double): Double` * Computes `ln(x + 1)`. * This function can be implemented to produce more precise result for [x] near zero. * Special cases: `ln1p(NaN)` is `NaN` * `ln1p(x)` is `NaN` where `x < -1.0` * `ln1p(-1.0)` is `-Inf` * `ln1p(+Inf)` is `+Inf` * @see [ln] function * @see [expm1] function

`public expect fun ln1p(x: Double): Double` * Rounds the given value [x] to an integer towards positive infinity. * @return the smallest double value that is greater than or equal to the given value [x] and is a mathematical integer. * Special cases: `ceil(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer.

`public expect fun ceil(x: Double): Double` * Rounds the given value [x] to an integer towards negative infinity. * @return the largest double value that is smaller than or equal to the given value [x] and is a mathematical integer. * Special cases: `floor(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer.

`public expect fun floor(x: Double): Double` * Rounds the given value [x] to an integer towards zero. * @return the value [x] having its fractional part truncated. * Special cases: `truncate(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer.

`public expect fun truncate(x: Double): Double` * Rounds the given value [x] towards the closest integer with ties rounded towards even integer. * Special cases: `round(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer.

`public expect fun round(x: Double): Double` * Returns the absolute value of the given value [x]. * Special cases: `abs(NaN)` is `NaN` * @see absoluteValue extension property for [Double]

`public expect fun abs(x: Double): Double` * Returns the sign of the given value [x]: `-1.0` if the value is negative, `0` if the value is zero, `1.0` if the value is positive * Special case: `sign(NaN)` is `NaN`

`public expect fun sign(x: Double): Double` * Returns the smaller of two values. * If either value is `NaN`, then the result is `NaN`.

`public expect fun min(a: Double, b: Double): Double` * Returns the greater of two values. * If either value is `NaN`, then the result is `NaN`.

`public expect fun max(a: Double, b: Double): Double` // extensions

`Double.pow(x: Double): Double` * Raises this value to the power [x]. * Special cases: `b.pow(0.0)` is `1.0` * `b.pow(1.0) == b` * `b.pow(NaN)` is `NaN` * `NaN.pow(x)` is `NaN` for `x != 0.0` * `b.pow(Inf)` is `NaN` for `abs(b) == 1.0` * `b.pow(x)` is `NaN` for `b < 0` and `x` is finite and not an integer

`Double.pow(x: Double): Double` * Raises this value to the integer power [n]. * See the other overload of [pow] for details.

`Double.pow(n: Int): Double`

Double\n/n/**\n * Returns the absolute value of this value.\n *\n * Special cases:\n * - `NaN.absoluteValue` is `NaN`\n *\n * @see abs function\n */\n@SinceKotlin("1.2")\npublic expect val Double.absoluteValue: Double\n/n/n/**\n * Returns the sign of this value:\n * - `-1.0` if the value is negative,\n * - zero if the value is zero,\n * - `1.0` if the value is positive\n *\n * Special case:\n * - `NaN.sign` is `NaN`\n */\n@SinceKotlin("1.2")\npublic expect val Double.sign: Double\n/n/n/**\n * Returns this value with the sign bit same as of the [sign] value.\n *\n * If [sign] is `NaN` the sign of the result is undefined.\n */\n@SinceKotlin("1.2")\npublic expect fun Double.withSign(sign: Double): Double\n/n/n/**\n * Returns this value with the sign bit same as of the [sign] value.\n */\n@SinceKotlin("1.2")\npublic expect fun Double.withSign(sign: Int): Double\n/n/n/**\n * Returns the ulp (unit in the last place) of this value.\n *\n * An ulp is a positive distance between this value and the next nearest [Double] value larger in magnitude.\n *\n * Special Cases:\n * - `NaN.ulp` is `NaN`\n * - `x.ulp` is `+Inf` when `x` is `+Inf` or `-Inf`\n * - `0.0.ulp` is `Double.MIN_VALUE`\n */\n@SinceKotlin("1.2")\npublic expect val Double.ulp: Double\n/n/n/**\n * Returns the [Double] value nearest to this value in direction of positive infinity.\n */\n@SinceKotlin("1.2")\npublic expect fun Double.nextUp(): Double\n/n/n/**\n * Returns the [Double] value nearest to this value in direction of negative infinity.\n */\n@SinceKotlin("1.2")\npublic expect fun Double.nextDown(): Double\n/n/n/**\n * Returns the [Double] value nearest to this value in direction from this value towards the value [to].\n *\n * Special cases:\n * - `x.nextTowards(y)` is `NaN` if either `x` or `y` are `NaN`\n * - `x.nextTowards(x) == x`\n */\n@SinceKotlin("1.2")\npublic expect fun Double.nextTowards(to: Double): Double\n/n/n/**\n * Rounds this [Double] value to the nearest integer and converts the result to [Int].\n *\n * Ties are rounded towards positive infinity.\n *\n * Special cases:\n * - `x.roundToInt() == Int.MAX_VALUE` when `x > Int.MAX_VALUE`\n * - `x.roundToInt() == Int.MIN_VALUE` when `x < Int.MIN_VALUE`\n *\n * @throws IllegalArgumentException when this value is `NaN`\n */\n@SinceKotlin("1.2")\npublic expect fun Double.roundToInt(): Int\n/n/n/**\n * Rounds this [Double] value to the nearest integer and converts the result to [Long].\n *\n * Ties are rounded towards positive infinity.\n *\n * Special cases:\n * - `x.roundToLong() == Long.MAX_VALUE` when `x > Long.MAX_VALUE`\n * - `x.roundToLong() == Long.MIN_VALUE` when `x < Long.MIN_VALUE`\n *\n * @throws IllegalArgumentException when this value is `NaN`\n */\n@SinceKotlin("1.2")\npublic expect fun Double.roundToLong(): Long\n/n/n// endregion\n/n/n\n// region ===== Float Math\n/n/n/**\n * Computes the sine of the angle [x] given in radians.\n *\n * Special cases:\n * - `sin(NaN|+Inf|-Inf)` is `NaN`\n */\n@SinceKotlin("1.2")\npublic expect fun sin(x: Float): Float\n/n/n/**\n * Computes the cosine of the angle [x] given in radians.\n *\n * Special cases:\n * - `cos(NaN|+Inf|-Inf)` is `NaN`\n */\n@SinceKotlin("1.2")\npublic expect fun cos(x: Float): Float\n/n/n/**\n * Computes the tangent of the angle [x] given in radians.\n *\n * Special cases:\n * - `tan(NaN|+Inf|-Inf)` is `NaN`\n */\n@SinceKotlin("1.2")\npublic expect fun tan(x: Float): Float\n/n/n/**\n * Computes the arc sine of the value [x].\n *\n * the returned value is an angle in the range from $-\pi/2$ to $\pi/2$ radians.\n *\n * Special cases:\n * - `asin(x)` is `NaN`, when `abs(x) > 1` or x is `NaN`\n */\n@SinceKotlin("1.2")\npublic expect fun asin(x: Float): Float\n/n/n/**\n * Computes the arc cosine of the value [x].\n *\n * the returned value is an angle in the range from 0.0 to π radians.\n *\n * Special cases:\n * - `acos(x)` is `NaN`, when `abs(x) > 1` or x is `NaN`\n */\n@SinceKotlin("1.2")\npublic expect fun acos(x: Float): Float\n/n/n/**\n * Computes the arc tangent of the value [x].\n *\n * the returned value is an angle in the range from $-\pi/2$ to $\pi/2$ radians.\n *\n * Special cases:\n * - `atan(NaN)` is `NaN`\n */\n@SinceKotlin("1.2")\npublic expect fun atan(x: Float): Float\n/n/n/**\n * Returns the angle `theta` of the polar coordinates `(r, theta)` that correspond\n * to the rectangular coordinates `(x, y)` by computing the arc tangent of the value [y] / [x].\n *\n * the returned value is an angle in the range from $-\pi$ to π radians.\n *\n * Special cases:\n * - `atan2(0.0, 0.0)` is 0.0 \n * - `atan2(0.0, x)` is 0.0 for $x > 0$ and π for $x < 0$ \n * - `atan2(-0.0, x)` is -0.0 for $x > 0$ and $-\pi$ for $x < 0$ \n * - `atan2(y, +Inf)` is 0.0 for $0 < y < +Inf$ and -0.0 for $-Inf < y < 0$ \n * - `atan2(y, -Inf)` is π for $0 < y < +Inf$ and $-\pi$ for $-Inf < y < 0$ \n * - `atan2(y, 0.0)` is $\pi/2$ for $y > 0$ and $-\pi/2$ for $y < 0$ \n * - `atan2(+Inf, x)` is $\pi/2$ for finite x \n * - `atan2(-Inf, x)` is $-\pi/2$ for finite x \n * - `atan2(NaN, x)` and `atan2(y, NaN)` is `NaN`\n */\n@SinceKotlin("1.2")\npublic expect fun atan2(y: Float, x: Float): Float\n/n/n/**\n * Computes the hyperbolic

sine of the value [x].
Special cases: $\sinh(\text{NaN})$ is NaN $\sinh(+\text{Inf})$ is $+\text{Inf}$ $\sinh(-\text{Inf})$ is $-\text{Inf}$
@SinceKotlin("1.2")\npublic expect fun sinh(x: Float): Float\n/**\n * Computes the hyperbolic cosine of the value [x].
Special cases: $\cosh(\text{NaN})$ is NaN $\cosh(+\text{Inf})$ is $+\text{Inf}$ $\cosh(-\text{Inf})$ is $+\text{Inf}$
@SinceKotlin("1.2")\npublic expect fun cosh(x: Float): Float\n/**\n * Computes the hyperbolic tangent of the value [x].
Special cases: $\tanh(\text{NaN})$ is NaN $\tanh(+\text{Inf})$ is 1.0 $\tanh(-\text{Inf})$ is -1.0
@SinceKotlin("1.2")\npublic expect fun tanh(x: Float): Float\n/**\n * Computes the inverse hyperbolic sine of the value [x].
The returned value is y such that $\sinh(y) == x$.
Special cases: $\text{asinh}(\text{NaN})$ is NaN $\text{asinh}(+\text{Inf})$ is $+\text{Inf}$ $\text{asinh}(-\text{Inf})$ is $-\text{Inf}$
@SinceKotlin("1.2")\npublic expect fun asinh(x: Float): Float\n/**\n * Computes the inverse hyperbolic cosine of the value [x].
The returned value is positive y such that $\cosh(y) == x$.
Special cases: $\text{acosh}(\text{NaN})$ is NaN $\text{acosh}(x)$ is NaN when $x < 1$ $\text{acosh}(+\text{Inf})$ is $+\text{Inf}$
@SinceKotlin("1.2")\npublic expect fun acosh(x: Float): Float\n/**\n * Computes the inverse hyperbolic tangent of the value [x].
The returned value is y such that $\tanh(y) == x$.
Special cases: $\text{atanh}(\text{NaN})$ is NaN $\text{atanh}(x)$ is NaN when $x > 1$ or $x < -1$ $\text{atanh}(1.0)$ is $+\text{Inf}$ $\text{atanh}(-1.0)$ is $-\text{Inf}$
@SinceKotlin("1.2")\npublic expect fun atanh(x: Float): Float\n/**\n * Computes $\sqrt{x^2 + y^2}$ without intermediate overflow or underflow.
Special cases: returns $+\text{Inf}$ if any of arguments is infinite returns NaN if any of arguments is NaN and the other is not infinite
@SinceKotlin("1.2")\npublic expect fun hypot(x: Float, y: Float): Float\n/**\n * Computes the positive square root of the value [x].
Special cases: \sqrt{x} is NaN when $x < 0$ or x is NaN
@SinceKotlin("1.2")\npublic expect fun sqrt(x: Float): Float\n/**\n * Computes Euler's number e raised to the power of the value [x].
Special cases: $\exp(\text{NaN})$ is NaN $\exp(+\text{Inf})$ is $+\text{Inf}$ $\exp(-\text{Inf})$ is 0.0
@SinceKotlin("1.2")\npublic expect fun exp(x: Float): Float\n/**\n * Computes $\exp(x) - 1$.
This function can be implemented to produce more precise result for [x] near zero.
Special cases: $\text{expm1}(\text{NaN})$ is NaN $\text{expm1}(+\text{Inf})$ is $+\text{Inf}$ $\text{expm1}(-\text{Inf})$ is -1.0
@see [exp] function.
@SinceKotlin("1.2")\npublic expect fun expm1(x: Float): Float\n/**\n * Computes the logarithm of the value [x] to the given [base].
Special cases: $\log(x, b)$ is NaN if either x or b are NaN $\log(x, b)$ is NaN when $x < 0$ or $b \leq 0$ or $b == 1.0$ $\log(+\text{Inf}, +\text{Inf})$ is NaN $\log(+\text{Inf}, b)$ is $+\text{Inf}$ for $b > 1$ and $-\text{Inf}$ for $b < 1$ $\log(0.0, b)$ is $-\text{Inf}$ for $b > 1$ and $+\text{Inf}$ for $b > 1$
See also logarithm functions for common fixed bases: [ln], [log10] and [log2].
@SinceKotlin("1.2")\npublic expect fun log(x: Float, base: Float): Float\n/**\n * Computes the natural logarithm (base E) of the value [x].
Special cases: $\ln(\text{NaN})$ is NaN $\ln(x)$ is NaN when $x < 0.0$ $\ln(+\text{Inf})$ is $+\text{Inf}$ $\ln(0.0)$ is $-\text{Inf}$
@SinceKotlin("1.2")\npublic expect fun ln(x: Float): Float\n/**\n * Computes the common logarithm (base 10) of the value [x].
@see [ln] function for special cases.
@SinceKotlin("1.2")\npublic expect fun log10(x: Float): Float\n/**\n * Computes the binary logarithm (base 2) of the value [x].
@see [ln] function for special cases.
@SinceKotlin("1.2")\npublic expect fun log2(x: Float): Float\n/**\n * Computes $\ln(a + 1)$.
This function can be implemented to produce more precise result for [x] near zero.
Special cases: $\ln1p(\text{NaN})$ is NaN $\ln1p(x)$ is NaN where $x < -1.0$ $\ln1p(-1.0)$ is $-\text{Inf}$ $\ln1p(+\text{Inf})$ is $+\text{Inf}$
@see [ln] function
@see [expm1] function.
@SinceKotlin("1.2")\npublic expect fun ln1p(x: Float): Float\n/**\n * Rounds the given value [x] to an integer towards positive infinity.
@return the smallest Float value that is greater than or equal to the given value [x] and is a mathematical integer.
Special cases: $\text{ceil}(x)$ is x where x is NaN or $+\text{Inf}$ or $-\text{Inf}$ or already a mathematical integer.
@SinceKotlin("1.2")\npublic expect fun ceil(x: Float): Float\n/**\n * Rounds the given value [x] to an integer towards negative infinity.
@return the largest Float value that is smaller than or equal to the given value [x] and is a mathematical integer.
Special cases: $\text{floor}(x)$ is x where x is NaN or $+\text{Inf}$ or $-\text{Inf}$ or already a mathematical integer.
@SinceKotlin("1.2")\npublic expect fun floor(x: Float): Float\n/**\n * Rounds the given value [x] to an integer towards zero.
@return the value [x] having its fractional part truncated.
Special cases: $\text{truncate}(x)$ is x where x is NaN or $+\text{Inf}$ or $-\text{Inf}$ or already a mathematical integer.

`*\n@SinceKotlin("1.2")\npublic expect fun truncate(x: Float): Float\n\n/**\n * Rounds the given value [x]
towards the closest integer with ties rounded towards even integer.\n * \n * Special cases:\n * - `round(x)` is `x`
where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer.\n */\n@SinceKotlin("1.2")\npublic expect
fun round(x: Float): Float\n\n/**\n * Returns the absolute value of the given value [x].\n * \n * Special cases:\n *
- `abs(NaN)` is `NaN`\n * \n * @see absoluteValue extension property for [Float]\n */\n@SinceKotlin("1.2")\npublic expect fun abs(x: Float): Float\n\n/**\n * Returns the sign of the given value
[x]:\n * - `-1.0` if the value is negative,\n * - zero if the value is zero,\n * - `1.0` if the value is positive\n * \n * \n * Special case:\n * - `sign(NaN)` is `NaN`\n */\n@SinceKotlin("1.2")\npublic expect fun sign(x: Float):
Float\n\n/**\n * Returns the smaller of two values.\n * \n * If either value is `NaN`, then the result is `NaN`.\n */\n@SinceKotlin("1.2")\npublic expect fun min(a: Float, b: Float): Float\n\n/**\n * Returns the greater of two
values.\n * \n * If either value is `NaN`, then the result is `NaN`.\n */\n@SinceKotlin("1.2")\npublic expect fun
max(a: Float, b: Float): Float\n\n// extensions\n\n/**\n * Raises this value to the power [x].\n * \n * Special
cases:\n * - `b.pow(0.0)` is `1.0`\n * - `b.pow(1.0) == b`\n * - `b.pow(NaN)` is `NaN`\n * - `NaN.pow(x)` is
`NaN` for `x != 0.0`\n * - `b.pow(Inf)` is `NaN` for `abs(b) == 1.0`\n * - `b.pow(x)` is `NaN` for `b < 0` and `x` is
finite and not an integer\n */\n@SinceKotlin("1.2")\npublic expect fun Float.pow(x: Float): Float\n\n/**\n * Raises
this value to the integer power [n].\n * \n * See the other overload of [pow] for details.\n */\n@SinceKotlin("1.2")\npublic expect fun Float.pow(n: Int): Float\n\n/**\n * Returns the absolute value of this
value.\n * \n * Special cases:\n * - `NaN.absoluteValue` is `NaN`\n * \n * @see abs function\n */\n@SinceKotlin("1.2")\npublic expect val Float.absoluteValue: Float\n\n/**\n * Returns the sign of this value:\n *
* - `-1.0` if the value is negative,\n * - zero if the value is zero,\n * - `1.0` if the value is positive\n * \n * \n * Special
case:\n * - `NaN.sign` is `NaN`\n */\n@SinceKotlin("1.2")\npublic expect val Float.sign: Float\n\n/**\n * Returns
this value with the sign bit same as of the [sign] value.\n * \n * If [sign] is `NaN` the sign of the result is undefined.\n */\n@SinceKotlin("1.2")\npublic expect fun Float.withSign(sign: Float): Float\n\n/**\n * Returns this value with
the sign bit same as of the [sign] value.\n */\n@SinceKotlin("1.2")\npublic expect fun Float.withSign(sign: Int):
Float\n\n/**\n * Rounds this [Float] value to the nearest integer and converts the result to [Int].\n * \n * Ties are
rounded towards positive infinity.\n * \n * Special cases:\n * - `x.roundToInt() == Int.MAX_VALUE` when `x >
Int.MAX_VALUE`\n * - `x.roundToInt() == Int.MIN_VALUE` when `x < Int.MIN_VALUE`\n * \n * @throws
IllegalArgumentException when this value is `NaN`\n */\n@SinceKotlin("1.2")\npublic expect fun
Float.roundToInt(): Int\n\n/**\n * Rounds this [Float] value to the nearest integer and converts the result to
[Long].\n * \n * Ties are rounded towards positive infinity.\n * \n * Special cases:\n * - `x.roundToLong() ==
Long.MAX_VALUE` when `x > Long.MAX_VALUE`\n * - `x.roundToLong() == Long.MIN_VALUE` when `x
< Long.MIN_VALUE`\n * \n * @throws IllegalArgumentException when this value is `NaN`\n */\n@SinceKotlin("1.2")\npublic expect fun
Float.roundToLong(): Long\n\n// endregion\n\n// region
==== Integer Math =====\n\n/**\n * Returns
the absolute value of the given value [n].\n * \n * Special cases:\n * - `abs(Int.MIN_VALUE)` is
`Int.MIN_VALUE` due to an overflow\n * \n * @see absoluteValue extension property for [Int]\n */\n@SinceKotlin("1.2")\npublic expect fun abs(n: Int): Int\n\n/**\n * Returns the smaller of two values.\n */\n@SinceKotlin("1.2")\npublic expect fun min(a: Int, b: Int): Int\n\n/**\n * Returns the greater of two values.\n */\n@SinceKotlin("1.2")\npublic expect fun max(a: Int, b: Int): Int\n\n/**\n * Returns the absolute value of this
value.\n * \n * Special cases:\n * - `Int.MIN_VALUE.absoluteValue` is `Int.MIN_VALUE` due to an overflow\n *
* \n * @see abs function\n */\n@SinceKotlin("1.2")\npublic expect val Int.absoluteValue: Int\n\n/**\n * Returns
the sign of this value:\n * - `-1` if the value is negative,\n * - `0` if the value is zero,\n * - `1` if the value is
positive\n */\n@SinceKotlin("1.2")\npublic expect val Int.sign: Int\n\n/**\n * Returns the absolute value of the
given value [n].\n * \n * Special cases:\n * - `abs(Long.MIN_VALUE)` is `Long.MIN_VALUE` due to an
overflow\n * \n * @see absoluteValue extension property for [Long]\n */\n@SinceKotlin("1.2")\npublic expect fun
abs(n: Long): Long\n\n/**\n * Returns the smaller of two values.\n */\n@SinceKotlin("1.2")\npublic expect fun
min(a: Long, b: Long): Long\n\n/**\n * Returns the greater of two values.\n */\n@SinceKotlin("1.2")\npublic
expect fun max(a: Long, b: Long): Long\n\n/**\n * Returns the absolute value of this value.\n * \n * Special cases:\n */`

```

* - `Long.MIN_VALUE.absoluteValue` is `Long.MIN_VALUE` due to an overflow\n * @see abs function\n
*\n@SinceKotlin("1.2")\npublic expect val Long.absoluteValue: Long\n/**\n * Returns the sign of this value:\n * - `1` if the value is negative,\n * - `0` if the value is zero,\n * - `1` if the value is positive\n
*\n@SinceKotlin("1.2")\npublic expect val Long.sign: Int\n\n\nendregion\n"], "names": [], "mappings": "AAWC,CAXA,yB;EACG,IAAI,OAAO,MAAO,KAAL,UAAW,IAAG,MAA
M,IAAIC,C;IACI,MAAM,CAAC,QAAD,EAAW,CAAC,SAAD,CAAX,EAAXB,C;SAEL,IAAI,OAAO,O
AAQ,KAAL,QAAXB,C;IACD,OAAO,CAAC,MAAM,QAAP,C;;IAGP,IAAI,OAAQ,GAAE,E;IACd,OAAO,CAAC
,IAAI,OAAL,C;;CAEd,CAAC,IAAD,EAAO,kB;EACJ,IAAI,IAAI,M;ECPZ,MAAM,eAAgB,GAAE,a;IACpB,OA
AoD,CAA5C,KAAK,QAAQ,CAAC,CAAD,CAAI,IAAG,CAAE,YAAW,SAAW,KAAg,CAAC,OAAQ,KAAL,c;G
;EAGxE,MAAM,YAAa,GAAE,a;IACjB,OAAO,CAAE,YAAW,SAAU,IAAG,CAAC,OAAQ,KAAL,c;G;EAGlD,M
AAM,aAAc,GAAE,a;IACIB,OAAO,CAAE,YAAW,U;G;EAGxB,MAAM,YAAa,GAAE,a;IACjB,OAAO,CAAE,Y
AAW,WAAy,IAAG,CAAC,OAAQ,KAAL,W;G;EAGpD,MAAM,WAAy,GAAE,a;IACbB,OAAO,CAAE,YAAW,
U;G;EAGxB,MAAM,aAAc,GAAE,a;IACIB,OAAO,CAAE,YAAW,Y;G;EAGxB,MAAM,cAAe,GAAE,a;IACnB,O
AAO,CAAE,YAAW,Y;G;EAGxB,MAAM,YAAa,GAAE,a;IACjB,OAAO,KAAK,QAAQ,CAAC,CAAD,CAAI,IA
AG,CAAC,OAAQ,KAAL,W;G;EAG5C,MAAM,QAAS,GAAE,a;IACb,OAAO,KAAK,QAAQ,CAAC,CAAD,CAA
I,IAAG,CAAC,CAAC,O;G;EAGjC,MAAM,WAAy,GAAE,a;IACbB,OAAO,KAAK,QAAQ,CAAC,CAAD,CAAI,
IAAG,WAAW,OAAO,CAAC,CAAD,C;G;EAGjD,MAAM,cAAe,GAAE,a;IACnB,IAAI,CAAE,KAAL,IAAV,C;M
AAgB,OAAO,M;IACvB,IAAI,WAAW,MAAM,YAAy,CAAC,CAAD,CAAI,GAAE,MAAM,aAAR,GAAXB,MA
AM,S;IACnE,OAAO,GAAL,GAAE,KAAK,UAAU,IAAI,KAAK,CAAC,CAAD,EAAL,a;MAAc,OAAO,QAAQ,CA
AC,CAAD,C;KAAjC,CAAwC,KAAK,CAAC,IAAD,CAAO,GAAE,G;G;EAG/F,MAAM,kBAAmB,GAAE,e;IACv
B,OAAO,MAAM,OAAO,YAAy,wBAAXB,CAAC,GAAD,C;G;EAG5D,MAAM,YAAa,GAAE,gB;IACjB,IAAI,C
AAE,KAAL,CAAV,C;MACI,OAAO,I;;IAEX,IAAI,CAAE,KAAL,IAAK,IAAG,CAAE,KAAL,IAAK,IAAG,CAAC,
MAAM,WAAW,CAAC,CAAD,CAAI,IAAG,CAAC,OAAQ,KAAL,CAAC,OAAXE,C;MACI,OAAO,K;;IAGX,KA
AK,IAAI,IAAI,CAAR,EAAX,IAAI,CAAC,OAARB,EAAXB,CAAE,GAAE,CAALC,EAAXQ,CAAC,EAATC,C;MA
CI,IAAI,CAAC,MAAM,OAAO,CAAC,CAAC,CAAC,CAAD,CAAF,EAAX,CAAC,CAAC,CAAD,CAAR,CAALB
,C;QACI,OAAO,K;;IAGf,OAAO,I;G;EAGX,MAAM,gBAALB,GAAE,gB;IACrB,OAAO,MAAM,OAAO,YAAy,s
BAAsB,CAAC,CAAD,EAAL,CAAJ,C;G;EAGlD,MAAM,cAAe,GAAE,e;IACnB,IAAI,GAAL,KAAL,IAAZ,C;MA
AkB,OAAO,C;IACzB,IAAI,SAAS,C;IACb,KAAK,IAAI,IAAI,CAAR,EAAX,IAAI,GAAG,OAAXB,EAAGC,CAA
E,GAAE,CAAPC,EAAXC,CAAC,EAAXC,C;MACI,MAAO,GAAXB,CAAJB,EAAG,GAAE,MAAO,GAAE,CAAG
,IAAE,MAAM,SAAS,CAAC,GAAG,CAAC,CAAD,CAAJ,CAAU,GAAE,C;;IAE7D,OAAO,M;G;EAGX,MAAM,
kBAAmB,GAAE,e;IACvB,OAAO,MAAM,OAAO,YAAy,wBAAXB,CAAC,GAAD,C;G;EAG5D,MAAM,mBAA
oB,GAAE,iB;IACxB,KAAK,KAAK,CAAC,MAAM,gBAAP,C;G;ECPfD,MAAM,eAAgB,GAAE,mB;IACpB,CA
AC,aAAc,GAAE,I;IACjB,OAAO,C;G;EAGX,MAAM,uBAAXB,GAAE,4C;IAC5B,MAAM,IAAK,GAAE,M;IAC
b,MAAM,IAAK,GAAE,M;IACb,MAAM,aAAc,GAAE,I;IACtB,OAAO,mBAAmB,CAAC,MAAD,EAAS,MAAT,
EAALB,6BAA6B,CAAC,UAAD,CAA9C,C;G;EAG9B,iD;IACI,GAAG,WAAy,GAAE,sBAAsB,CAAC,OAAO,M
AAO,KAAL,UAAW,GAAE,KAAK,QAAP,GAAXB,KAAK,UAArD,C;IACvC,GAAG,YAAa,GAAE,G;IACIB,OA
AO,G;G;EAGX,IAAI,gCAAgC,CACc,UACa,QAAS,IAAT,wBAAXC,Y;IAC1C,OAAO,MAAM,OAAO,QAAQ,k
B;GADvB,CADb,aALe,QAAS,IAAT,wBAAXC,Y;IAC5C,OAAO,MAAM,OAAO,QAAQ,W;GADrB,CAJf,CADgC
,EAShC,UACa,QAAS,IAAT,wBAAXC,Y;IAC1C,OAAO,MAAM,OAAO,QAAQ,kB;GADvB,CADb,aALe,QAAS,I
AAT,wBAAXC,Y;IAC5C,OAAO,MAAM,OAAO,QAAQ,W;GADrB,CAJf,CATgC,C;EAmBpC,uC;IACI,IAAI,KA
AK,MAAO,KAAL,IAApB,C;MACI,KAAK,MAAO,GAAE,aACE,CAAC,KAAK,qBAAXB,EAAXB,CADF,aAEC,I
AFD,aAGC,EAHD,cAIE,EAJF,SAKH,EALG,iBAMK,EANL,C;;IASIB,OAAO,KAAK,M;G;EChDhB,MAAM,QA
AS,GAAE,a;IACb,OAAoB,CAAZ,CAAE,GAAE,KAAQ,KAAG,EAAG,IAAG,E;G;EAGjC,MAAM,OAAQ,GAA
E,a;IACZ,OAAXB,CAAV,CAAE,GAAE,GAAM,KAAG,EAAG,IAAG,E;G;EAG/B,MAAM,OAAQ,GAAE,a;IAC
Z,OAAO,CAAE,GAAE,K;G;EAGf,MAAM,aAAc,GAAE,a;IACIB,OAAO,CAAE,YAAW,MAAM,KAAM,GAAE,
CAAF,GAAM,MAAM,KAAK,WAAW,CAAC,CAAD,C;G;EAGhE,MAAM,YAAa,GAAE,a;IACjB,OAAO,CAAE
,YAAW,MAAM,KAAM,GAAE,CAAC,MAAM,EAAT,GAAC,MAAM,YAAy,CAAC,CAAD,C;G;EAGpE,MAA
M,cAAe,GAAE,a;IACnB,OAAO,MAAM,QAAQ,CAAC,MAAM,YAAy,CAAC,CAAD,CAAnB,C;G;EAGzB,MA

```

AM,aAAc,GAAE,a;IACIB,OAAO,MAAM,OAAO,CAAC,MAAM,YAAY,CAAC,CAAD,CAAnB,C;G;EAGxB,M
AAM,eAAgB,GAAE,a;IACpB,OAAO,CAAC,C;G;EAGZ,MAAM,aAAc,GAAE,a;IACIB,OAAO,MAAM,OAAO,
CAAC,MAAM,YAAY,CAAC,CAAD,CAAnB,C;G;EAGxB,MAAM,YAAa,GAAE,a;IACjB,IAAI,CAAE,GAAE,U
AAR,C;MAAoB,OAAO,U;IAC3B,IAAI,CAAE,GAAE,WAAR,C;MAAqB,OAAO,W;IAC5B,OAAO,CAAE,GAA
E,C;G;EAGf,MAAM,YAAa,GAAE,a;IACjB,IAAI,CAAE,IAAG,IAAT,C;MAAe,OAAO,C;IACtB,IAAI,CAAE,Y
AAW,MAAM,UAAvB,C;MAAmC,OAAO,C;IAC1C,OAAO,IAAI,MAAM,UAAV,CAAqB,CAArB,C;G;EAGX,M
AAM,UAAW,GAAE,a;IACf,IAAI,CAAE,IAAG,IAAT,C;MAAe,OAAO,C;IACtB,OAAO,MAAM,OAAO,CAAC,
CAAD,C;G;ECIDxB,MAAM,OAAQ,GAAE,sB;IACZ,IAAI,IAAK,IAAG,IAAZ,C;MACI,OAAO,IAAK,IAAG,I;I
AGnB,IAAI,IAAK,IAAG,IAAZ,C;MACI,OAAO,K;IAGX,IAAI,IAAK,KAAL,IAAb,C;MACI,OAAO,IAAK,KAA
LI,I;IAGpB,IAAI,OAAO,IAAK,KAAL,QAAS,IAAG,OAAO,IAAI,OAAQ,KAAL,UAAvD,C;MACI,OAAO,IAAI,O
AAO,CAAC,IAAD,C;IAGtB,IAAI,OAAO,IAAK,KAAL,QAAS,IAAG,OAAO,IAAK,KAAL,QAAhD,C;MACI,OA
AO,IAAK,KAAL,IAAK,KAAL,IAAK,KAAL,CAAE,IAAG,CAAE,GAAE,IAAK,KAAL,CAAE,GAAE,IAAnC,C;I
AGzB,OAAO,IAAK,KAAL,I;G;EAGpB,MAAM,SAAU,GAAE,e;IACd,IAAI,GAAL,IAAG,IAAX,C;MACI,OAAO,
C;IAEX,IAAI,UAAU,OAAO,G;IACrB,IAAI,QAAS,KAAL,OAAjB,C;MACI,OAAO,UAAW,KAAL,OAAO,GAA
G,SAAU,GAAE,GAAG,SAAS,EAAd,GAAMb,iBAAiB,CAAC,GAAD,C;IAEIF,IAAI,UAAW,KAAL,OAAnB,C;
MACI,OAAO,iBAAiB,CAAC,GAAD,C;IAE5B,IAAI,QAAS,KAAL,OAAjB,C;MACI,OAAO,MAAM,eAAe,CAA
C,GAAD,C;IAEhC,IAAI,SAAU,KAAL,OAAlB,C;MACI,OAAO,MAAM,CAAC,GAAD,C;IAGjB,IAAI,MAAM,
MAAM,CAAC,GAAD,C;IACHb,OAAO,iBAAiB,CAAC,GAAD,C;G;EAI5B,MAAM,SAAU,GAAE,a;IACd,IAAI,
CAAE,IAAG,IAAT,C;MACI,OAAO,M;WAEN,IAAI,MAAM,WAAW,CAAC,CAAD,CAArB,C;MACD,OAAO,O
;MAGP,OAAO,CAAC,SAAS,E;G;EAKzB,IAAI,WAAW,a;EAGf,IAAI,iCAAiC,sB;EAErC,gC;IACI,IAAI,EA
AE,8BAA+B,IAAG,GAAPc,CAAJ,C;MACI,IAAI,OAAQ,IAAI,OAAO,EAAG,GAAE,QAAU,GAAE,C;MACxC,MA
AM,eAAe,CAAC,GAAD,EAAM,8BAAN,EAAsC,QAAU,IAAV,cAA4B,KAA5B,CAAtC,C;IAEzB,OAAO,GAA
G,CAAC,8BAAD,C;G;EAGd,gC;IACI,IAAI,OAAO,C;IACX,KAAK,IAAI,IAAI,CAAb,EAAGb,CAAE,GAAE,G
AAG,OAAvB,EAAGc,CAAC,EAajC,C;MACI,IAAI,OAAQ,GAAG,WAAW,CAAC,CAAD,C;MAC1B,IAAM,G
AAG,IAAK,GAAE,EAAG,GAAE,IAAM,GAAE,C;IAEjC,OAAO,I;G;EAGX,MAAM,iBAakB,GAAE,iB;EC9C1
B,MAAM,KAAM,GAAE,qB;IAKZ,IAAI,KAAM,GAAE,GAAL,GAAE,C;IAMIB,IAAI,MAAO,GAAE,IAAK,GA
AE,C;G;EAGtB,MAAM,KAAK,WAAY,GAAE,OACf,OADe,cAET,MAFS,cAGV,EAHU,C;EAgBzB,MAAM,KA
AK,UAAW,GAAE,E;EAQxB,MAAM,KAAK,QAAS,GAAE,iB;IACpB,IAAI,IAAK,IAAG,KAAM,IAAG,KAAM,
GAAE,GAA7B,C;MACE,IAAI,YAAY,MAAM,KAAK,UAAU,CAAC,KAAD,C;MACrC,IAAI,SAAJ,C;QACE,O
AAO,S;IAIX,IAAI,MAAM,IAAI,MAAM,KAAV,CAAgB,KAAM,GAAE,CAAxB,EA2B,KAAM,GAAE,CAA
E,GAAE,EAaf,GAAG,CAA5C,C;IACV,IAAI,IAAK,IAAG,KAAM,IAAG,KAAM,GAAE,GAA7B,C;MACE,MA
AM,KAAK,UAAU,CAAC,KAAD,CAAQ,GAAE,G;IAEjC,OAAO,G;G;EAYT,MAAM,KAAK,WAAY,GAAE,iB
;IACvB,IAAI,KAAK,CAAC,KAAD,CAAT,C;MACE,OAAO,MAAM,KAAK,K;WACb,IAAI,KAAM,IAAG,CAA
C,MAAM,KAAK,gBAazB,C;MACL,OAAO,MAAM,KAAK,U;WACb,IAAI,KAAM,GAAE,CAAZ,C;MACL,OAAO,MA
AM,KAAK,WAAW,CAAC,CAAC,KAAf,CAAQ,OAAO,E;MAE5C,OAAO,IAAI,MAAM,KAAV,CACf,KAAM
,GAAE,MAAM,KAAK,gBAakB,GAAE,CADrC,EAef,KAAM,GAAE,MAAM,KAAK,gBAakB,GAAE,CAFrC,
C;G;EAcX,MAAM,KAAK,SAAU,GAAE,6B;IACrB,OAAO,IAAI,MAAM,KAAV,CAAgB,OAahB,EAyB,QA
AzB,C;G;EAWT,MAAM,KAAK,WAAY,GAAE,0B;IACvB,IAAI,GAAG,OAAQ,IAAG,CAAIb,C;MACE,MAAM
,KAAK,CAAC,mCAAD,C;IAGb,IAAI,QAAQ,SAAU,IAAG,E;IACzB,IAAI,KAAM,GAAE,CAAE,IAAG,EAAG,
GAAE,KAAtB,C;MACE,MAAM,KAAK,CAAC,sBAaUB,GAAE,KAA1B,C;IAGb,IAAI,GAAG,OAAO,CAAC,C
AAD,CAAI,IAAG,GAARb,C;MACE,OAAO,MAAM,KAAK,WAAW,CAAC,GAAG,UAAU,CAAC,CAAD,CAAd
,EAAMb,KAAnB,CAAyB,OAAO,E;WACxD,IAAI,GAAG,QAAQ,CAAC,GAAD,CAAM,IAAG,CAAxB,C;MAC
L,MAAM,KAAK,CAAC,+CAAgD,GAAE,GAAnD,C;IAKb,IAAI,eAAe,MAAM,KAAK,WAAW,CAAC,IAAI,IA
AI,CAAC,KAAD,EAQ,CAAR,CAAT,C;IAEzC,IAAI,SAAS,MAAM,KAAK,K;IACxB,KAAK,IAAI,IAAI,CAA
b,EAAGb,CAAE,GAAE,GAAG,OAAvB,EAAGc,CAAE,IAAG,CAArC,C;MACE,IAAI,OAAO,IAAI,IAAI,CAAC
,CAAD,EAAL,GAAG,OAAQ,GAAE,CAAjB,C;MACnB,IAAI,QAAQ,QAAQ,CAAC,GAAG,UAAU,CAAC,CAA
D,EAAL,CAAE,GAAE,IAAR,CAAd,EA6B,KAA7B,C;MACpB,IAAI,IAAK,GAAE,CAAX,C;QACE,IAAI,QAA

Q,MAAM,KAAK,WAAW,CAAC,IAAI,IAAI,CAAC,KAAD,EAAQ,IAAR,CAAT,C;QACIC,MAAO,GAAE,MAA
M,SAAS,CAAC,KAAD,CAAO,IAAI,CAAC,MAAM,KAAK,WAAW,CAAC,KAAD,CAA vB,C;;QAE nC,MAAO,
GAAE,MAAM,SAAS,CAAC,YAAD,C;QACx B,MAAO,GAAE,MAAM,IAAI,CAAC,MAAM,KAAK,WAAW,CA
AC,KAAD,CAA vB,C;;IAGv B,OAAO,M;G;EAcT,MAAM,KAAK,gBAAiB,GAAE,CAAE,IAAG,E;EAO nC,MA
AM,KAAK,gBAAiB,GAAE,CAAE,IAAG,E;EAO nC,MAAM,KAAK,gBAAiB,GACx B,MAAM,KAAK,gBAAiB,
GAAE,MAAM,KAAK,gB;EAO7C,MAAM,KAAK,gBAAiB,GACx B,MAAM,KAAK,gBAAiB,GAAE,C;EAOIC,
MAAM,KAAK,gBAAiB,GACx B,MAAM,KAAK,gBAAiB,GAAE,MAAM,KAAK,gB;EAO7C,MAAM,KAAK,gB
AAiB,GACx B,MAAM,KAAK,gBAAiB,GAAE,MAAM,KAAK,gB;EAO7C,MAAM,KAAK,gBAAiB,GACx B,MA
AM,KAAK,gBAAiB,GAAE,C;EAIIC,MAAM,KAAK,KAAM,GAAE,MAAM,KAAK,QAAQ,CAAC,CAAD,C;EA
ItC,MAAM,KAAK,IAAK,GAAE,MAAM,KAAK,QAAQ,CAAC,CAAD,C;EAIrC,MAAM,KAAK,QAAS,GAAE,
MAAM,KAAK,QAAQ,CAAC,EAAD,C;EAIzC,MAAM,KAAK,UAAW,GACIB,MAAM,KAAK,SAAS,CAAC,aA
AW,GAAE,CAAd,EAAiB,UAAW,GAAE,CAA9B,C;EAIxB,MAAM,KAAK,UAAW,GAAE,MAAM,KAAK,SAA
S,CAAC,CAAD,EAAI,aAAW,GAAE,CAAjB,C;EAO5C,MAAM,KAAK,YAAa,GAAE,MAAM,KAAK,QAAQ,C
AAC,CAAE,IAAG,EAAN,C;EAI7C,MAAM,KAAK,UAAU,MAAO,GAAE,Y;IAC5B,OAAO,IAAI,K;G;EAKb,M
AAM,KAAK,UAAU,SAAU,GAAE,Y;IAC/B,OAAO,IAAI,MAAO,GAAE,MAAM,KAAK,gBAAiB,GACzC,IAAI
,mBAAmB,E;G;EAIhC,MAAM,KAAK,UAAU,SAAU,GAAE,Y;IAC/B,OAAO,IAAI,MAAO,GAAE,IAAI,K;G;E
AQ1B,MAAM,KAAK,UAAU,SAAU,GAAE,qB;IAC/B,IAAI,QAAQ,SAAU,IAAG,E;IACzB,IAAI,KAAM,GAAE
,CAAE,IAAG,EAAG,GAAE,KAAtB,C;MACE,MAAM,KAAK,CAAC,sBAAuB,GAAE,KAA1B,C;;IAGb,IAAI,I
AAI,OAAO,EAaf,C;MACE,OAAO,G;;IAGT,IAAI,IAAI,WAAW,EAAnB,C;MACE,IAAI,IAAI,WAAW,CAAC,
MAAM,KAAK,UAAZ,CAAnB,C;QAGE,IAAI,YAAY,MAAM,KAAK,WAAW,CAAC,KAAD,C;QACtC,IAAI,M
AAM,IAAI,IAAI,CAAC,SAAD,C;QACIB,IAAI,MAAM,GAAG,SAAS,CAAC,SAAD,CAAW,SAAS,CAAC,IAA
D,C;QACIC,OAAO,GAAG,SAAS,CAAC,KAAD,CAAQ,GAAE,GAAG,MAAM,EAAE,SAAS,CAAC,KAAD,C;;
QAEjD,OAAO,GAAI,GAAE,IAAI,OAAO,EAAE,SAAS,CAAC,KAAD,C;;IAMvC,IAAI,eAAe,MAAM,KAAK,
WAAW,CAAC,IAAI,IAAI,CAAC,KAAD,EAAQ,CAAR,CAAT,C;IAEzC,IAAI,MAAM,I;IACV,IAAI,SAAS,E;IA
Cb,OAAO,IAAP,C;MACE,IAAI,SAAS,GAAG,IAAI,CAAC,YAAD,C;MACpB,IAAI,SAAS,GAAG,SAAS,CAAC
,MAAM,SAAS,CAAC,YAAD,CAAhB,CAA+B,MAAM,E;MAC9D,IAAI,SAAS,MAAM,SAAS,CAAC,KAAD,C;
MAE5B,GAAI,GAAE,M;MACN,IAAI,GAAG,OAAO,EAAd,C;QACE,OAAO,MAAO,GAAE,M;;QAEhB,OAAO,
MAAM,OAAQ,GAAE,CAA vB,C;UACE,MAAO,GAAE,GAAI,GAAE,M;;QAEjB,MAAO,GAAE,EAAG,GAAE,
MAAO,GAAE,M;;G;EAO7B,MAAM,KAAK,UAAU,YAAa,GAAE,Y;IACIC,OAAO,IAAI,M;G;EAKb,MAAM,K
AAK,UAAU,WAA Y,GAAE,Y;IACjC,OAAO,IAAI,K;G;EAKb,MAAM,KAAK,UAAU,mBAAoB,GAAE,Y;IACz
C,OAAQ,IAAI,KAAM,IAAG,CAAG,GACpB,IAAI,KADgB,GACR,MAAM,KAAK,gBAAiB,GAAE,IAAI,K;G;E
AQpD,MAAM,KAAK,UAAU,cAAe,GAAE,Y;IACpC,IAAI,IAAI,WAAW,EAAnB,C;MACE,IAAI,IAAI,WAAW,
CAAC,MAAM,KAAK,UAAZ,CAAnB,C;QACE,OAAO,E;;QAEp,OAAO,IAAI,OAAO,EAAE,cAAc,E;;MAGpC,
IAAI,MAAM,IAAI,MAAO,IAAG,CAAE,GAAE,IAAI,MAAN,GAAe,IAAI,K;MAC7C,KAAK,IAAI,MAAM,EA
Af,EAAMB,GAAI,GAAE,CAAzB,EAA4B,GAAG,EAA/B,C;QACE,IAAuB,CAAIB,GAAI,GAAG,CAAE,IAAG,
GAAM,KAAG,CAA1B,C;UACE,K;;MAGJ,OAAO,IAAI,MAAO,IAAG,CAAE,GAAE,GAAI,GAAE,EAAR,GA
Aa,GAAI,GAAE,C;;G;EAM9C,MAAM,KAAK,UAAU,OAAQ,GAAE,Y;IAC7B,OAAO,IAAI,MAAO,IAAG,CAA
E,IAAG,IAAI,KAAM,IAAG,C;G;EAKzC,MAAM,KAAK,UAAU,WAA Y,GAAE,Y;IACjC,OAAO,IAAI,MAAO,
GAAE,C;G;EAKtB,MAAM,KAAK,UAAU,MAAO,GAAE,Y;IAC5B,OAAuB,CAAf,IAAI,KAAM,GAAE,CAAG,
KAAG,C;G;EAQ5B,MAAM,KAAK,UAAU,WAA Y,GAAE,iB;IACjC,OAAQ,IAAI,MAAO,IAAG,KAAK,MAAQ
,IAAI,IAAI,KAAM,IAAG,KAAK,K;G;EAQ3D,MAAM,KAAK,UAAU,cAAe,GAAE,iB;IACpC,OAAQ,IAAI,MA
AO,IAAG,KAAK,MAAQ,IAAI,IAAI,KAAM,IAAG,KAAK,K;G;EAQ3D,MAAM,KAAK,UAAU,SAAU,GAAE,i
B;IAC/B,OAAO,IAAI,QAAQ,CAAC,KAAD,CAAQ,GAAE,C;G;EAQ/B,MAAM,KAAK,UAAU,gBAAiB,GAAE,
iB;IACtC,OAAO,IAAI,QAAQ,CAAC,KAAD,CAAQ,IAAG,C;G;EAQhC,MAAM,KAAK,UAAU,YAAa,GAAE,iB
;IACIC,OAAO,IAAI,QAAQ,CAAC,KAAD,CAAQ,GAAE,C;G;EAQ/B,MAAM,KAAK,UAAU,mBAAoB,GAAE,i
B;IACzC,OAAO,IAAI,QAAQ,CAAC,KAAD,CAAQ,IAAG,C;G;EAUhC,MAAM,KAAK,UAAU,QAAS,GAAE,iB
;IAC9B,IAAI,IAAI,WAAW,CAAC,KAAD,CAAnB,C;MACE,OAAO,C;;IAGT,IAAI,UAAU,IAAI,WAAW,E;IAC
7B,IAAI,WAAW,KAAK,WAAW,E;IAC/B,IAAI,OAAQ,IAAG,CAAC,QAAhB,C;MACE,OAAO,E;;IAET,IAAI,C

AAC,OAAQ,IAAG,QAaB,C;MACE,OAAO,C;;IAIT,IAAI,IAAI,SAAS,CAAC,KAAD,CAAO,WAAW,EAAnC,
C;MACE,OAAO,E;;MAEP,OAAO,C;;G;EAMX,MAAM,KAAC,UAAU,OAAQ,GAAE,Y;IAC7B,IAAI,IAAI,WA
AW,CAAC,MAAM,KAAC,UAAZ,CAAnB,C;MACE,OAAO,MAAM,KAAC,U;;MAEIB,OAAO,IAAI,IAAI,EA
E,IAAI,CAAC,MAAM,KAAC,IAAZ,C;;G;EAUzB,MAAM,KAAC,UAAU,IAAK,GAAE,iB;IAG1B,IAAI,MAAM
,IAAI,MAAO,KAAC,E;IACzB,IAAI,MAAM,IAAI,MAAO,GAAE,K;IACvB,IAAI,MAAM,IAAI,KAAM,KAAC,E;I
ACxB,IAAI,MAAM,IAAI,KAAM,GAAE,K;IAEtB,IAAI,MAAM,KAAC,MAAO,KAAC,E;IAC1B,IAAI,MAAM,K
AAK,MAAO,GAAE,K;IACxB,IAAI,MAAM,KAAC,KAAM,KAAC,E;IACzB,IAAI,MAAM,KAAC,KAAM,GAA
E,K;IAEvB,IAAI,MAAM,CAAV,EAaA,MAAM,CAAnB,EAAsB,MAAM,CAA5B,EAa+B,MAAM,C;IACrC,GA
AI,IAAG,GAAC,GAAE,G;IACb,GAAC,IAAG,GAAC,KAAC,E;IACf,GAAC,IAAG,K;IACP,GAAC,IAAG,GAAC,GA
AE,G;IACb,GAAC,IAAG,GAAC,KAAC,E;IACf,GAAC,IAAG,K;IACP,GAAC,IAAG,GAAC,GAAE,G;IACb,GAAC,I
AAG,GAAC,KAAC,E;IACf,GAAC,IAAG,K;IACP,GAAC,IAAG,GAAC,GAAE,G;IACb,GAAC,IAAG,K;IACP,OAAO
,MAAM,KAAC,SAAS,CAAE,GAAC,IAAG,EAAC,GAAE,GAAf,EAaQb,GAAC,IAAG,EAAC,GAAE,GAAIC,C;G;
EAS7B,MAAM,KAAC,UAAU,SAAU,GAAE,iB;IAC/B,OAAO,IAAI,IAAI,CAAC,KAAC,OAAO,EAAb,C;G;EA
SjB,MAAM,KAAC,UAAU,SAAU,GAAE,iB;IAC/B,IAAI,IAAI,OAAO,EAaf,C;MACE,OAAO,MAAM,KAAC,K
;WAcB,IAAI,KAAC,OAAO,EAaB,C;MACL,OAAO,MAAM,KAAC,K;;IAGpB,IAAI,IAAI,WAAW,CAAC,MA
AM,KAAC,UAAZ,CAAnB,C;MACE,OAAO,KAAC,MAAM,EAAG,GAAE,MAAM,KAAC,UAAb,GAA0B,MA
AM,KAAC,K;WACrD,IAAI,KAAC,WAAW,CAAC,MAAM,KAAC,UAAZ,CAApB,C;MACL,OAAO,IAAI,MA
AM,EAAG,GAAE,MAAM,KAAC,UAAb,GAA0B,MAAM,KAAC,K;;IAG3D,IAAI,IAAI,WAAW,EAAnB,C;MA
CE,IAAI,KAAC,WAAW,EAApB,C;QACE,OAAO,IAAI,OAAO,EAAC,SAAS,CAAC,KAAC,OAAO,EAAb,C;;Q
AE7B,OAAO,IAAI,OAAO,EAAC,SAAS,CAAC,KAAD,CAAO,OAAO,E;;WAExC,IAAI,KAAC,WAAW,EAAPB
,C;MACL,OAAO,IAAI,SAAS,CAAC,KAAC,OAAO,EAAb,CAAgB,OAAO,E;;IAI7C,IAAI,IAAI,SAAS,CAAC,
MAAM,KAAC,YAAZ,CAA0B,IACvC,KAAC,SAAS,CAAC,MAAM,KAAC,YAAZ,CADIB,C;MAEE,OAAO,M
AAM,KAAC,WAAW,CAAC,IAAI,SAAS,EAAG,GAAE,KAAC,SAAS,EAajC,C;;IAM/B,IAAI,MAAM,IAAI,M
AAO,KAAC,E;IACzB,IAAI,MAAM,IAAI,MAAO,GAAE,K;IACvB,IAAI,MAAM,IAAI,KAAM,KAAC,E;IACxB,I
AAI,MAAM,IAAI,KAAM,GAAE,K;IAEtB,IAAI,MAAM,KAAC,MAAO,KAAC,E;IAC1B,IAAI,MAAM,KAAC,
MAAO,GAAE,K;IACxB,IAAI,MAAM,KAAC,KAAM,KAAC,E;IACzB,IAAI,MAAM,KAAC,KAAM,GAAE,K;I
AEvB,IAAI,MAAM,CAAV,EAaA,MAAM,CAAnB,EAAsB,MAAM,CAA5B,EAa+B,MAAM,C;IACrC,GAAC,IA
AG,GAAC,GAAE,G;IACb,GAAC,IAAG,GAAC,KAAC,E;IACf,GAAC,IAAG,K;IACP,GAAC,IAAG,GAAC,GAAE,G;I
ACb,GAAC,IAAG,GAAC,KAAC,E;IACf,GAAC,IAAG,K;IACP,GAAC,IAAG,GAAC,GAAE,G;IACb,GAAC,IAAG,G
AAC,KAAC,E;IACf,GAAC,IAAG,K;IACP,GAAC,IAAG,GAAC,GAAE,G;IACb,GAAC,IAAG,GAAC,KAAC,E;IACf,G
AAC,IAAG,K;IACP,GAAC,IAAG,GAAC,GAAE,G;IACb,GAAC,IAAG,GAAC,KAAC,E;IACf,GAAC,IAAG,K;IACP,
GAAC,IAAG,GAAC,GAAE,G;IACb,GAAC,IAAG,GAAC,KAAC,E;IACf,GAAC,IAAG,K;IACP,GAAC,IAAG,GAAC,
GAAE,GAAI,GAAE,GAAI,GAAE,GAAI,GAAE,GAAI,GAAE,GAAI,GAAE,GAAI,GAAE,GAAI,GAAE,G;IACjD,GAAC,IAA
G,K;IACP,OAAO,MAAM,KAAC,SAAS,CAAE,GAAC,IAAG,EAAC,GAAE,GAAf,EAaQb,GAAC,IAAG,EAAC,G
AAE,GAAIC,C;G;EAS7B,MAAM,KAAC,UAAU,IAAK,GAAE,iB;IAC1B,IAAI,KAAC,OAAO,EAaB,C;MACE
,MAAM,KAAC,CAAC,kBAAD,C;WACN,IAAI,IAAI,OAAO,EAaf,C;MACL,OAAO,MAAM,KAAC,K;;IAGpB,
IAAI,IAAI,WAAW,CAAC,MAAM,KAAC,UAAZ,CAAnB,C;MACE,IAAI,KAAC,WAAW,CAAC,MAAM,KAAC,
IAAZ,CAakB,IACIC,KAAC,WAAW,CAAC,MAAM,KAAC,QAaz,CADpB,C;QAEE,OAAO,MAAM,KAAC,
U;aAcB,IAAI,KAAC,WAAW,CAAC,MAAM,KAAC,UAAZ,CAApB,C;QACL,OAAO,MAAM,KAAC,I;;QAGIB,
IAAI,WAAW,IAAI,WAAW,CAAC,CAAD,C;QAC9B,IAAI,SAAS,QAaQ,IAAI,CAAC,KAAD,CAAO,UAAU,C
AAC,CAAD,C;QAC1C,IAAI,MAAM,WAAW,CAAC,MAAM,KAAC,KAaz,CAArB,C;UACE,OAAO,KAAC,W
AAW,EAAG,GAAE,MAAM,KAAC,IAAb,GAAoB,MAAM,KAAC,Q;;UAEdD,IAAI,MAAM,IAAI,SAAS,CAAC,
KAAC,SAAS,CAAC,MAAD,CAaf,C;UACvB,IAAI,SAAS,MAAM,IAAI,CAAC,GAAG,IAAI,CAAC,KAAD,CA
AR,C;UACvB,OAAO,M;;WAGN,IAAI,KAAC,WAAW,CAAC,MAAM,KAAC,UAAZ,CAApB,C;MACL,OAAO
,MAAM,KAAC,K;;IAGpB,IAAI,IAAI,WAAW,EAAnB,C;MACE,IAAI,KAAC,WAAW,EAAPB,C;QACE,OAAO,
IAAI,OAAO,EAAC,IAAI,CAAC,KAAC,OAAO,EAAb,C;;QAExB,OAAO,IAAI,OAAO,EAAC,IAAI,CAAC,KA
AD,CAAO,OAAO,E;;WAEnC,IAAI,KAAC,WAAW,EAAPB,C;MACL,OAAO,IAAI,IAAI,CAAC,KAAC,OAAO,E
AAb,CAAgB,OAAO,E;;IAQxC,IAAI,MAAM,MAAM,KAAC,K;IACrB,IAAI,MAAM,I;IACV,OAAO,GAAG,mB

AaMB,CAAC,KAAD,CAA7B,C;MAGE,IAAI,SAAS,IAAI,IAAI,CAAC,CAAD,EAAl,IAAI,MAAM,CAAC,GAA
G,SAAS,EAAG,GAAE,KAaK,SAAS,EAaHc,CAAd,C;MAIrb,IAAI,OAAO,IAAI,KAaK,CAAC,IAAI,IAAI,CA
AC,MAAD,CAAS,GAAE,IAAI,IAAxB,C;MACpb,IAAI,QAAS,IAAK,IAAG,EAAl,GAAE,CAAF,GAAM,IAAI,I
AAI,CAAC,CAAD,EAAl,IAAK,GAAE,EAAX,C;MAIvc,IAAI,YAAy,MAAM,KAaK,WAaW,CAAC,MAAD,C
;MACtc,IAAI,YAAy,SAAS,SAAS,CAAC,KAAD,C;MACIC,OAAO,SAAS,WAaW,EAAG,IAAG,SAAS,YAAy,
CAAC,GAAD,CAAtD,C;QACE,MAAO,IAAG,K;QACV,SAAU,GAAE,MAAM,KAaK,WAaW,CAAC,MAAD,C
;QACIC,SAAU,GAAE,SAAS,SAAS,CAAC,KAAD,C;;MAKhC,IAAI,SAAS,OAAO,EAAPb,C;QACE,SAAU,GA
AE,MAAM,KAaK,I;;MAGzB,GAAl,GAAE,GAAG,IAAI,CAAC,SAAD,C;MACb,GAAl,GAAE,GAAG,SAAS,C
AAC,SAAD,C;;IAEpB,OAAO,G;G;EAST,MAAM,KAaK,UAAU,OAAQ,GAAE,iB;IAC7B,OAAO,IAAI,SAAS,C
AAC,IAAI,IAAI,CAAC,KAAD,CAAO,SAAS,CAAC,KAAD,CAAzB,C;G;EAKtB,MAAM,KAaK,UAAU,IAAK,
GAAE,Y;IAC1B,OAAO,MAAM,KAaK,SAAS,CAAC,CAAC,IAAI,KAAN,EAaA,CAAC,IAAI,MAAIB,C;G;EAS
7B,MAAM,KAaK,UAAU,IAAK,GAAE,iB;IAC1B,OAAO,MAAM,KAaK,SAAS,CAAC,IAAI,KAAM,GAAE,K
AAK,KAaIB,EACI,IAAI,MAAO,GAAE,KAaK,MADtB,C;G;EAU7B,MAAM,KAaK,UAAU,GAAl,GAAE,iB;I
ACzB,OAAO,MAAM,KAaK,SAAS,CAAC,IAAI,KAAM,GAAE,KAaK,KAaIB,EACI,IAAI,MAAO,GAAE,KA
AK,MADtB,C;G;EAU7B,MAAM,KAaK,UAAU,IAAK,GAAE,iB;IAC1B,OAAO,MAAM,KAaK,SAAS,CAAC,I
AAI,KAAM,GAAE,KAaK,KAaIB,EACI,IAAI,MAAO,GAAE,KAaK,MADtB,C;G;EAU7B,MAAM,KAaK,UA
AU,UAAW,GAAE,mB;IAChC,OAAQ,IAAG,E;IACX,IAAI,OAAQ,IAAG,CAAf,C;MACE,OAAO,I;;MAEP,IAAI
,MAAM,IAAI,K;MACd,IAAI,OAAQ,GAAE,EAAd,C;QACE,IAAI,OAAO,IAAI,M;QACf,OAAO,MAAM,KAaK
,SAAS,CACvB,GAAl,IAAG,OADgB,EAEtB,IAAK,IAAG,OAAS,GAAG,GAAl,KAaK,EAAG,GAAE,OFZ,C;;
QAI3B,OAAO,MAAM,KAaK,SAAS,CAAC,CAAD,EAAl,GAAl,IAAI,OAAQ,GAAE,EAAtB,C;;G;EAWjC,MA
AM,KAaK,UAAU,WAAY,GAAE,mB;IACjC,OAAQ,IAAG,E;IACX,IAAI,OAAQ,IAAG,CAAf,C;MACE,OAAO,
I;;MAEP,IAAI,OAAO,IAAI,M;MACf,IAAI,OAAQ,GAAE,EAAd,C;QACE,IAAI,MAAM,IAAI,K;QACd,OAAO,
MAAM,KAaK,SAAS,CACtB,GAAl,KAAl,OAAS,GAAG,IAAK,IAAI,EAAG,GAAE,OADZ,EAEvB,IAAK,IAA
G,OFa,C;;QAI3B,OAAO,MAAM,KAaK,SAAS,CACvB,IAAK,IAAI,OAAQ,GAAE,EADI,EAEvB,IAAK,IAAG
,CAAE,GAAE,CAAF,GAAM,EAFO,C;;G;EAejC,MAAM,KAaK,UAAU,mBAaOB,GAAE,mB;IACzC,OAAQ,I
AAG,E;IACX,IAAI,OAAQ,IAAG,CAAf,C;MACE,OAAO,I;;MAEP,IAAI,OAAO,IAAI,M;MACf,IAAI,OAAQ,G
AAE,EAAd,C;QACE,IAAI,MAAM,IAAI,K;QACd,OAAO,MAAM,KAaK,SAAS,CACtB,GAAl,KAAl,OAAS,GA
AG,IAAK,IAAI,EAAG,GAAE,OADZ,EAEvB,IAAK,KAAl,OFa,C;aAgTB,IAAI,OAAQ,IAAG,EAaf,C;QACL,
OAAO,MAAM,KAaK,SAAS,CAAC,IAAD,EAaO,CAAP,C;;QAE3B,OAAO,MAAM,KAaK,SAAS,CAAC,IAA
K,KAaK,OAAQ,GAAE,EAArB,EAaOB,CAAIB,C;;G;EAMjC,MAAM,KAaK,UAAU,OAAQ,GAAE,iB;IAC3B,
OAAO,KAAM,YAAW,MAAM,KAAM,IAAG,IAAI,WAaW,CAAC,KAAD,C;G;EAGID,MAAM,KAaK,UAAU,
gBAaIB,GAAE,MAAM,KAaK,UAAU,Q;EAE7D,MAAM,KAaK,UAAU,IAAK,GAAE,Y;IACxB,OAAO,IAAI,I
AAI,CAAC,MAAM,KAaK,IAAZ,C;G;EAGnB,MAAM,KAaK,UAAU,IAAK,GAAE,Y;IACxB,OAAO,IAAI,IAA
I,CAAC,MAAM,KAaK,QAaZ,C;G;EAGnB,MAAM,KAaK,UAAU,QAAS,GAAE,Y;IAC5B,OAAO,IAAI,SAAS
,E;G;EAGxB,MAAM,KAaK,UAAU,UAAW,GAAE,Y;IAC9B,OAAO,I;G;EAGX,MAAM,KAaK,UAAU,WAAY,
GAAE,MAAM,KAaK,UAAU,O;EACxD,MAAM,KAaK,UAAU,IAAK,GAAE,MAAM,KAaK,UAAU,I;EAEjD,
MAAM,KAaK,UAAU,QAAS,GAAE,iB;IAC5B,OAAO,IAAI,MAAM,OAAO,OAAO,UAAxB,CAAmC,IAAnC,E
AAyC,KAaZC,C;G;EC1zBX,MAAM,aAAc,GAAE,2B;G;EAGtB,MAAM,qBAAsB,GAAE,oB;IAC1B,OAAO,G;
G;EAGX,MAAM,aAAc,GAAE,e;IACIB,IAAI,IAAI,Y;MACJ,CAAE,GAAE,GAAG,E;MACP,OAAO,CAAC,MA
AM,CAAC,IAAD,EAaO,SAAP,C;K;IAEIB,OAAO,Y;MACH,OAAO,CAAC,MAAM,CAAC,IAAD,EAaO,SAAP
,C;K;G;EAItB,MAAM,SAAU,GAAE,gB;IACd,OAAO,kB;MACH,OAAO,OAAO,MAAO,KAAl,I;K;G;EAIjC,MA
AM,aAAc,GAAE,iB;IACIB,OAAO,kB;MACH,OAAO,MAAM,OAAO,CAAC,MAAD,EAAS,KAAT,C;K;G;EAI5
B,MAAM,OAAQ,GAAE,c;IACZ,OAAO,kB;MACH,OAAO,MAAO,IAAG,IAAK,IAAG,EAaE,CAAC,MAAD,C;
K;G;EAIInC,MAAM,aAAc,GAAE,gB;IACIB,OAAO,kB;MACH,OAAO,CAAC,CAAC,MAAD,CAAS,IAAG,CAA
C,CAAC,MAAD,C;K;G;EAI7B,MAAM,qBAAsB,GAAE,wC;G;EAG9B,MAAM,YAAa,GAAE,iB;IACjB,OAAO,
K;G;EAGX,MAAM,gBAaIB,GAAE,qB;IACrB,gBAAgB,E;G;EAGpB,MAAM,oBAaQb,GAAE,qB;IACzB,gBA
AgB,E;G;EAGpB,MAAM,kBAaMB,GAAE,qB;IACvB,gBAAgB,E;G;EAGpB,MAAM,mBAaOB,GAAE,4B;IACx
B,gBAAgB,E;G;EAGpB,MAAM,6BAa8B,GAAE,yB;IACIC,gBAAgB,E;G;EAGpB,4B;IACI,MAAM,IAAI,KAaJ

,CACF,iDAaKd,GACID,qDAAsD,GACtD,uDAHE,C;G;EAMV,MAAM,gBAaiB,GAAE,4B;IACrB,OAAO,Y;MACH,OAAO,Y;K;G;ECJFf,MAAM,UAAW,GAAE,gB;IACf,IAAI,QAAQ,OAAO,C;IACnB,IAAI,KAAM,KAAL,QAAd,C;MACI,IAAI,OAAO,CAAE,KAAL,QAAjB,C;QACI,OAAO,MAAM,gBAAgB,CAAC,CAAD,EAAI,CAAJ,C;;MAEjC,OAAO,MAAM,mBAAmB,CAAC,CAAD,EAAI,CAAJ,C;;IAEpC,IAAI,KAAM,KAAL,QAAS,IAAG,KAAAM,KAAL,SAAP,C,C;MACI,OAAO,MAAM,mBAAmB,CAAC,CAAD,EAAI,CAAJ,C;;IAEpC,OAAO,CAAC,gBAAgB,CAAC,CAAD,C;G;EAG5B,MAAM,mBAaOB,GAAE,gB;IACxB,OAAO,CAAE,GAAE,CAAE,GAAE,EA AF,GAAO,CAAE,GAAE,CAAE,GAAE,CAAF,GAAM,C;G;EAGpC,MAAM,gBAaiB,GAAE,gB;IACrB,IAAI,CAAE,GAAE,CAAR,C;MAAW,OAAO,E;IACIB,IAAI,CAAE,GAAE,CAAR,C;MAAW,OAAO,C;IAEIB,IAAI,CAAE,KAAL,CAAV,C;MACI,IAAI,CAAE,KAAL,CAAV,C;QAAa,OAAO,C;MAEpB,IAAI,KAAL,CAAE,GAAE,C;MACb,OAAO,EAAG,KAAL,CAAE,GAAE,CAAE,GAAE,CAAF,GAAO,EAAG,GAAE,CAAE,GAAE,EA AF,GA AO,C;;IAG7C,OAAO,CAAE,KAAL,CAAE,GAAG,CAAE,KAAL,CAAE,GAAE,CAAF,GAAM,CAAJB,GAASB,E;G;EAGzC,MAAM,QAAS,GAAE,iB;IACb,OAAO,MAAM,OAAO,CAAC,KAAL,GAAC,CAAP,C;G;EAGxB,MAAM,QAAS,GAAE,iB;IACb,OAAO,MAAM,OAAO,CAAC,KAAL,GAAC,CAAP,C;G;EAGxB,MAAM,KAAM,GAAE,IAAI,KAAM,IAAG,I;EAE3B,MAAM,aAAc,GAAE,I;EAEtB,oB;IACI,OAAyB,CAAhB,CAAE,GAAE,YAA Y,KAAG,CAAE,GAAE,KAAP,CAAe,GAAe,CAAZ,CAAE,GAAE,KAAQ,KAAG,CAAE,GAAE,CAAP,CAAW,GAAE,C;G;EA6DtE,CAIDD,Y;IACG,IAAI,MAAM,IAAI,WAAJ,CAAgB,CAAhB,C;IACV,IAAI,aAAa,IAAI,YAAJ,CAAiB,GAAjB,C;IACjB,IAAI,aAAa,IAAI,YAAJ,CAAiB,GAAjB,C;IACjB,IAAI,WAAW,IAAI,UAAJ,CAAe,GA Af,C;IACf,IAAI,WAAW,C;IACf,IAAI,YAA Y,C;IAEHb,UAAU,CAAC,CAAD,CAAI,GAAE,E;IACHb,IAAI,QAAQ,CAAC,QAAD,CAAW,KAAL,CAA3B,C;MACI,QAAS,GAAE,C;MACX,SAAU,GAAE,C;;IAGhB,MAAM,aAAc,GAAE,iB;MACIB,OAAO,MAAM,gBAAgB,CAAC,KAAL,CAAC,KAAD,CAAQ,GAAE,GA AF,GAAQ,KAAtB,C;K;IAGjC,MAAM,gBAaiB,GAAE,iB;MACrB,UAAU,CAAC,CAAD,CAAI,GAAE,K;MACHb,OAAO,MAAM,KAAL,SAAS,CAAC,QAAQ,CAAC,QAAD,CAAT,EAAqB,QAAQ,CAAC,SAAD,CAA7B,C;K;IAG/B,MAAM,eAAgB,GAAE,iB;MACpB,QAAQ,CAAC,QAAD,CAAW,GAAE,KAAL,K;MAC1B,QAAQ,CAAC,SAAD,CAAY,GAAE,KAAL,M;MAC3B,OAAO,UAAU,CAAC,CAAD,C;K;IAGrB,MAAM,YAAa,GAAE,iB;MACjB,OAAO,MAAM,eAAe,CAAC,KAAL,CAAC,KAAD,CAAQ,GAAE,GA AF,GAAQ,KAAtB,C;K;IAGhC,MAAM,eAAgB,GAAE,iB;MACpB,UAAU,CAAC,CAAD,CAAI,GAAE,K;MACHb,OAAO,QAAQ,CAAC,CAAD,C;K;IAGnB,MAAM,cAAe,GAAE,iB;MACnB,QAAQ,CAAC,CAAD,CAAI,GAAE,K;MACd,OAAO,UAAU,CAAC,CAAD,C;K;IAIrB,MAAM,cAAe,GAAE,iB;MACnB,UAAU,CAAC,CAAD,CAAI,GAAE,K;MACHb,OAAO,QAAQ,CAAC,SAAD,CAAY,GAAE,a;K;IAGjC,MAAM,eAAgB,GAAE,e;MACpB,IAAc,CAAT,GAAL,GAAE,CAAG,MAAL,GAALB,C;QACI,OAAO,GAAL,GAAE,C;;QAGb,UAAU,CAAC,CAAD,CAAI,GAAE,G;QACHb,OAAc,CAA9B,QAAQ,CAAC,SAAD,CAAY,GAAE,EAAG,GAAE,CAAG,IAAE,QAAQ,CAAC,QAAD,CAAW,GAAE,C;;K;GAGvE,G;EA EF,MAAM,cAAe,GAAE,a;IACnB,OAAO,CAAE,IAAG,IAAK,GAAE,CAAF,GAAM,MAAM,SAAS,E;G;EC7G1C,IAAI,OAAO,MAAM,UAAU,WAA Y,KAAL,WAA3C,C;IACI,MAAM,eAAe,CAAC,MAAM,UAAP,EAAmB,YAAAnB,EAAiC,QAC3C,kC;MACH,QAAS,GAAE,QAAS,IAAG,C;MACvB,OAAO,IAAI,YAA Y,CAAC,YAAD,EAAe,QA Af,CAAyB,KAAL,Q;KAHN,CAAJ,C,C;;EAOzB,IAAI,OAAO,MAAM,UAAU,SAAU,KAAL,WAAzC,C;IACI,MAAM,eAAe,CAAC,MAAM,UAAP,EAAmB,YAAAnB,EAA+B,QACzC,kC;MACH,IAAI,gBAAgB,IAAI,SAAS,E;MACjC,IAAI,QAAS,KAAL,SAAU,IAAG,QAAS,GAAE,aAAa,OAAtD,C;QACI,QAAS,GAAE,aAAa,O;;MAE5B,QAAS,IAAG,YAA Y,O;MACxB,IAAI,YAA Y,aAAa,QAAQ,CAAC,YAAD,EAAe,QA Af,C;MACrC,OAAO,SAAU,KAAL,EAAG,IAAG,SAAU,KAAL,Q;KARG,CAA/B,C;;EAazB,IAAI,OAAO,IAAI,KAAM,KAAL,WAAzB,C;IACI,IAAI,KAAM,GAAE,a;MACR,CAAE,GAAE,CAAC,C;MACL,IAAI,CAAE,KAAL,CAAE,IAAG,KAAL,CAAC,CAAD,CAApB,C;QACI,OAAO,MAAM,CAAC,CAAD,C;;MAEjB,OAAO,CAAE,GAAE,CAAE,GA AE,CAAF,GAAM,E;K;;EAG3B,IAAI,OAAO,IAAI,MAAO,KAAL,WAA1B,C;IACI,IAAI,MAAO,GAAE,a;MACT,IAAI,KAAL,CAAC,CAAD,CAAT,C;QACI,OAAO,G;;MAEX,IAAI,CAAE,GAAE,CAAR,C;QACI,OAAO,IAAI,MAAM,CAAC,CAAD,C;;MAErB,OAAO,IAAI,KAAL,CAAC,CAAD,C;K;;EAuKtB,CAnKD,Y;IACG,IAAI,UAAU,qB;IACd,IAAI,iBAaiB,IAAI,KAAL,CAAC,OAAD,C;IAC9B,IAAI,iBAaiB,IAAI,KAAL,CAAC,cAAD,C;IAC9B,IAAI,uBAauB,CAAC,GAAC,c;IAC7B,IAAI,uBAauB,CAAC,GAAC,c;IAE7B,IAAI,OAAO,IAAI,KAAM,KAAL,WAAzB,C;MACI,IAAI,KAAM,GAAE,a;QACR,IAAI,IAAI,IAAI,CAAC,CAAD,CAAI,GAAE,cAAIB,C;UACI,IAAI,SAAS,C;UACb,IAAI,IAAI,IAAI,CAAC,CAAD,CAAI,GAAE,cAAIB,C;YACI,MAAO,IAAI,CAAE,GAA

Q,KAAl,WAAIC,C;IACI,WAAW,OAAQ,GAAE,a;MACjB,OAAO,CAAE,IAAG,IAAK,IAAG,CAAC,UAAW,IAAG,IAAK,IAAG,CAAC,UAAU,UAAW,KAAI,SAAS,UAAU,U;K;:EAIhG,IAAI,OAAO,KAAK,UAAU,KAAM,KAAI,WAApC,C;IAEI,MAAM,eAAe,CAAC,KAAK,UAAU,EAakB,MAAIB,EAA0B,QACpC,iB;MAGH,IAAI,IAAK,IAAG,IAAZ,C;QACI,MAAM,IAAI,SAAJ,CAAc,6BAAd,C;:MAGV,IAAI,IAAI,MAAM,CAAC,IAAD,C;MAGd,IAAI,MAAM,CAAC,OAAQ,KAAI,C;MAGvB,IAAI,QAAQ,SAAS,CAAC,CAAD,C;MACrB,IAAI,gBAAgB,KAAM,IAAG,C;MAG7B,IAAI,IAAI,aAAc,GAAE,CAAE,GACIB,IAAI,IAAI,CAAC,GAAI,GAAE,aAAP,EAA5B,CAAtB,CADU,GAEIB,IAAI,IAAI,CAAC,aAAD,EAAGB,GAAhB,C;MAGhB,IAAI,MAAM,SAAS,CAAC,CAAD,C;MACnB,IAAI,cAAc,GAAI,KAAI,SAAU,GACIB,GADkB,GACZ,GAAI,IAAG,C;MAG/B,IAAI,aAAa,WAAy,GAAE,CAAE,GACHB,IAAI,IAAI,CAAC,GAAI,GAAE,WAAP,EAAoB,CAApB,CADQ,GAehB,IAAI,IAAI,CAAC,WAAD,EAAC,GAAc,C;MAGzB,OAAO,CAAE,GAAE,UAAx,C;QACI,CAAC,CAAC,CAAD,CAAI,GAAE,K;QACP,CAAC,E;:MAIL,OAAO,C;KAvCgC,CAA1B,C;:EA4HvB,CAhFD,Y;IACG,yC;MACI,IAAI,MAAO,GAAE,CAAb,C;QAAgB,OAAO,IAAI,IAAI,CAAC,CAAD,EAAI,MAAO,GAAE,MAAb,C;MAC/B,OAAO,IAAI,IAAI,CAAC,MAAD,EAAS,MAAT,C;K;IAEnB,qC;MACI,IAAI,OAAO,GAAI,KAAI,WAAhB,C;QACI,GAAI,GAAE,IAAI,O;:MAEd,KAAM,GAAE,eAAe,CAAC,KAAM,IAAG,CAAV,EAAa,IAAI,OAAjB,C;MACvB,GAAI,GAAE,IAAI,IAAI,CAAC,KAAD,EAAQ,eAAe,CAAC,GAAD,EAAM,IAAI,OAAV,CAAvB,C;MACd,OAAO,IAAI,IAAI,YAAR,CAAqB,IAAI,SAAS,CAAC,KAAD,EAAQ,GAAR,CAAIC,C;K;IAGX,IAAI,SAAS,CAAC,SAAD,EAAY,UAAZ,EAAwB,WAAxB,EAAqC,UAArC,EAAiD,YAAjD,EAA+D,YAA/D,C;IACb,KAAK,IAAI,IAAI,CAAb,EAAGB,CAAE,GAAE,MAAM,OAA1B,EAAMC,EAAE,CAArC,C;MACI,IAAI,aAAa,MAAM,CAAC,CAAD,C;MACvB,IAAI,OAAO,UAAU,UAAU,KAAM,KAAI,WAAzC,C;QACI,MAAM,eAAe,CAAC,UAAU,UAAx,EAauB,MAAvB,EAA+B,QACzC,KAAK,UAAU,KAD0B,CAA/B,C;:MAIzB,IAAI,OAAO,UAAU,UAAU,MAAO,KAAI,WAA1C,C;QACI,MAAM,eAAe,CAAC,UAAU,UAAx,EAauB,OAAvB,EAAGC,QAC1C,eAD0C,CAAhC,C;:MAQJ,CAApB,Y;OAAc,MAAM,CAAC,IAAD,EAAO,IAAI,UAAJ,CAAE,CAAF,CAAP,E;:MAErB,IAAI,QAAQ,QAAQ,UAAU,M;MAC9B,MAAM,eAAe,CAAC,QAAQ,UAAJ,EAAGB,OAArB,EAA8B,QACxC,uB;QACH,OAAO,KAAK,KAAK,CAAC,IAAD,EAAO,IAAP,EAAa,EAAE,MAAM,KAAK,CAAC,KAAD,CAA1B,C;OAF0B,CAA9B,C;:IASzB,KAAK,IAAI,IAAI,CAAb,EAAGB,CAAE,GAAE,MAAM,OAA1B,EAAMC,EAAE,CAArC,C;MACI,IAAI,aAAa,MAAM,CAAC,CAAD,C;MACvB,IAAI,OAAO,UAAU,UAAU,IAAK,KAAI,WAAxC,C;QACI,MAAM,eAAe,CAAC,UAAU,UAAx,EAauB,KAAvB,EAA8B,QACxC,0B;UACH,OAAO,EAAE,MAAM,KAAK,CAAC,IAAD,CAAM,IAAI,CAAC,QAAD,EAAW,IAAX,C;SAFa,CAA9B,C;:IAU7B,IAAI,uBAAuB,gB;MACvB,IAAI,CAAE,GAAE,CAAR,C;QAAW,OAAO,E;MACIB,IAAI,CAAE,GAAE,CAAR,C;QAAW,OAAO,C;MAEIB,IAAI,CAAE,KAAI,CAAV,C;QACI,IAAI,CAAE,KAAI,CAAV,C;UAAa,OAAO,C;QAEpB,IAAI,KAAK,CAAE,GAAE,C;QACb,OAAO,EAAG,KAAI,CAAE,GAAE,CAAE,GAAE,CAAF,GAAO,EAAG,GAAE,CAAE,GAAE,EAAF,GAAO,C;:MAG7C,OAAO,CAAE,KAAI,CAAE,GAAG,CAAE,KAAI,CAAE,GAAE,CAAF,GAAM,CAAjB,GAAsB,E;K;IAGzC,KAAK,IAAI,IAAI,CAAb,EAAGB,CAAE,GAAE,MAAM,OAA1B,EAAMC,EAAE,CAArC,C;MACI,IAAI,aAAa,MAAM,CAAC,CAAD,C;MACvB,IAAI,OAAO,UAAU,UAAU,KAAM,KAAI,WAAzC,C;QACI,MAAM,eAAe,CAAC,UAAU,UAAx,EAauB,MAAvB,EAA+B,QACzC,2B;UACH,OAAO,KAAK,UAAU,KAAK,KAAK,CAAC,IAAD,EAAO,eAAgB,IAAG,oBAA1B,C;SAFY,CAA/B,C;:GAO/B,G;ECxXF,MAAM,KAAM,GAAE,QACH,OADG,aAEC,WAFD,UAGF,QAHE,C;EAMd,MAAM,WAAy,GAAE,2C;IACHB,IAAI,qBAAqB,MAAM,yBAAyB,CAAC,KAAD,EAAQ,YAAR,C;IACxD,IAAI,kBAAmB,IAAG,IAAK,IAAG,kBAakB,IAAK,IAAG,IAA5D,C;MACI,OAAO,kBAakB,IAAI,KAAK,CAAC,UAAD,C;:IAGtC,kBAAmB,GAAE,MAAM,yBAAyB,CAAC,UAAD,EAAa,YAAb,C;IACpD,IAAI,kBAAmB,IAAG,IAAK,IAAG,OAAQ,IAAG,kBAA7C,C;MACI,OAAO,UAAU,CAAC,YAAD,C;:IAGrB,OAAO,MAAM,WAAW,CAAC,UAAD,EAAa,MAAM,eAAe,CAAC,KAAD,CAAIC,EAA2C,YAA3C,C;G;EAG5B,MAAM,WAAy,GAAE,kD;IACHB,IAAI,qBAAqB,MAAM,yBAAyB,CAAC,KAAD,EAAQ,YAAR,C;IACxD,IAAI,kBAAmB,IAAG,IAAK,IAAG,kBAakB,IAAK,IAAG,IAA5D,C;MACI,kBAakB,IAAI,KAAK,CAAC,UAAD,EAAa,KAAb,C;MAC3B,M;:IAGJ,kBAAmB,GAAE,MAAM,yBAAyB,CAAC,UAAD,EAAa,YAAb,C;IACpD,IAAI,kBAAmB,IAAG,IAAK,IAAG,OAAQ,IAAG,kBAA7C,C;MACI,UAAU,CAAC,YAAD,CAAE,GAAE,K;MAC3B,M;:IAGJ,MAAM,WAAW,CAAC,UAAD,EAAa,MAAM,eAAe,CAAC,KAAD,CAAIC,EAA2C,YAA3C,EAAyD,KAAzD,C;G;EAGrB,iD;IACI,IAAI,IAAK,KAAI,KAAb,C;MAAoB,OAAO,I;IAE3B,IAAI,WAAW,IAAI,W;IACnB,IAAI,QAAS,IAAG,IAAhB,C;MACI,IAAI,aAAa,QAAQ,W;MACzB,KAAK,IAAI,IAAI,CA

Ab,EAAgB,CAAE,GAAE,UAAU,OAA9B,EAAuC,CAAC,EAAxC,C;QACI,IAAI,0BAA0B,CAAC,UAAU,CAAC
,CAAD,CAAX,EAAgB,KAAhB,CAA9B,C;UACI,OAAO,I;;;IAKnB,IAAI,iBAAiB,IAAI,UAAW,IAAG,IAAK,G
AAE,MAAM,eAAe,CAAC,IAAI,UAAU,CAAxB,GAA0C,I;IACtF,IAAI,mBAAmB,cAAe,IAAG,IAAK,GAAE,cA
Ac,YAAhB,GAA+B,I;IAC7E,OAAO,gBAAiB,IAAG,IAAK,IAAG,0BAA0B,CAAC,gBAAD,EAAmB,KAAhB,C;
G;EASjE,MAAM,OAAQ,GAAE,yB;IACZ,IAAI,KAAM,KAAl,MAAd,C;MACI,QAAQ,OAAO,MAAf,C;aACS,Q;
aACA,Q;aACA,S;aACA,U;UACD,OAAO,I;UAEP,OAAO,MAAO,YAAW,M;;;IAIrC,IAAI,MAAO,IAAG,IAAK,
IAAG,KAAM,IAAG,IAAK,KAAl,OAAO,MAAO,KAAl,QAAS,IAAG,OAAO,MAAO,KAAl,UAApD,CAApC,C;
MACI,OAAO,K;;IAGX,IAAI,OAAO,KAAM,KAAl,UAAW,IAAG,MAAO,YAAW,KAArD,C;MACI,OAAO,I;IA
GX,IAAI,QAAQ,MAAM,eAAe,CAAC,KAAD,C;IACjC,IAAI,cAAc,KAAM,IAAG,IAAK,GAAE,KAAK,YAAP,
GAAsB,I;IACtD,IAAI,WAAy,IAAG,IAAK,IAAG,YAAa,IAAG,WAA3C,C;MACI,IAAI,WAAW,WAAW,W;MA
C1B,IAAI,QAAQ,KAAM,KAAl,MAAM,KAAK,OAAjC,C;QACI,OAAO,MAAO,KAAl,K;;;IAI1B,IAAI,gBAAg
B,KAAK,W;IAGzB,IAAI,aAAc,IAAG,IAArB,C;MACI,OAAO,MAAO,YAAW,K;;IAG7B,IAAI,aAAa,KAAM,K
AAI,MAAM,KAAK,UAAW,IAAG,MAAM,YAAa,IAAG,IAA1E,C;MACI,OAAO,0BAA0B,CAAC,MAAM,YAA
P,EAAqB,KAArB,C;;IAGrC,OAAO,K;G;EAGX,MAAM,SAAU,GAAE,a;IACd,OAAO,OAAO,CAAE,IAAG,QA
AS,IAAG,CAAE,YAAW,MAAM,K;G;EAGtD,MAAM,OAAQ,GAAE,iB;IACZ,OAAO,KAAM,YAAW,MAAM,
U;G;EAGIC,MAAM,aAAc,GAAE,iB;IACIB,IAAI,OAAO,OAAO,K;IAEIB,OAAO,IAAK,KAAl,QAAS,IACIB,IA
AK,KAAl,SAAU,IACnB,MAAM,SAAS,CAAC,KAAD,CAAQ,IACvB,MAAM,OAAO,CAAC,KAAD,EAAQ,MA
AM,OAAO,WAArB,C;G;EAGxB,MAAM,eAAgB,GAAE,iB;IACpB,OAAO,OAAO,KAAM,KAAl,QAAS,IAAG,
MAAM,OAAO,CAAC,KAAD,EAAQ,MAAM,OAAO,aAArB,C;G;;;;;aCnDV,gB;;;ICrE3C,gB;MAkBI,4B;MA
jBA,aAA6C,E;MAC7C,gBAAgD,C;K;4EAG5C,Y;MAAQ,iB;K;+EAGR,Y;MAAQ,oB;K;qCAEZ,iB;MAAyC,OA
AQ,0BAAR,YAAQ,EAAU,KAAM,QAAhB,C;K;4BAEjD,iB;MAAmC,gBAAS,K;K;8BAE5C,Y;MAA+B,OAAnc
,MAAmC,kBAA8B,IAA9B,C;K;8BAE/B,Y;MAA0B,gB;K;IAE1B,0B;MAAA,8B;K;;;IAAA,sC;MAAA,qC;QAA
A,oB;;MAAA,8B;K;;IDfJ,mC;MAC4C,oBAAa,MAAS,IAAT,CAAb,EAA6B,SAAT7B,C;K;gEAE5C,yB;MAAA,m
B;MAAA,6B;QAC2D,YAAa,QAAS,IAAT,C;QAIvD,Q;QAAA,OAAA,KAAM,OAAN,GAAa,CAAb,I;QAAb,aAA
U,CAAV,iB;UACI,MAAM,CAAN,IALgF,IAKrE,CAAK,CAAL,C;;QALwC,OAOhD,K;O;KARX,C;gEAGA,uB;
MAEiB,Q;MAAA,OAAA,KAAM,OAAN,GAAa,CAAb,I;MAAb,aAAU,CAAV,iB;QACI,MAAM,CAAN,IAAW,
KAAK,CAAL,C;;MAEf,OAAO,K;K;IAGX,kC;MAIiB,IAAN,I;MAFP,aAAsB,MAAe,IAAf,C;MACtB,gBAAkB,c;
MAEd,IADS,IACt,mBADS,IACt,EAAM,IAAN,E;QAAc,oBAAa,MAAb,EAAqB,KAArB,C;WACd,WAFS,IAET
,S;QAAS,a;;QAZA,U;QAAA,SAaqB,Mabf,OAAN,GAAa,CAAb,I;QAAb,aAAU,CAAV,mB;UAakC,MAZ9B,CA
AM,CAAN,IAyS,C,IAZ3B,CAAK,CAAL,C;;QAYH,OAAsB,M;;MAHIC,W;K;2EAOJ,yB;MAAA,iC;MAAA,6B;
QACoF,YAAa,aAAa,IAAb,EAAmB,KAAhB,C;QAIbHf,Q;QAAA,OAAA,KAAM,OAAN,GAAa,CAAb,I;QAAb,a
AAU,CAAV,iB;UACI,MAAM,CAAN,IAiBoH,IAjBzG,CAAK,CAAL,C;;QAIbIE,OafzE,K;O;KAcX,C;IAGA,+B
;MAKiB,IAAN,I;MAFP,aAAa,IAAb,WAAa,CAAD,IAAC,C;MACb,gBAAkB,W;MAEd,IADS,IACt,mBADS,IA
CT,EAAM,IAAN,YADS,IACt,EAAY,KAAZ,E;QAAqB,a;;QA1BZ,U;QAAA,SA2BkB,MA3BZ,OAAN,GAAa,C
AAb,I;QAAb,aAAU,CAAV,mB;UA2B+B,MA1B3B,CAAM,CAAN,IA0BmC,IA1BxB,CAAK,CAAL,C;;QA0BH,
OAAmB,M;;MAF/B,W;K;qEAMJ,yB;MAAA,2B;MAAA,gC;MAAA,6B;QAGiB,Q;QADb,YAAy,UAAU,IAAV,
EAAgB,IAAhB,C;QACC,OAAA,KAAM,OAAN,GAAa,CAAb,I;QAAb,aAAU,CAAV,iB;UACI,YACY,eAAK,CA
AL,E;UACpB,KAAK,CAAC,CAAD,CAAG,GAAG,K;;QAEP,OAAO,K;O;KARX,C;mFAWA,yB;MAAA,mB;M
AAA,gC;MAAA,6B;QAGiB,Q;QADb,YAAy,QAAY,IAAZ,C;QACC,OAAA,KAAM,OAAN,GAAa,CAAb,I;QA
Ab,aAAU,CAAV,iB;UACI,YACY,eAAK,CAAL,E;UACpB,KAAK,CAAC,CAAD,CAAG,GAAG,K;;QAEP,OAA
O,K;O;KARX,C;IAWA,+B;MAIiB,IAAN,I;MAFP,aAAsB,MAAY,IAAZ,C;MACtB,gBAAkB,W;MAEd,IADS,IA
CT,mBADS,IACt,EAAM,IAAN,E;QAAc,oBAAa,MAAb,K;WACd,WAFS,IAET,S;QAAS,a;;QA3DA,U;QAAA,S
A4DkB,MA5DZ,OAAN,GAAa,CAAb,I;QAAb,aAAU,CAAV,mB;UA4D+B,MA3D3B,CAAM,CAAN,IA2DmC,I
A3DxB,CAAK,CAAL,C;;QA2DH,OAAmB,M;;MAH/B,W;K;qEAOJ,yB;MAAA,2B;MAAA,6B;QAC2E,YAAa,U
AAU,IAAV,EAAgB,KAAhB,C;QAjEvE,Q;QAAA,OAAA,KAAM,OAAN,GAAa,CAAb,I;QAAb,aAAU,CAAV,iB
;UACI,MAAM,CAAN,IAgEwG,IAhE7F,CAAK,CAAL,C;;QAgEwD,OA9DhE,K;O;KA6DX,C;IAGA,wC;MACiB
,Q;MAAA,OAAA,KAAM,OAAN,GAAa,CAAb,I;MAAb,aAAU,CAAV,iB;QACI,MAAM,CAAN,IAAW,S;;MAEf
,OAAO,K;K;IEIFX,iC;MAAA,qC;MAEI,iBAC8B,Q;MAE9B,iBAC8B,sB;MAE9B,yBAEsC,MAAM,G;MAE5C,y

BAEsC,CAAC,GAAD,GAAO,G;MAE7C,WAEwB,EAAE,MAAM,GAAR,C;MAExB,kBACuB,C;MAEvB,iBACs
B,E;K;;IAxB1B,6C;MAAA,4C;QAAA,2B;;MAAA,qC;K;IA2BA,gC;MAAA,oC;MAEI,iBAC6B,O;MAE7B,iBAC
6B,Y;MAE7B,yBAEqC,MAAO,G;MAE5C,yBAEqC,CAAC,GAAD,GAAQ,G;MAE7C,WAEuB,EAAE,MAAO,G
AAT,C;MAEvB,kBACuB,C;MAEvB,iBACsB,E;K;;IAxB1B,4C;MAAA,2C;QAAA,0B;;MAAA,oC;K;IA2BA,8B;
MAAA,kC;MAEI,iBACqB,W;MAErB,iBACqB,U;MAErB,kBACuB,C;MAEvB,iBACsB,E;K;;IAZ1B,0C;MAAA,
yC;QAAA,wB;;MAAA,kC;K;IAeA,+B;MAAA,mC;MAEI,iBACJ,MAAM,KAAoB,U;MAEtB,iBACJ,MAAM,KA
AoB,U;MAEtB,kBACuB,C;MAEvB,iBACsB,E;K;;IAZ1B,2C;MAAA,0C;QAAA,yB;;MAAA,mC;K;IAeA,gC;M
AAA,oC;MAEI,iBACuB,U;MAEvB,iBACuB,K;MAEvB,kBACuB,C;MAEvB,iBACsB,E;K;;IAZ1B,4C;MAAA,2
C;QAAA,0B;;MAAA,oC;K;IAeA,+B;MAAA,mC;MAEI,iBACsB,Q;MAEtB,iBACsB,G;MAEtB,kBACuB,C;MAE
vB,iBACsB,C;K;;IAZ1B,2C;MAAA,0C;QAAA,yB;;MAAA,mC;K;IAeA,+B;MAAA,mC;MAEI,iBACmC,C;MAE
nC,iBACmC,K;MAEnC,0BAC4C,K;MAE5C,0BAC4C,K;MAE5C,yBAC2C,K;MAE3C,yBAC2C,K;MAE3C,qBA
CuC,uB;MAEvC,qBACuC,sB;MAEvC,kBACuB,C;MAEvB,iBACsB,E;K;;IA9B1B,2C;MAAA,0C;QAAA,yB;;M
AAA,mC;K;IAiCA,iC;MAAA,qC;K;;IAAA,6C;MAAA,4C;QAAA,2B;;MAAA,qC;K;IAEA,kC;MAAA,sC;K;;IA
AA,8C;MAAA,6C;QAAA,4B;;MAAA,sC;K;;aCkkuBoB,gB;;cC/ntB0C,mB;;gBAyEvC,
yB;eAAyB,wB;;uBAgBzB,gC;sBAAwB,+B;mCA4JjC,qB;mCA5ImC,qB;;kBAQ1B,2B;iBAA0B,0B;;eC3YgB,
wB;sBCoBA,sB;iBcNBA,0B;;aC5P8B,e;;gCCIdhD,yC;+BCVA,uC;+BCAA,sC;;gCCyJ/B,+B;+BAIW,sC
;gCCqwCc,+B;0BAHvB,kC;uBAr6BO,gC;yBA8WD,iC;0BACA,mC;yBA4JA,iC;gCAmZP,oC;+BAbc,oC;+BAEC
,+B;yBAEQ,kC;;gBCr0C6C,yB;;IC/ErF,kD;MAMuF,wC;K;IANvF,4CAOI,
Y;MAAuC,8B;K;IAP3C,8E;ICGA,kD;MAQuF,wC;K;IARvF,4CASI,Y;MAAuC,8B;K;IAT3C,8E;0FbOA,qB;MA
QI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,
C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,
OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;
K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;0FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,O
AAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;
4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OA
AO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;0
FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,
UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4F
AGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,
UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;0FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAG
X,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UA
AI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,
qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAA
I,CAAJ,C;K;0FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB
;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,C
AAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;M
AQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;IAGX,sC;MAII,OAAO,mBAAQ,OAA
R,KAAoB,C;K;IAG/B,wC;MAII,OAAO,qBAAQ,OAAR,KAAoB,C;K;IAG/B,wC;MAII,OAAO,qBAAQ,OAAR,K
AAoB,C;K;IAG/B,wC;MAII,OAAO,qBAAQ,OAAR,KAAoB,C;K;IAG/B,wC;MAII,OAAO,qBAAQ,OAAR,KAA
oB,C;K;IAG/B,wC;MAOI,OAAO,qBAAQ,OAAR,KAAoB,C;K;IAG/B,wC;MAOI,OAAO,qBAAQ,OAAR,KAAo
B,C;K;IAG/B,wC;MAII,OAAO,qBAAQ,OAAR,KAAoB,C;K;IAG/B,wC;MAII,OAAO,qBAAQ,OAAR,KAAoB,C
;K;0GakE/B,yB;MAAA,8D;MAAA,iD;QAOI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAc,UAAI,KA
AJ,CAAtC,GAAsD,aAAa,KAAb,C;O;KAPjE,C;sGAUA,yB;MAAA,8D;MAAA,iD;QAOI,OAAW,SAAS,CAAT,IAA
c,SAAS,wBAA3B,GAAc,UAAI,KA AJ,CAAtC,GAAsD,aAAa,KAAb,C;O;KAPjE,C;sGAUA,yB;MAAA,8D;MA
AA,iD;QAOI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAc,UAAI,KA AJ,CAAtC,GAAsD,aAAa,KAAb,C;
O;KAPjE,C;sGAUA,yB;MAAA,8D;MAAA,iD;QAOI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAc,UAA
I,KA AJ,CAAtC,GAAsD,aAAa,KAAb,C;O;KAPjE,C;sGAUA,yB;MAAA,8D;MAAA,iD;QAOI,OAAW,SAAS,CA

qB;;UA42CO,Q;UAAA,OAAa,SAAR,sBAAQ,CAAb,W;UAAAd,OAAc,cAAAd,C;YAAc,uB;YACV,cAAc,UAAK,K
AAL,C;YACd,IA92Cc,SA82CV,CAAU,OAAV,CAAJ,C;cAAwB,oBAAO,O;cAAP,sB;;;UAE5B,oBAAO,I;;;QAh3
CP,wB;O;KAPJ,C;wFAUA,yB;Mag3CA,0D;MAAA,+C;MAAA,oC;MAh3CA,uC;QAOW,qB;;UA+2CO,Q;UAA
A,OAAa,SAAR,sBAAQ,CAAb,W;UAAAd,OAAc,cAAAd,C;YAAc,uB;YACV,cAAc,UAAK,KAAL,C;YACd,IAj3Cc,
SAi3CV,CAAU,oBAAV,CAAJ,C;cAAwB,oBAAO,O;cAAP,sB;;;UAE5B,oBAAO,I;;;QAn3CP,wB;O;KAPJ,C;IA
UA,0B;MAKI,IA4uNO,qBAAQ,CA5uNf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO,UAAK,CAAL,C;
K;IAGX,4B;MAKI,IA0uNO,qBAAQ,CA1uNf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO,UAAK,CAA
L,C;K;IAGX,4B;MAKI,IAwuNO,qBAAQ,CXuuNf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO,UAAK,
CAAL,C;K;IAGX,4B;MAKI,IASuNO,qBAAQ,CAtuNf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO,UA
AK,CAAL,C;K;IAGX,4B;MAKI,IAouNO,qBAAQ,CApuNf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO
,UAAK,CAAL,C;K;IAGX,4B;MAKI,IAkuNO,qBAAQ,CAluNf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,O
AAO,UAAK,CAAL,C;K;IAGX,4B;MAKI,IAguNO,qBAAQ,CAhuNf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MAC
V,OAAO,UAAK,CAAL,C;K;IAGX,4B;MAKI,IA8tNO,qBAAQ,CA9tNf,C;QACI,MAAM,2BAAuB,iBAAvB,C;M
ACV,OAAO,UAAK,CAAL,C;K;IAGX,4B;MAKI,IA4tNO,qBAAQ,CA5tNf,C;QACI,MAAM,2BAAuB,iBAAvB,
C;MACV,OAAO,UAAK,CAAL,C;K;kFAGX,yB;MAAA,iE;MAAA,uC;QAKoB,Q;QAAhB,wBAAGB,SAAhB,gB
;UAAgB,cAAA,SAAhB,M;UAAsB,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QACrD,MAAM,gCAAuB,
mDAAvB,C;O;KANV,C;kFASA,yB;MAAA,iE;MAAA,uC;QAKoB,Q;QAAhB,wBAAGB,SAAhB,gB;UAAgB,cA
AA,SAAhB,M;UAAsB,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QACrD,MAAM,gCAAuB,mDAAvB,C;
O;KANV,C;mFASA,yB;MAAA,iE;MAAA,uC;QAKoB,Q;QAAhB,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,
M;UAAsB,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QACrD,MAAM,gCAAuB,mDAAvB,C;O;KANV,C;
mFASA,yB;MAAA,iE;MAAA,uC;QAKoB,Q;QAAhB,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UAAsB,I
AAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QACrD,MAAM,gCAAuB,mDAAvB,C;O;KANV,C;mFASA,yB;
MAAA,iE;MAAA,uC;QAKoB,Q;QAAhB,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UAAsB,IAAI,UAAU,
OAAV,CAAJ,C;YAAwB,OAAO,O;;QACrD,MAAM,gCAAuB,mDAAvB,C;O;KANV,C;mFASA,yB;MAAA,iE;M
AAA,uC;QAKoB,Q;QAAhB,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UAAsB,IAAI,UAAU,OAAV,CAAJ
,C;YAAwB,OAAO,O;;QACrD,MAAM,gCAAuB,mDAAvB,C;O;KANV,C;mFASA,yB;MAAA,iE;MAAA,uC;QA
KoB,Q;QAAhB,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UAAsB,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,
OAAO,O;;QACrD,MAAM,gCAAuB,mDAAvB,C;O;KANV,C;mFASA,yB;MAAA,iE;MAAA,uC;QAKoB,Q;QA
AhB,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UAAsB,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;Q
ACrD,MAAM,gCAAuB,mDAAvB,C;O;KANV,C;mFASA,yB;MAAA,oC;MAAA,gC;MAAA,iE;MAAA,uC;QAK
oB,Q;QAAhB,wBAAGB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UAAsB,IAAI,UAAU,oBAAV,CAAJ,C;
YAAwB,OAAO,O;;QACrD,MAAM,gCAAuB,mDAAvB,C;O;KANV,C;kGASA,yB;MAAA,iE;MAAA,uC;QASW
,Q;QAAA,+B;;UAYS,U;UAAhB,uD;YAAgB,cAAhB,iB;YACI,aAbwB,SAaX,CAAU,OAAV,C;YACb,IAAI,CAAJ
,C;cACI,8BAAO,M;cAAP,gC;;;UAGR,8BAAO,I;;;QAIbA,kC;QAAA,iB;UAAmC,MAAM,gCAAuB,8DAAvB,C;;
QAAhD,OAAO,I;O;KATX,C;8GAYA,gC;MASoB,Q;MAAhB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;Q
ACI,aAAa,UAAU,OAAV,C;QACb,IAAI,CAAJ,C;UACI,OAAO,M;;;MAGf,OAAO,I;K;IAGX,gC;MAII,OAoiNO,
qBAAQ,CapiNR,GAAe,IAAf,GAAYB,UAAK,CAAL,C;K;IAGpC,kC;MAII,OAqiNO,qBAAQ,CAriNR,GAAe,IA
Af,GAAYB,UAAK,CAAL,C;K;IAGpC,kC;MAII,OAsiNO,qBAAQ,CAtiNR,GAAe,IAAf,GAAYB,UAAK,CAAL,C
;K;IAGpC,kC;MAII,OAuiNO,qBAAQ,CAviNR,GAAe,IAAf,GAAYB,UAAK,CAAL,C;K;IAGpC,kC;MAII,OAwi
NO,qBAAQ,CAXiNR,GAAe,IAAf,GAAYB,UAAK,CAAL,C;K;IAGpC,kC;MAII,OAYiNO,qBAAQ,CAziNR,GAA
e,IAAf,GAAYB,UAAK,CAAL,C;K;IAGpC,kC;MAII,OA0iNO,qBAAQ,CA1iNR,GAAe,IAAf,GAAYB,UAAK,CA
AL,C;K;IAGpC,kC;MAII,OA2iNO,qBAAQ,CA3iNR,GAAe,IAAf,GAAYB,UAAK,CAAL,C;K;IAGpC,kC;MAII,O
A4iNO,qBAAQ,CA5iNR,GAAe,IAAf,GAAYB,UAAK,CAAL,C;K;8FAGpC,gC;MAIoB,Q;MAAhB,wBAAGB,SA
AhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,O;;MACrD,OAAO,I;K
;8FAGX,gC;MAIoB,Q;MAAhB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,C
AAJ,C;UAAwB,OAAO,O;;MACrD,OAAO,I;K;+FAGX,gC;MAIoB,Q;MAAhB,wBAAGB,SAAhB,gB;QAAGB,cA
AA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,O;;MACrD,OAAO,I;K;+FAGX,gC;MAIoB
,Q;MAAhB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OA

AO,O;;MACrD,OAAO,I;K;+FAGX,gC;MAIoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAA
sB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,O;;MACrD,OAAO,I;K;+FAGX,gC;MAIoB,Q;MAAhB,wBAAg
B,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,O;;MACrD,OAA
O,I;K;+FAGX,gC;MAIoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OA
AV,CAAJ,C;UAAwB,OAAO,O;;MACrD,OAAO,I;K;+FAGX,gC;MAIoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAA
gB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,O;;MACrD,OAAO,I;K;+FAGX,yB;
MAAA,oC;MAAA,gC;MAAA,uC;QAIoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;U
AAsB,IAAI,UAAU,oBAAV,CAAJ,C;YAAwB,OAAO,O;;QACrD,OAAO,I;O;KALX,C;wFAQA,yB;MAAA,8D;M
AAA,iD;QAKI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAsC,UAAI,KAAJ,CAAtC,GAAsD,aAAa,KAAb,
C;O;KALjE,C;0FAQA,yB;MAAA,8D;MAAA,iD;QAKI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAsC,UA
AI,KAAJ,CAAtC,GAAsD,aAAa,KAAb,C;O;KALjE,C;0FAQA,yB;MAAA,8D;MAAA,iD;QAKI,OAAW,SAAS,C
AAT,IAAc,SAAS,wBAA3B,GAAsC,UAAI,KAAJ,CAAtC,GAAsD,aAAa,KAAb,C;O;KALjE,C;0FAQA,yB;MAA
A,8D;MAAA,iD;QAKI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAsC,UAAI,KAAJ,CAAtC,GAAsD,aAAa
,KAAb,C;O;KALjE,C;0FAQA,yB;MAAA,8D;MAAA,iD;QAKI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GA
AsC,UAAI,KAAJ,CAAtC,GAAsD,aAAa,KAAb,C;O;KALjE,C;0FAQA,yB;MAAA,8D;MAAA,iD;QAKI,OAAW,
SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAsC,UAAI,KAAJ,CAAtC,GAAsD,aAAa,KAAb,C;O;KALjE,C;0FAQA,y
B;MAAA,8D;MAAA,iD;QAKI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAsC,UAAI,KAAJ,CAAtC,GAAs
D,aAAa,KAAb,C;O;KALjE,C;0FAQA,yB;MAAA,8D;MAAA,iD;QAKI,OAAW,SAAS,CAAT,IAAc,SAAS,wBA
A3B,GAAsC,UAAI,KAAJ,CAAtC,GAAsD,aAAa,KAAb,C;O;KALjE,C;0FAQA,yB;MAAA,8D;MAAA,gC;MAA
A,iD;QAKI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAsC,UAAI,KAAJ,CAAtC,GAAsD,uBAAa,KAAb,E;
O;KALjE,C;IAQA,qC;MAMI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAsC,UAAI,KAAJ,CAAtC,GAAsD
,I;K;IAGjE,uC;MAMI,OAAW,SAAS,CAAT,IAAc,SAAS,0BAA3B,GAAsC,UAAI,KAAJ,CAAtC,GAAsD,I;K;IA
GjE,uC;MAMI,OAAW,SAAS,CAAT,IAAc,SAAS,0BAA3B,GAAsC,UAAI,KAAJ,CAAtC,GAAsD,I;K;IAGjE,uC;
MAMI,OAAW,SAAS,CAAT,IAAc,SAAS,0BAA3B,GAAsC,UAAI,KAAJ,CAAtC,GAAsD,I;K;IAGjE,uC;MAMI,
OAAW,SAAS,CAAT,IAAc,SAAS,0BAA3B,GAAsC,UAAI,KAAJ,CAAtC,GAAsD,I;K;IAGjE,uC;MAMI,OAAW,
SAAS,CAAT,IAAc,SAAS,0BAA3B,GAAsC,UAAI,KAAJ,CAAtC,GAAsD,I;K;IAGjE,uC;MAMI,OAAW,SAAS,
CAAT,IAAc,SAAS,0BAA3B,GAAsC,UAAI,KAAJ,CAAtC,GAAsD,I;K;IAGjE,uC;MAMI,OAAW,SAAS,CAAT,I
AAc,SAAS,0BAA3B,GAAsC,UAAI,KAAJ,CAAtC,GAAsD,I;K;IAGjE,uC;MAMI,OAAW,SAAS,CAAT,IAAc,SA
AS,0BAA3B,GAAsC,UAAI,KAAJ,CAAtC,GAAsD,I;K;IAGjE,qC;MAIL,IAAI,eAAJ,C;QACI,wD;UACI,IAAI,UA
AK,KAAL,SAAJ,C;YACI,OAAO,K;;;QAI,f,8D;UACI,IAAI,gBAAW,UAAK,OAAL,CAAX,CAAJ,C;YACI,OAA
O,O;;;MAInB,OAAO,E;K;IAGX,uC;MAIL,wD;QACI,IAAI,YAAW,UAAK,KAAL,CAAF,C;UACI,OAAO,K;;;M
AGf,OAAO,E;K;IAGX,uC;MAIL,wD;QACI,IAAI,YAAW,UAAK,KAAL,CAAF,C;UACI,OAAO,K;;;MAGf,OAA
O,E;K;IAGX,uC;MAIL,wD;QACI,IAAI,YAAW,UAAK,KAAL,CAAF,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;IAG
X,uC;MAIL,wD;QACI,IAAI,gBAAW,UAAK,KAAL,CAAX,CAAJ,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;IAGX,
uC;MAMI,wD;QACI,IAAI,YAAW,UAAK,KAAL,CAAF,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;IAGX,uC;MAM
I,wD;QACI,IAAI,YAAW,UAAK,KAAL,CAAF,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;IAGX,uC;MAIL,wD;QACI
,IAAI,YAAW,UAAK,KAAL,CAAF,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;IAGX,uC;MAIL,wD;QACI,IAAI,YAA
W,UAAK,KAAL,CAAF,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;8FAGX,gC;MAIL,wD;QACI,IAAI,UAAU,UAAK
,KAAL,CAAV,CAAJ,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;gGAGX,gC;MAIL,wD;QACI,IAAI,UAAU,UAAK,K
AAL,CAAV,CAAJ,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;gGAGX,gC;MAIL,wD;QACI,IAAI,UAAU,UAAK,KA
AL,CAAV,CAAJ,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;gGAGX,gC;MAIL,wD;QACI,IAAI,UAAU,UAAK,KAA
L,CAAV,CAAJ,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;gGAGX,gC;MAIL,wD;QACI,IAAI,UAAU,UAAK,KAAL,
CAAV,CAAJ,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;gGAGX,gC;MAIL,wD;QACI,IAAI,UAAU,UAAK,KAAL,C
AAV,CAAJ,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;gGAGX,gC;MAIL,wD;QACI,IAAI,UAAU,UAAK,KAAL,CA
AV,CAAJ,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;gGAGX,yB;MAAA,oC;MAAA,uC;QAI,wD;UACI,IAAI,UAAU
,sBAAK,KAAL,EA AV,CAAJ,C;YACI,OAAO,K;;;QAGf,OAAO,E;O;KATX,C;4FAYA,yB;MAAA,0D;MAAA,+
C;MAAA,uC;QAIkB,Q;QAAA,OAAQ,SAAR,sBAAQ,CAAR,W;QAAd,OAAC,cAAAd,C;UAAc,uB;UACV,IAAI,U

AAU,UAAK,KAAL,CAAV,CAAJ,C;YACI,OAAO,K;;;QAGf,OAAO,E;O;KATX,C;8FAYA,yB;MAAA,0D;MAA
A,+C;MAAA,uC;QAikB,Q;QAAA,OAAQ,SAAR,sBAAQ,CAAR,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,IA
AI,UAAU,UAAK,KAAL,CAAV,CAAJ,C;YACI,OAAO,K;;;QAGf,OAAO,E;O;KATX,C;8FAYA,yB;MAAA,0D;
MAAA,+C;MAAA,uC;QAikB,Q;QAAA,OAAQ,SAAR,sBAAQ,CAAR,W;QAAd,OAAc,cAAd,C;UAAc,uB;UAC
V,IAAI,UAAU,UAAK,KAAL,CAAV,CAAJ,C;YACI,OAAO,K;;;QAGf,OAAO,E;O;KATX,C;8FAYA,yB;MAAA,
0D;MAAA,+C;MAAA,uC;QAikB,Q;QAAA,OAAQ,SAAR,sBAAQ,CAAR,W;QAAd,OAAc,cAAd,C;UAAc,uB;U
ACV,IAAI,UAAU,UAAK,KAAL,CAAV,CAAJ,C;YACI,OAAO,K;;;QAGf,OAAO,E;O;KATX,C;8FAYA,yB;MA
AA,0D;MAAA,+C;MAAA,uC;QAikB,Q;QAAA,OAAQ,SAAR,sBAAQ,CAAR,W;QAAd,OAAc,cAAd,C;UAAc,u
B;UACV,IAAI,UAAU,UAAK,KAAL,CAAV,CAAJ,C;YACI,OAAO,K;;;QAGf,OAAO,E;O;KATX,C;8FAYA,yB;
MAAA,0D;MAAA,+C;MAAA,uC;QAikB,Q;QAAA,OAAQ,SAAR,sBAAQ,CAAR,W;QAAd,OAAc,cAAd,C;UA
Ac,uB;UACV,IAAI,UAAU,UAAK,KAAL,CAAV,CAAJ,C;YACI,OAAO,K;;;QAGf,OAAO,E;O;KATX,C;8FAYA
,yB;MAAA,0D;MAAA,+C;MAAA,uC;QAikB,Q;QAAA,OAAQ,SAAR,sBAAQ,CAAR,W;QAAd,OAAc,cAAd,C;
UAAc,uB;UACV,IAAI,UAAU,UAAK,KAAL,CAAV,CAAJ,C;YACI,OAAO,K;;;QAGf,OAAO,E;O;KATX,C;8FA
YA,yB;MAAA,0D;MAAA,+C;MAAA,uC;QAikB,Q;QAAA,OAAQ,SAAR,sBAAQ,CAAR,W;QAAd,OAAc,cAAd
,C;UAAc,uB;UACV,IAAI,UAAU,UAAK,KAAL,CAAV,CAAJ,C;YACI,OAAO,K;;;QAGf,OAAO,E;O;KATX,C;8
FAYA,yB;MAAA,0D;MAAA,+C;MAAA,oC;MAAA,uC;QAikB,Q;QAAA,OAAQ,SAAR,sBAAQ,CAAR,W;QA
Ad,OAAc,cAAd,C;UAAc,uB;UACV,IAAI,UAAU,sBAAK,KAAL,EAAV,CAAJ,C;YACI,OAAO,K;;;QAGf,OAA
O,E;O;KATX,C;IAYA,yB;MAQI,IAg7LO,qBAAQ,CAh7Lf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO
,UAAK,wBAAL,C;K;IAGX,2B;MAQI,IA26LO,qBAAQ,CA36Lf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,
OAAO,UAAK,0BAAL,C;K;IAGX,2B;MAQI,IAs6LO,qBAAQ,CAt6Lf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MA
CV,OAAO,UAAK,0BAAL,C;K;IAGX,2B;MAQI,IAi6LO,qBAAQ,CAj6Lf,C;QACI,MAAM,2BAAuB,iBAAvB,C;
MACV,OAAO,UAAK,0BAAL,C;K;IAGX,2B;MAQI,IA45LO,qBAAQ,CA55Lf,C;QACI,MAAM,2BAAuB,iBAA
vB,C;MACV,OAAO,UAAK,0BAAL,C;K;IAGX,2B;MAQI,IAu5LO,qBAAQ,CAv5Lf,C;QACI,MAAM,2BAAuB,i
BAAvB,C;MACV,OAAO,UAAK,0BAAL,C;K;IAGX,2B;MAQI,IAk5LO,qBAAQ,CAI5Lf,C;QACI,MAAM,2BA
AuB,iBAAvB,C;MACV,OAAO,UAAK,0BAAL,C;K;IAGX,2B;MAQI,IA64LO,qBAAQ,CA74Lf,C;QACI,MAAM
,2BAAuB,iBAAvB,C;MACV,OAAO,UAAK,0BAAL,C;K;IAGX,2B;MAQI,IAw4LO,qBAAQ,CAx4Lf,C;QACI,M
AAM,2BAAuB,iBAAvB,C;MACV,OAAO,UAAK,0BAAL,C;K;gFAGX,yB;MAAA,0D;MAAA,+C;MAAA,iE;M
AAA,uC;QAQkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAAc,cAAd,C;UAAc,uB;UAC
V,cAAc,UAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,MAAM,gCAAuB,mD
AAvB,C;O;KAZV,C;gFAeA,yB;MAAA,0D;MAAA,+C;MAAA,iE;MAAA,uC;QAQkB,Q;QAAA,OAAa,SAAR,Y
AAL,SAAK,CAAQ,CAAb,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,cAAc,UAAK,KAAL,C;UACd,IAAI,UAA
U,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,MAAM,gCAAuB,mDAAvB,C;O;KAZV,C;iFAeA,yB;MAAA,0D;
MAAA,+C;MAAA,iE;MAAA,uC;QAQkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAAc,
cAAd,C;UAAc,uB;UACV,cAAc,UAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAE
nC,MAAM,gCAAuB,mDAAvB,C;O;KAZV,C;iFAeA,yB;MAAA,0D;MAAA,+C;MAAA,iE;MAAA,uC;QAQkB,Q;
QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,cAAc,UAAK,KAA
L,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,MAAM,gCAAuB,mDAAvB,C;O;KAZV,C;i
FAeA,yB;MAAA,0D;MAAA,+C;MAAA,iE;MAAA,uC;QAQkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,C
AAb,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,cAAc,UAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YA
AwB,OAAO,O;;QAEnC,MAAM,gCAAuB,mDAAvB,C;O;KAZV,C;iFAeA,yB;MAAA,0D;MAAA,+C;MAAA,iE;
MAAA,uC;QAQkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAAc,cAAd,C;UAAc,uB;UA
CV,cAAc,UAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,MAAM,gCAAuB,m
DAAvB,C;O;KAZV,C;iFAeA,yB;MAAA,0D;MAAA,+C;MAAA,iE;MAAA,uC;QAQkB,Q;QAAA,OAAa,SAAR,
YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,cAAc,UAAK,KAAL,C;UACd,IAAI,UA
AU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,MAAM,gCAAuB,mDAAvB,C;O;KAZV,C;iFAeA,yB;MAAA,0
D;MAAA,+C;MAAA,iE;MAAA,uC;QAQkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAA
c,cAAd,C;UAAc,uB;UACV,cAAc,UAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QA
EnC,MAAM,gCAAuB,mDAAvB,C;O;KAZV,C;iFAeA,yB;MAAA,0D;MAAA,+C;MAAA,oC;MAAA,iE;MAAA,

uC;QAQkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,cAA
c,UAAK,KAAL,C;UACd,IAAI,UAAU,oBAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,MAAM,gCAAuB,mDAAvB
,C;O;KAZV,C;IAeA,yC;MAKsB,UAMA,M;MAPIB,IAAI,eAAJ,C;QACkB,OAAQ,WAAR,sBAAQ,CAAR,W;QA
Ad,OAAc,cAAd,C;UAAc,uB;UACV,IAAI,UAAK,KAAL,SAAJ,C;YACI,OAAO,K;;;QAID,SAAQ,WAAR,sBAA
Q,CAAR,W;QAAd,OAAc,gBAAd,C;UAAc,2B;UACV,IAAI,gBAAW,UAAK,OAAL,CAAX,CAAJ,C;YACI,OAA
O,O;;;MAInB,OAAO,E;K;IAGX,2C;MAIkB,Q;MAAA,OAAQ,WAAR,wBAAQ,CAAR,W;MAAd,OAAc,cAAd,
C;QAAc,uB;QACV,IAAI,YAAW,UAAK,KAAL,CAAf,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;IAGX,2C;MAIkB,
Q;MAAA,OAAQ,WAAR,wBAAQ,CAAR,W;MAAd,OAAc,cAAd,C;QAAc,uB;QACV,IAAI,YAAW,UAAK,KAA
L,CAAf,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;IAGX,2C;MAIkB,Q;MAAA,OAAQ,WAAR,wBAAQ,CAAR,W;
MAAd,OAAc,cAAd,C;QAAc,uB;QACV,IAAI,YAAW,UAAK,KAAL,CAAf,C;UACI,OAAO,K;;;MAGf,OAAO,E;
K;IAGX,2C;MAIkB,Q;MAAA,OAAQ,WAAR,wBAAQ,CAAR,W;MAAd,OAAc,cAAd,C;QAAc,uB;QACV,IAAI,
gBAAW,UAAK,KAAL,CAAX,CAAJ,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;IAGX,2C;MAMkB,Q;MAAA,OAA
Q,WAAR,wBAAQ,CAAR,W;MAAd,OAAc,cAAd,C;QAAc,uB;QACV,IAAI,YAAW,UAAK,KAAL,CAAf,C;UA
CI,OAAO,K;;;MAGf,OAAO,E;K;IAGX,2C;MAMkB,Q;MAAA,OAAQ,WAAR,wBAAQ,CAAR,W;MAAd,OAAc,
cAAd,C;QAAc,uB;QACV,IAAI,YAAW,UAAK,KAAL,CAAf,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;IAGX,2C;
MAIkB,Q;MAAA,OAAQ,WAAR,wBAAQ,CAAR,W;MAAd,OAAc,cAAd,C;QAAc,uB;QACV,IAAI,YAAW,UA
AK,KAAL,CAAf,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;IAGX,2C;MAIkB,Q;MAAA,OAAQ,WAAR,wBAAQ,C
AAR,W;MAAd,OAAc,cAAd,C;QAAc,uB;QACV,IAAI,YAAW,UAAK,KAAL,CAAf,C;UACI,OAAO,K;;;MAGf,
OAAO,E;K;IAGX,+B;MAMI,OA8jLO,qBAAQ,CA9jLR,GAAe,IAAf,GAAYB,UAAK,mBAAO,CAAP,IAAL,C;K;
IAGpC,iC;MAMI,OA6jLO,qBAAQ,CA7jLR,GAAe,IAAf,GAAYB,UAAK,mBAAO,CAAP,IAAL,C;K;IAGpC,iC;
MAMI,OA4jLO,qBAAQ,CA5jLR,GAAe,IAAf,GAAYB,UAAK,mBAAO,CAAP,IAAL,C;K;IAGpC,iC;MAMI,OA
2jLO,qBAAQ,CA3jLR,GAAe,IAAf,GAAYB,UAAK,mBAAO,CAAP,IAAL,C;K;IAGpC,iC;MAMI,OA0jLO,qBA
AQ,CA1jLR,GAAe,IAAf,GAAYB,UAAK,mBAAO,CAAP,IAAL,C;K;IAGpC,iC;MAMI,OAyjLO,qBAAQ,CAzjL
R,GAAe,IAAf,GAAYB,UAAK,mBAAO,CAAP,IAAL,C;K;IAGpC,iC;MAMI,OAwjLO,qBAAQ,CAxjLR,GAAe,I
AAf,GAAYB,UAAK,mBAAO,CAAP,IAAL,C;K;IAGpC,iC;MAMI,OAujLO,qBAAQ,CAvjLR,GAAe,IAAf,GAAY
B,UAAK,mBAAO,CAAP,IAAL,C;K;IAGpC,iC;MAMI,OAsjLO,qBAAQ,CAtjLR,GAAe,IAAf,GAAYB,UAAK,m
BAAO,CAAP,IAAL,C;K;4FAGpC,yB;MAAA,0D;MAAA,+C;MAAA,uC;QAMkB,Q;QAAA,OAAa,SAAR,YAAL
,SAAK,CAAQ,CAAb,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,cAAc,UAAK,KAAL,C;UACd,IAAI,UAAU,OA
AV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,OAAO,I;O;KAVX,C;6FAaA,yB;MAAA,0D;MAAA,+C;MAAA,uC;QA
MkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,cAAc,UA
AK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,OAAO,I;O;KAVX,C;6FAaA,yB;
MAAA,0D;MAAA,+C;MAAA,uC;QAMkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAAc,
cAAd,C;UAAc,uB;UACV,cAAc,UAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAE
nC,OAAO,I;O;KAVX,C;6FAaA,yB;MAAA,0D;MAAA,+C;MAAA,uC;QAMkB,Q;QAAA,OAAa,SAAR,YAAL,S
AAK,CAAQ,CAAb,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,cAAc,UAAK,KAAL,C;UACd,IAAI,UAAU,OAA
V,CAAJ,C;YAAwB,OAAO,O;;QAEnC,OAAO,I;O;KAVX,C;6FAaA,yB;MAAA,0D;MAAA,+C;MAAA,uC;QAM
kB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,cAAc,UAAK
,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,OAAO,I;O;KAVX,C;6FAaA,yB;MA
AA,0D;MAAA,+C;MAAA,uC;QAMkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAAc,cA
Ad,C;UAAc,uB;UACV,cAAc,UAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,
OAAO,I;O;KAVX,C;6FAaA,yB;MAAA,0D;MAAA,+C;MAAA,uC;QAMkB,Q;QAAA,OAAa,SAAR,YAAL,SAA
K,CAAQ,CAAb,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,cAAc,UAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,C
AAJ,C;YAAwB,OAAO,O;;QAEnC,OAAO,I;O;KAVX,C;6FAaA,yB;MAAA,0D;MAAA,+C;MAAA,uC;QAMkB,
Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,cAAc,UAAK,K
AAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,OAAO,I;O;KAVX,C;6FAaA,yB;MAAA
,0D;MAAA,+C;MAAA,oC;MAAA,uC;QAMkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAd,O
AAc,cAAd,C;UAAc,uB;UACV,cAAc,UAAK,KAAL,C;UACd,IAAI,UAAU,oBAAV,CAAJ,C;YAAwB,OAAO,O;;
QAEnC,OAAO,I;O;KAVX,C;kFAaA,yB;MAAA,mC;MAAA,gD;MAAA,4B;QAQI,OAAO,kBAAO,cAAP,C;O;K

C;UAAK,iBAAK,CAAL,C;UAAL,K;;UACQ,MAAM,gCAAyB,kCAAzB,C;;MAHIB,W;K;IAOJ,6B;MAliB,IAAN ,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,MAAM,2BAAuB,iBAAvB,C;aACX,C;UAAK,iBAAK,CAAL,C;UA AL,K;;UACQ,MAAM,gCAAyB,kCAAzB,C;;MAHIB,W;K;oFAOJ,yB;MAAA,kF;MAAA,iE;MAAA,gB;MAAA,8 B;MAAA,uC;QAMoB,UAST,M;QAXP,aAAiB,I;QACjB,YAAY,K;QACZ,wBAAgB,SAAhB,gB;UAAgB,cAAA,S AAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,IAAI,KAAJ,C;cAAW,MAAM,8BAAyB,gDAAzB,C;YACjB, SAAS,O;YACT,QAAQ,I;;;QAGhB,IAAI,CAAC,KAAL,C;UAAy,MAAM,gCAAuB,mDAAvB,C;QAEIB,OAAO, 6E;O;KafX,C;oFakBA,yB;MAAA,kF;MAAA,iE;MAAA,8B;MAAA,uC;QAMoB,UAST,M;QAXP,aAAoB,I;QA CpB,YAAY,K;QACZ,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YAC I,IAAI,KAAJ,C;cAAW,MAAM,8BAAyB,gDAAzB,C;YACjB,SAAS,O;YACT,QAAQ,I;;;QAGhB,IAAI,CAAC,K AAL,C;UAAy,MAAM,gCAAuB,mDAAvB,C;QAEIB,OAAO,2D;O;KafX,C;qFakBA,yB;MAAA,kF;MAAA,iE; MAAA,8B;MAAA,uC;QAMoB,UAST,M;QAXP,aAAqB,I;QACrB,YAAY,K;QACZ,wBAAgB,SAAhB,gB;UAAg B,cAAA,SAAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,IAAI,KAAJ,C;cAAW,MAAM,8BAAyB,gDAAzB, C;YACjB,SAAS,O;YACT,QAAQ,I;;;QAGhB,IAAI,CAAC,KAAL,C;UAAy,MAAM,gCAAuB,mDAAvB,C;QAEI B,OAAO,2D;O;KafX,C;qFakBA,yB;MAAA,kF;MAAA,iE;MAAA,8B;MAAA,uC;QAMoB,UAST,M;QAXP,aA AmB,I;QACnB,YAAY,K;QACZ,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,IAAI,UAAU,OAAV,CA AJ,C;YACI,IAAI,KAAJ,C;cAAW,MAAM,8BAAyB,gDAAzB,C;YACjB,SAAS,O;YACT,QAAQ,I;;;QAGhB,IAAI ,CAAC,KAAL,C;UAAy,MAAM,gCAAuB,mDAAvB,C;QAEIB,OAAO,2D;O;KafX,C;qFakBA,yB;MAAA,kF;M AAA,iE;MAAA,8B;MAAA,uC;QAMoB,UAST,M;QAXP,aAAoB,I;QACpB,YAAY,K;QACZ,wBAAgB,SAAhB,g B;UAAgB,cAAA,SAAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,IAAI,KAAJ,C;cAAW,MAAM,8BAAyB, gDAAzB,C;YACjB,SAAS,O;YACT,QAAQ,I;;;QAGhB,IAAI,CAAC,KAAL,C;UAAy,MAAM,gCAAuB,mDAAv B,C;QAEIB,OAAO,iE;O;KafX,C;qFakBA,yB;MAAA,kF;MAAA,iE;MAAA,8B;MAAA,uC;QAMoB,UAST,M;Q AXp,aAAqB,I;QACrB,YAAY,K;QACZ,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,IAAI,UAAU,OA AV,CAAJ,C;YACI,IAAI,KAAJ,C;cAAW,MAAM,8BAAyB,gDAAzB,C;YACjB,SAAS,O;YACT,QAAQ,I;;;QAG hB,IAAI,CAAC,KAAL,C;UAAy,MAAM,gCAAuB,mDAAvB,C;QAEIB,OAAO,2D;O;KafX,C;qFakBA,yB;MA AA,kF;MAAA,iE;MAAA,8B;MAAA,uC;QAMoB,UAST,M;QAXP,aAAoB,I;QACtB,YAAY,K;QACZ,wBAAgB,S AAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,IAAI,KAAJ,C;cAAW,MAAM,8 BAAyB,gDAAzB,C;YACjB,SAAS,O;YACT,QAAQ,I;;;QAGhB,IAAI,CAAC,KAAL,C;UAAy,MAAM,gCAAuB, mDAAvB,C;QAEIB,OAAO,2D;O;KafX,C;qFakBA,yB;MAAA,kF;MAAA,iE;MAAA,8B;MAAA,uC;QAMoB,U AST,M;QAXP,aAAuB,I;QACvB,YAAY,K;QACZ,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,IAAI,U AAU,OAAV,CAAJ,C;YACI,IAAI,KAAJ,C;cAAW,MAAM,8BAAyB,gDAAzB,C;YACjB,SAAS,O;YACT,QAAQ ,I;;;QAGhB,IAAI,CAAC,KAAL,C;UAAy,MAAM,gCAAuB,mDAAvB,C;QAEIB,OAAO,4D;O;KafX,C;qFakBA ,yB;MAAA,oC;MAAA,kF;MAAA,gC;MAAA,iE;MAAA,8B;MAAA,uC;QAMoB,UAST,M;QAXP,aAAoB,I;QAC pB,YAAY,K;QACZ,wBAAgB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UACI,IAAI,UAAU,oBAAV,CAA J,C;YACI,IAAI,KAAJ,C;cAAW,MAAM,8BAAyB,gDAAzB,C;YACjB,SAAS,O;YACT,QAAQ,I;;;QAGhB,IAAI, CAAC,KAAL,C;UAAy,MAAM,gCAAuB,mDAAvB,C;QAEIB,OAAO,4E;O;KafX,C;IAkBA,iC;MAII,OAAW,q BAAQ,CAAZ,GAAe,UAAK,CAAL,CAAf,GAA4B,I;K;IAGvC,mC;MAII,OAAW,qBAAQ,CAAZ,GAAe,UAAK,CAAL,CAAf,GAA4B,I;K;IA GvC,mC;MAII,OAAW,qBAAQ,CAAZ,GAAe,UAAK,CAAL,CAAf,GAA4B,I;K;IAGvC,mC;MAII,OAAW,qBAAQ,CAAZ,GAAe,UAAK,CAA L,CAAf,GAA4B,I;K;IAGvC,mC;MAII,OAAW,qBAAQ,CAAZ,GAAe,UAAK,CAAL,CAAf,GAA4B,I;K;IAGvC, mC;MAII,OAAW,qBAAQ,CAAZ,GAAe,UAAK,CAAL,CAAf,GAA4B,I;K;IAGvC,mC;MAII,OAAW,qBAAQ,CA AZ,GAAe,UAAK,CAAL,CAAf,GAA4B,I;K;gGAGvC,gC;MAMoB,Q;MAFhB,aAAiB,I;MACjB,YAAY,K;MACZ ,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,IAAI,UAAU,OAAV,CAAJ,C;UACI,IAAI,KAAJ,C;YAA W,OAAO,I;UACIB,SAAS,O;UACT,QAAQ,I;;;MAGhB,IAAI,CAAC,KAAL,C;QAAY,OAAO,I;MACnB,OAAO, M;K;gGAGX,gC;MAMoB,Q;MAFhB,aAAoB,I;MACpB,YAAY,K;MACZ,wBAAgB,SAAhB,gB;QAAGB,cAAA,S AAhB,M;QACI,IAAI,UAAU,OAAV,CAAJ,C;UACI,IAAI,KAAJ,C;YAAW,OAAO,I;UACIB,SAAS,O;UACT,QA AQ,I;;;MAGhB,IAAI,CAAC,KAAL,C;QAAY,OAAO,I;MACnB,OAAO,M;K;iGAGX,gC;MAMoB,Q;MAFhB,aA AqB,I;MACrB,YAAY,K;MACZ,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,IAAI,UAAU,OAAV,CA

AJ,C;UACI,IAAI,KAAJ,C;YAAW,OAAO,I;UACIB,SAAS,O;UACT,QAAQ,I;;;MAGhB,IAAI,CAAC,KAAL,C;Q
AAy,OAAO,I;MACnB,OAAO,M;K;iGAGX,gC;MAMoB,Q;MAFhB,aAAmB,I;MACnB,YAAy,K;MACZ,wBAA
gB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,IAAI,UAAU,OAAV,CAAJ,C;UACI,IAAI,KAAJ,C;YAAW,OAA
O,I;UACIB,SAAS,O;UACT,QAAQ,I;;;MAGhB,IAAI,CAAC,KAAL,C;QAAy,OAAO,I;MACnB,OAAO,M;K;iGA
GX,gC;MAMoB,Q;MAFhB,aAAoB,I;MACpB,YAAy,K;MACZ,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;
QACI,IAAI,UAAU,OAAV,CAAJ,C;UACI,IAAI,KAAJ,C;YAAW,OAAO,I;UACIB,SAAS,O;UACT,QAAQ,I;;;M
AGhB,IAAI,CAAC,KAAL,C;QAAy,OAAO,I;MACnB,OAAO,M;K;iGAGX,gC;MAMoB,Q;MAFhB,aAAqB,I;M
ACrB,YAAy,K;MACZ,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,IAAI,UAAU,OAAV,CAAJ,C;UA
CI,IAAI,KAAJ,C;YAAW,OAAO,I;UACIB,SAAS,O;UACT,QAAQ,I;;;MAGhB,IAAI,CAAC,KAAL,C;QAAy,OA
AO,I;MACnB,OAAO,M;K;iGAGX,gC;MAMoB,Q;MAFhB,aAAsB,I;MACtB,YAAy,K;MACZ,wBAAgB,SAAhB
,gB;QAAgB,cAAA,SAAhB,M;QACI,IAAI,UAAU,OAAV,CAAJ,C;UACI,IAAI,KAAJ,C;YAAW,OAAO,I;UACIB
,SAAS,O;UACT,QAAQ,I;;;MAGhB,IAAI,CAAC,KAAL,C;QAAy,OAAO,I;MACnB,OAAO,M;K;iGAGX,gC;M
AMoB,Q;MAFhB,aAAuB,I;MACvB,YAAy,K;MACZ,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,IA
AI,UAAU,OAAV,CAAJ,C;UACI,IAAI,KAAJ,C;YAAW,OAAO,I;UACIB,SAAS,O;UACT,QAAQ,I;;;MAGhB,IA
AI,CAAC,KAAL,C;QAAy,OAAO,I;MACnB,OAAO,M;K;iGAGX,yB;MAAA,oC;MAAA,gC;MAAA,uC;QAMo
B,Q;QAFhB,aAAoB,I;QACpB,YAAy,K;QACZ,wBAAgB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UACI,
IAAI,UAAU,oBAAV,CAAJ,C;YACI,IAAI,KAAJ,C;cAAW,OAAO,I;YACIB,SAAS,O;YACT,QAAQ,I;;;QAGhB,I
AAI,CAAC,KAAL,C;UAAy,OAAO,I;QACnB,OAAO,M;O;KAdX,C;IAiBA,4B;McvqGI,IAAI,Ed+qGI,KAAK,Cc
/qGT,CAAJ,C;QACI,cd8qGc,sD;Qc7qGd,MAAM,gCAAyB,OAAQ,WAAjC,C;;Md8qGV,OAAO,oBAAoB,gBAA
V,mBAAO,CAAP,IAAU,EAAC,CAAd,CAApB,C;K;IAGX,8B;McnrGI,IAAI,Ed2rGI,KAAK,Cc3rGT,CAAJ,C;QA
CI,cd0rGc,sD;QczrGd,MAAM,gCAAyB,OAAQ,WAAjC,C;;Md0rGV,OAAO,sBAAoB,gBAAV,mBAAO,CAAP,I
AAU,EAAC,CAAd,CAApB,C;K;IAGX,8B;Mc/rGI,IAAI,EdusGI,KAAK,CcvsGT,CAAJ,C;QACI,cdssGc,sD;Qers
Gd,MAAM,gCAAyB,OAAQ,WAAjC,C;;MdssGV,OAAO,sBAAoB,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd,
CAApB,C;K;IAGX,8B;Mc3sGI,IAAI,EdmtGI,KAAK,CentGT,CAAJ,C;QACI,cdktGc,sD;QcjtGd,MAAM,gCAAy
B,OAAQ,WAAjC,C;;MdktGV,OAAO,sBAAoB,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd,CAApB,C;K;IAGX,
8B;McvtGI,IAAI,Ed+tGI,KAAK,Cc/tGT,CAAJ,C;QACI,cd8tGc,sD;Qc7tGd,MAAM,gCAAyB,OAAQ,WAAjC,C;;
Md8tGV,OAAO,sBAAoB,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd,CAApB,C;K;IAGX,8B;McnuGI,IAAI,Ed
2uGI,KAAK,Cc3uGT,CAAJ,C;QACI,cd0uGc,sD;QczuGd,MAAM,gCAAyB,OAAQ,WAAjC,C;;Md0uGV,OAAO,
sBAAoB,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd,CAApB,C;K;IAGX,8B;Mc/uGI,IAAI,EduvGI,KAAK,Ccuv
GT,CAAJ,C;QACI,cdsvGc,sD;QcervGd,MAAM,gCAAyB,OAAQ,WAAjC,C;;MdsvGV,OAAO,sBAAoB,gBAAV,
mBAAO,CAAP,IAAU,EAAC,CAAd,CAApB,C;K;IAGX,8B;Mc3vGI,IAAI,EdmwGI,KAAK,CcnwGT,CAAJ,C;Q
ACI,cdkwGc,sD;QcjkGd,MAAM,gCAAyB,OAAQ,WAAjC,C;;MdkwGV,OAAO,sBAAoB,gBAAV,mBAAO,CA
AP,IAAU,EAAC,CAAd,CAApB,C;K;IAGX,8B;McvwGI,IAAI,Ed+wGI,KAAK,Cc/wGT,CAAJ,C;QACI,cd8wGc,s
D;Qc7wGd,MAAM,gCAAyB,OAAQ,WAAjC,C;;Md8wGV,OAAO,sBAAoB,gBAAV,mBAAO,CAAP,IAAU,EA
Ac,CAAd,CAApB,C;K;IAGX,gC;McnxGI,IAAI,Ed2xGI,KAAK,Cc3xGT,CAAJ,C;QACI,cd0xGc,sD;QczxGd,MA
AM,gCAAyB,OAAQ,WAAjC,C;;Md0xGV,OAAO,gBAAgB,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd,CAAh
B,C;K;IAGX,kC;Mc/xGI,IAAI,EduyGI,KAAK,CcuyGT,CAAJ,C;QACI,cdsyGc,sD;QcryGd,MAAM,gCAAyB,OA
AQ,WAAjC,C;;MdsyGV,OAAO,kBAAgB,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd,CAAhB,C;K;IAGX,kC;
Mc3yGI,IAAI,EdmzGI,KAAK,CcnzGT,CAAJ,C;QACI,cdkzGc,sD;QczGd,MAAM,gCAAyB,OAAQ,WAAjC,C;;
MdkzGV,OAAO,kBAAgB,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd,CAAhB,C;K;IAGX,kC;McvzGI,IAAI,Ed
+zGI,KAAK,Cc/zGT,CAAJ,C;QACI,cd8zGc,sD;Qc7zGd,MAAM,gCAAyB,OAAQ,WAAjC,C;;Md8zGV,OAAO,k
BAAgB,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd,CAAhB,C;K;IAGX,kC;Mcn0GI,IAAI,Ed20GI,KAAK,Cc30
GT,CAAJ,C;QACI,cd00Gc,sD;Qcz0Gd,MAAM,gCAAyB,OAAQ,WAAjC,C;;Md00GV,OAAO,kBAAgB,gBAAV,
mBAAO,CAAP,IAAU,EAAC,CAAd,CAAhB,C;K;IAGX,kC;Mc/0GI,IAAI,Edu1GI,KAAK,Ccv1GT,CAAJ,C;QAC
I,cds1Gc,sD;Qcr1Gd,MAAM,gCAAyB,OAAQ,WAAjC,C;;Mds1GV,OAAO,kBAAgB,gBAAV,mBAAO,CAAP,IA
AU,EAAC,CAAd,CAAhB,C;K;IAGX,kC;Mc31GI,IAAI,Edm2GI,KAAK,Ccn2GT,CAAJ,C;QACI,cdk2Gc,sD;Qcj2
Gd,MAAM,gCAAyB,OAAQ,WAAjC,C;;Mdk2GV,OAAO,kBAAgB,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd
,CAAhB,C;K;IAGX,kC;Mcv2GI,IAAI,Ed+2GI,KAAK,Cc/2GT,CAAJ,C;QACI,cd82Gc,sD;Qc72Gd,MAAM,gCA

AyB,OAAQ,WAAjC,C;;Md82GV,OAAO,kBAAGb,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd,CAAhB,C;K;IA
GX,kC;Mcn3GI,IAAI,Ed23GI,KAAC,Cc33GT,CAAJ,C;QACI,cd03Gc,sD;Qcz3Gd,MAAM,gCAAyB,OAAQ,WA
AjC,C;;Md03GV,OAAO,kBAAGb,gBAAV,mBAAO,CAAP,IAAU,EAAC,CAAd,CAAhB,C;K;gGAGX,yB;MAAA
,8D;MAAA,4C;MAAA,qD;MAAA,uC;QAMI,iBAAC,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,UA
AK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,W;O;KAXX,C;kGAcA
,yB;MAAA,8D;MAAA,2C;MAAA,qD;MAAA,uC;QAMI,iBAAC,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,
UAAU,UAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,W;O;KAX
X,C;kGAcA,yB;MAAA,8D;MAAA,4C;MAAA,qD;MAAA,uC;QAMI,iBAAC,wBAAd,WAA+B,CAA/B,U;UACI,I
AAI,CAAC,UAAU,UAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO
,W;O;KAXX,C;kGAcA,yB;MAAA,8D;MAAA,4C;MAAA,qD;MAAA,uC;QAMI,iBAAC,wBAAd,WAA+B,CAA/
B,U;UACI,IAAI,CAAC,UAAU,UAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;Q
AGf,OAAO,W;O;KAXX,C;kGAcA,yB;MAAA,8D;MAAA,4C;MAAA,qD;MAAA,uC;QAMI,iBAAC,wBAAd,WA
A+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,UAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,
IAAL,C;;;QAGf,OAAO,W;O;KAXX,C;kGAcA,yB;MAAA,8D;MAAA,4C;MAAA,qD;MAAA,uC;QAMI,iBAAC,
wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,UAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,Q
AAQ,CAAR,IAAL,C;;;QAGf,OAAO,W;O;KAXX,C;kGAcA,yB;MAAA,8D;MAAA,4C;MAAA,qD;MAAA,uC;Q
AMI,iBAAC,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,UAAK,KAAL,CAAV,CAAL,C;YACI,OAA
O,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,W;O;KAXX,C;kGAcA,yB;MAAA,8D;MAAA,4C;MAAA,qD;
MAAA,uC;QAMI,iBAAC,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,UAAK,KAAL,CAAV,CAAL,C
;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,W;O;KAXX,C;kGAcA,yB;MAAA,8D;MAAA,oC;
MAAA,4C;MAAA,qD;MAAA,uC;QAMI,iBAAC,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,sBAAK,
KAAL,EAHV,CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,W;O;KAXX,C;wFACa,yB;
MAAA,+D;MAAA,uC;QAQiB,Q;QAFb,eAAe,K;QACf,WAAW,gB;QACX,wBAAa,SAAb,gB;UAAa,WAAA,SA
Ab,M;UACI,IAAI,QAAJ,C;YACI,IAAK,WAAI,IAAJ,C;eACJ,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACD,IAAK,
WAAI,IAAJ,C;YACL,WAAW,I;;;QAEEnB,OAAO,I;O;KafX,C;0FakBA,yB;MAAA,+D;MAAA,uC;QAQiB,Q;QA
Fb,eAAe,K;QACf,WAAW,gB;QACX,wBAAa,SAAb,gB;UAAa,WAAA,SAAb,M;UACI,IAAI,QAAJ,C;YACI,IA
AK,WAAI,IAAJ,C;eACJ,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACD,IAAK,WAAI,IAAJ,C;YACL,WAAW,I;;;Q
AEnB,OAAO,I;O;KafX,C;0FakBA,yB;MAAA,+D;MAAA,uC;QAQiB,Q;QAFb,eAAe,K;QACf,WAAW,gB;QAC
X,wBAAa,SAAb,gB;UAAa,WAAA,SAAb,M;UACI,IAAI,QAAJ,C;YACI,IAAK,WAAI,IAAJ,C;eACJ,IAAI,CAA
C,UAAU,IAAV,CAAL,C;YACD,IAAK,WAAI,IAAJ,C;YACL,WAAW,I;;;QAEEnB,OAAO,I;O;KafX,C;0FakBA,
yB;MAAA,+D;MAAA,uC;QAQiB,Q;QAFb,eAAe,K;QACf,WAAW,gB;QACX,wBAAa,SAAb,gB;UAAa,WAAA,
SAAb,M;UACI,IAAI,QAAJ,C;YACI,IAAK,WAAI,IAAJ,C;eACJ,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACD,IA
AK,WAAI,IAAJ,C;YACL,WAAW,I;;;QAEEnB,OAAO,I;O;KafX,C;0FakBA,yB;MAAA,+D;MAAA,uC;QAQiB,Q
;QAFb,eAAe,K;QACf,WAAW,gB;QACX,wBAAa,SAAb,gB;UAAa,WAAA,SAAb,M;UACI,IAAI,QAAJ,C;YACI,
IAAK,WAAI,IAAJ,C;eACJ,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACD,IAAK,WAAI,IAAJ,C;YACL,WAAW,I;;;
QAEEnB,OAAO,I;O;KafX,C;0FakBA,yB;MAAA,+D;MAAA,uC;QAQiB,Q;QAFb,eAAe,K;QACf,WAAW,gB;QA
CX,wBAAa,SAAb,gB;UAAa,WAAA,SAAb,M;UACI,IAAI,QAAJ,C;YACI,IAAK,WAAI,IAAJ,C;eACJ,IAAI,CAA
C,UAAU,IAAV,CAAL,C;YACD,IAAK,WAAI,IAAJ,C;YACL,WAAW,I;;;QAEEnB,OAAO,I;O;KafX,C;0FakB
A,yB;MAAA,+D;MAAA,uC;QAQiB,Q;QAFb,eAAe,K;QACf,WAAW,gB;QACX,wBAAa,SAAb,gB;UAAa,WAA
A,SAAb,M;UACI,IAAI,QAAJ,C;YACI,IAAK,WAAI,IAAJ,C;eACJ,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACD,I
AAK,WAAI,IAAJ,C;YACL,WAAW,I;;;QAEEnB,OAAO,I;O;KafX,C;0FakBA,yB;MAAA,+D;MAAA,uC;QAQiB,
Q;QAFb,eAAe,K;QACf,WAAW,gB;QACX,wBAAa,SAAb,gB;UAAa,WAAA,SAAb,M;UACI,IAAI,QAAJ,C;YA
CI,IAAK,WAAI,IAAJ,C;eACJ,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACD,IAAK,WAAI,IAAJ,C;YACL,WAAW,
I;;;QAEEnB,OAAO,I;O;KafX,C;0FakBA,yB;MAAA,+D;MAAA,oC;MAAA,gC;MAAA,uC;QAQiB,Q;QAFb,eAA
e,K;QACf,WAAW,gB;QACX,wBAAa,SAAb,gB;UAAa,WAAb,UAAa,SAAb,O;UACI,IAAI,QAAJ,C;YACI,IAA
K,WAAI,iBAAJ,C;eACJ,IAAI,CAAC,UAAU,iBAAV,CAAL,C;YACD,IAAK,WAAI,iBAAJ,C;YACL,WAAW,I;;;
QAEEnB,OAAO,I;O;KafX,C;kFakBA,yB;MAAA,+D;MAAA,uC;QAMW,kBAAS,gB;QAmgBA,Q;QAAhB,iD;U
AAgB,cAAhB,e;UAAsB,IAngBU,SAmgBN,CAAU,OAAV,CAAJ,C;YAAwB,WAAy,WAAI,OAAJ,C;;QAngB1D

,OAogBO,W;O;KA1gBX,C;oFASA,yB;MAAA,+D;MAAA,uC;QAMW,kBAAS,gB;QAogBA,Q;QAaHb,iD;UAAgB,cAAhB,e;UAAsB,IApgBa,SAogBT,CAAU,OAAV,CAAJ,C;YAAwB,WAAy,WAAI,OAAJ,C;;QApgB1D,OAqgBO,W;O;KA3gBX,C;oFASA,yB;MAAA,+D;MAAA,uC;QAMW,kBAAS,gB;QAqgBA,Q;QAaHb,iD;UAAgB,cAAhB,e;UAAsB,IArgBc,SAqgBV,CAAU,OAAV,CAAJ,C;YAAwB,WAAy,WAAI,OAAJ,C;;QArgB1D,OAsgBO,W;O;KA5gBX,C;oFASA,yB;MAAA,+D;MAAA,uC;QAMW,kBAAS,gB;QAsgBA,Q;QAaHb,iD;UAAgB,cAAhB,e;UAAsB,IAtgBY,SAsgBR,CAAU,OAAV,CAAJ,C;YAAwB,WAAy,WAAI,OAAJ,C;;QAtgB1D,OAuBBO,W;O;KA7gBX,C;oFASA,yB;MAAA,+D;MAAA,uC;QAMW,kBAAS,gB;QAugBA,Q;QAaHb,iD;UAAgB,cAAhB,e;UAAsB,IAvgBa,SAugBT,CAAU,OAAV,CAAJ,C;YAAwB,WAAy,WAAI,OAAJ,C;;QAvB1D,OAwbBO,W;O;KA9gBX,C;oFASA,yB;MAAA,+D;MAAA,uC;QAMW,kBAAS,gB;QAwgBA,Q;QAaHb,iD;UAAgB,cAAhB,e;UAAsB,IAxgBc,SAwgBV,CAAU,OAAV,CAAJ,C;YAAwB,WAAy,WAAI,OAAJ,C;;QAxgB1D,OAYgBO,W;O;KA/gBX,C;oFASA,yB;MAAA,+D;MAAA,uC;QAMW,kBAAS,gB;QAYgBA,Q;QAaHb,iD;UAAgB,cAAhB,e;UAAsB,IAzgBe,SAygBX,CAAU,OAAV,CAAJ,C;YAAwB,WAAy,WAAI,OAAJ,C;;QAZgB1D,OA0gBO,W;O;KAhhBX,C;oFASA,yB;MAAA,+D;MAAA,uC;QAMW,kBAAS,gB;QA0gBA,Q;QAaHb,iD;UAAgB,cAAhB,e;UAAsB,IA1gBgB,SA0gBZ,CAAU,OAAV,CAAJ,C;YAAwB,WAAy,WAAI,OAAJ,C;;QA1gB1D,OA2gBO,W;O;KAjhBX,C;oFASA,yB;MAAA,+D;MAAA,uC;QAMW,kBAAS,gB;QA2gBA,Q;QAaHb,iD;UAAgB,cAAhB,e;UAAsB,IA3gBa,SA2gBT,CAAU,oBAAV,CAAJ,C;YAAwB,WAAy,WAAI,oBAAJ,C;;QA3gB1D,OA4gBO,W;O;KA1hBX,C;gGASA,yB;MAAA,+D;MAAA,uC;QAQW,kBAAGB,gB;QASgTV,gB;QADb,YAAy,C;QACZ,iD;UAAa,WAAb,e;UA16SI,IApGmC,SAoG/B,Eak6SkB,cA16SIB,Eak6SkB,sBA16SIB,Wak6S2B,IA16S3B,CAAJ,C;YAA2C,sBAk6SZ,IA16SY,C;;QApg/C,OASGO,W;O;KA9GX,C;kGAWA,yB;MAAA,+D;MAAA,uC;QAQW,kBAAGB,gB;QAqgTV,gB;QADb,YAAy,C;QACZ,iD;UAAa,WAAb,e;UA95SI,IAvGsC,SAuGIC,EA85SkB,cA95SIB,EA85SkB,sBA95SIB,WA85S2B,IA95S3B,CAAJ,C;YAA2C,sBA85SZ,IA95SY,C;;QAvG/C,OAYGO,W;O;KAjHX,C;kGAWA,yB;MAAA,+D;MAAA,uC;QAQW,kBAAGB,gB;QAogTV,gB;QADb,YAAy,C;QACZ,iD;UAAa,WAAb,e;UA15SI,IA1GuC,SA0GnC,EA05SkB,cA15SIB,EA05SkB,sBA15SIB,WA05S2B,IA15S3B,CAAJ,C;YAA2C,sBA05SZ,IA15SY,C;;QA1G/C,OA4GO,W;O;KApHX,C;kGAWA,yB;MAAA,+D;MAAA,uC;QAQW,kBAAGB,gB;QAmgTV,gB;QADb,YAAy,C;QACZ,iD;UAAa,WAAb,e;UA5SI,IA7GqC,SA6GjC,EAs5SkB,cAt5SIB,EAs5SkB,sBA5SIB,WAs5S2B,IA5S3B,CAAJ,C;YAA2C,sBA5SZ,IA5SY,C;;QA7G/C,OA+GO,W;O;KAvHX,C;kGAWA,yB;MAAA,+D;MAAA,uC;QAQW,kBAAGB,gB;QakgTV,gB;QADb,YAAy,C;QACZ,iD;UAAa,WAAb,e;UA15SI,IAhHsC,SAgHIC,Eak5SkB,cA15SIB,Eak5SkB,sBA15SIB,Wak5S2B,IA15S3B,CAAJ,C;YAA2C,sBAk5SZ,IA15SY,C;;QAhH/C,OAKHO,W;O;KA1HX,C;kGAWA,yB;MAAA,+D;MAAA,uC;QAQW,kBAAGB,gB;QAigTV,gB;QADb,YAAy,C;QACZ,iD;UAAa,WAAb,e;UA94SI,IANHuC,SAmHnC,EA84SkB,cA94SIB,EA84SkB,sBA94SIB,WA84S2B,IA94S3B,CAAJ,C;YAA2C,sBA84SZ,IA94SY,C;;QAnH/C,OAqHO,W;O;KA7HX,C;kGAWA,yB;MAAA,+D;MAAA,uC;QAQW,kBAAGB,gB;QAaggTV,gB;QADb,YAAy,C;QACZ,iD;UAAa,WAAb,e;UA14SI,IAthwC,SAsHpC,EA04SkB,cA14SIB,EA04SkB,sBA14SIB,WA04S2B,IA14S3B,CAAJ,C;YAA2C,sBA04SZ,IA14SY,C;;QAtH/C,OAwhO,W;O;KAhIX,C;kGAWA,yB;MAAA,+D;MAAA,uC;QAQW,kBAAGB,gB;QA+/SV,gB;QADb,YAAy,C;QACZ,iD;UAAa,WAAb,e;UA4SI,IAzHyC,SAyHrC,EAs4SkB,cAt4SIB,EAs4SkB,sBA4SIB,WAs4S2B,IA4S3B,CAAJ,C;YAA2C,sBA4SZ,IA4SY,C;;QAzH/C,OA2HO,W;O;KANIX,C;kGAWA,yB;MAAA,+D;MAAA,uC;QAQW,kBAAGB,gB;QA8/SV,gB;QADb,YAAy,C;QACZ,iD;UAAa,WAAb,e;UA4SI,IAAI,Wak6SkB,cA16SIB,Eak6SkB,sBA16SIB,Wak6S2B,IA16S3B,CAAJ,C;UAA2C,sBAk6SZ,IA16SY,C;;MAE/C,OAAO,W;K;qGAGX,6C;MAu6SiB,gB;MADb,YAAy,C;MACZ,iD;QAAa,WAAb,e;QA95SI,IAAI,WA85SkB,cA95SIB,EA85SkB,sBA95SIB,WA85S2B,IA95S3B,CAAJ,C;UAA2C,sBA85SZ,IA95SY,C;;MAE/C,OAAO,W;K;sGAGX,6C;MAm6SiB,gB;MADb,YAAy,C;MACZ,iD;QAAa,WAAb,e;QA15SI,IAAI,WA05SkB,cA15SIB,EA05SkB,sBA15SIB,WA05S2B,IA15S3B,CAAJ,C;UAA2C,sBA05SZ,IA15SY,C;;MAE/C,OAAO,W;K;qGAGX,6C;MA+5SiB,gB;MADb,YAAy,C;MACZ,iD;QAAa,WAAb,e;QA5SI,IAAI,WAs5SkB,cAt5SIB,EAs5SkB,sBA5SIB,WAs5S2B,IA5S3B,CAAJ,C;UAA2C,sBA5SZ,IA5SY,C;;MAE/C,OAAO,W;K;sGAGX,6C;MA25SiB,gB;MADb,YAAy,C;MACZ,iD;QAAa,WAAb,e;QA15SI,IAAI,Wak5SkB,cA15SIB,Eak5SkB,sBA15SIB,Wak5S2B,IA15S3B,CAAJ,C;UAA2C,sBAk5SZ,IA15SY,C;;MAE/C,OAAO,W;K;sGAGX,6C

QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,WAAy,WAAI,OAAJ,C;;MAC1D,OA
AO,W;K;wFAGX,6C;MAMoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU
,OAAV,CAAJ,C;UAAwB,WAAy,WAAI,OAAJ,C;;MAC1D,OAAO,W;K;wFAGX,6C;MAMoB,Q;MAAhB,wBA
AgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,WAAy,WAAI,OAAJ,
C;;MAC1D,OAAO,W;K;wFAGX,6C;MAMoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAA
sB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,WAAy,WAAI,OAAJ,C;;MAC1D,OAAO,W;K;wFAGX,6C;MAMoB,Q
;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,WAA
Y,WAAI,OAAJ,C;;MAC1D,OAAO,W;K;wFAGX,6C;MAMoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,
SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,WAAy,WAAI,OAAJ,C;;MAC1D,OAAO,W;K;wFAG
X,6C;MAMoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C
;UAAwB,WAAy,WAAI,OAAJ,C;;MAC1D,OAAO,W;K;wFAGX,yB;MAAA,oC;MAAA,gC;MAAA,oD;QAMoB,
Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UAAsB,IAAI,UAAU,oBAAV,CAAJ,C;YA
AwB,WAAy,WAAI,oBAAJ,C;;QAC1D,OAAO,W;O;KAPX,C;IAUA,mC;MAII,IAAI,OAAQ,UAAZ,C;QAAuB,O
Mhtle,W;;MNitItC,OAA4D,OAArD,yBAAY,OAAQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3B,CA
AqD,C;K;IAGhE,qC;MAII,IAAI,OAAQ,UAAZ,C;QAAuB,OMxtle,W;;MNytItC,Oe7rIsC,Of6rI/B,yBAAY,OAAQ
,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3B,Ce7rI+B,C;K;IfgsI1C,qC;MAII,IAAI,OAAQ,UAAZ,C;
QAAuB,OMhule,W;;MNiultC,Oe7rIuC,Of6rIhC,yBAAY,OAAQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB
,IAA3B,Ce7rIgC,C;K;IfgsI3C,qC;MAII,IAAI,OAAQ,UAAZ,C;QAAuB,OMxule,W;;MNyultC,Oe7rIqC,Of6rI9B,y
BAAY,OAAQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3B,Ce7rI8B,C;K;IfgsIzC,qC;MAII,IAAI,OA
AQ,UAAZ,C;QAAuB,OMhvle,W;;MNivItC,Oe7rIsC,Of6rI/B,yBAAY,OAAQ,MAApB,EAA2B,OAAQ,aAAR,GA
AuB,CAAvB,IAA3B,Ce7rI+B,C;K;IfgsI1C,qC;MAII,IAAI,OAAQ,UAAZ,C;QAAuB,OMxvle,W;;MNyvItC,Oe7rI
uC,Of6rIhC,yBAAY,OAAQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3B,Ce7rIgC,C;K;IfgsI3C,qC;M
AII,IAAI,OAAQ,UAAZ,C;QAAuB,OMhwle,W;;MNiwItC,Oe7rIwC,Of6rIjC,yBAAY,OAAQ,MAApB,EAA2B,O
AAQ,aAAR,GAAuB,CAAvB,IAA3B,Ce7rIiC,C;K;IfgsI5C,qC;MAII,IAAI,OAAQ,UAAZ,C;QAAuB,OMxwle,W;;
MNywItC,Oe7rIyC,Of6rIlC,0BAAY,OAAQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3B,Ce7rIkC,C;
K;IfgsI7C,qC;MAII,IAAI,OAAQ,UAAZ,C;QAAuB,OMhxle,W;;MNixItC,OAA4D,SAArD,0BAAY,OAAQ,MAA
pB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3B,CAAqD,C;K;IAGhE,qC;MAOkB,Q;MAHd,WAAmB,wBAAR
,OAAQ,EAAwB,EAAxB,C;MACnB,IAAI,SAAQ,CAAZ,C;QAae,OAAO,W;MActB,WAAW,iBAaAa,IAAb,C;M
ACG,yB;MAAd,OAAc,cAAd,C;QAac,uB;QACV,IAAK,WAAI,UAAI,KAAJ,CAAJ,C;;MAET,OAAO,I;K;IAGX,
qC;MAOkB,Q;MAHd,WAAmB,wBAAR,OAAQ,EAAwB,EAAxB,C;MACnB,IAAI,SAAQ,CAAZ,C;QAae,OAA
O,W;MActB,WAAW,iBAAgB,IAAhB,C;MACG,yB;MAAd,OAAc,cAAd,C;QAac,uB;QACV,IAAK,WAAI,UAA
I,KAAJ,CAAJ,C;;MAET,OAAO,I;K;IAGX,sC;MAOkB,Q;MAHd,WAAmB,wBAAR,OAAQ,EAAwB,EAAxB,C;
MACnB,IAAI,SAAQ,CAAZ,C;QAae,OAAO,W;MActB,WAAW,iBAAiB,IAAjB,C;MACG,yB;MAAd,OAAc,cA
Ad,C;QAac,uB;QACV,IAAK,WAAI,UAAI,KAAJ,CAAJ,C;;MAET,OAAO,I;K;IAGX,sC;MAOkB,Q;MAHd,WA
AmB,wBAAR,OAAQ,EAAwB,EAAxB,C;MACnB,IAAI,SAAQ,CAAZ,C;QAae,OAAO,W;MActB,WAAW,iBA
Ae,IAAf,C;MACG,yB;MAAd,OAAc,cAAd,C;QAac,uB;QACV,IAAK,WAAI,UAAI,KAAJ,CAAJ,C;;MAET,OA
AO,I;K;IAGX,sC;MAOkB,Q;MAHd,WAAmB,wBAAR,OAAQ,EAAwB,EAAxB,C;MACnB,IAAI,SAAQ,CAAZ,
C;QAae,OAAO,W;MActB,WAAW,iBAAgB,IAAhB,C;MACG,yB;MAAd,OAAc,cAAd,C;QAac,uB;QACV,IAA
K,WAAI,UAAI,KAAJ,CAAJ,C;;MAET,OAAO,I;K;IAGX,sC;MAOkB,Q;MAHd,WAAmB,wBAAR,OAAQ,EAA
wB,EAAxB,C;MACnB,IAAI,SAAQ,CAAZ,C;QAae,OAAO,W;MActB,WAAW,iBAAiB,IAAjB,C;MACG,yB;M
AAd,OAAc,cAAd,C;QAac,uB;QACV,IAAK,WAAI,UAAI,KAAJ,CAAJ,C;;MAET,OAAO,I;K;IAGX,sC;MAOkB
,Q;MAHd,WAAmB,wBAAR,OAAQ,EAAwB,EAAxB,C;MACnB,IAAI,SAAQ,CAAZ,C;QAae,OAAO,W;MAct
B,WAAW,iBAAkB,IAAiB,C;MACG,yB;MAAd,OAAc,cAAd,C;QAac,uB;QACV,IAAK,WAAI,UAAI,KAAJ,CA
AJ,C;;MAET,OAAO,I;K;IAGX,sC;MAOkB,Q;MAHd,WAAmB,wBAAR,OAAQ,EAAwB,EAAxB,C;MACnB,IA
AI,SAAQ,CAAZ,C;QAae,OAAO,W;MActB,WAAW,iBAAmB,IAAnB,C;MACG,yB;MAAd,OAAc,cAAd,C;QA
Ac,uB;QACV,IAAK,WAAI,UAAI,KAAJ,CAAJ,C;;MAET,OAAO,I;K;IAGX,sC;MAOkB,Q;MAHd,WAAmB,wB
AAR,OAAQ,EAAwB,EAAxB,C;MACnB,IAAI,SAAQ,CAAZ,C;QAae,OAAO,W;MActB,WAAW,iBAAgB,IAA
hB,C;MACG,yB;MAAd,OAAc,cAAd,C;QAac,uB;QACV,IAAK,WAAI,sBAAI,KAAJ,EAAJ,C;;MAET,OAAO,I;

K;IAGX,wC;MAMwB,UACT,M;MAHX,aAAa,aAAa,SAAb,EAAmB,OAAQ,KAA3B,C;MACb,kBAakB,C;MAC E,yB;MAApB,OAAoB,cAApB,C;QAAoB,6B;QACHB,OAAO,oBAAP,EAAO,4BAAP,YAAwB,UAAK,WAAL,C; ;MAE5B,OAAO,M;K;IAGX,0C;MAMwB,UACT,M;MAHX,aAAa,cAAU,OAAQ,KAAIB,C;MACb,kBAakB,C;M ACE,yB;MAApB,OAAoB,cAApB,C;QAAoB,6B;QACHB,OAAO,oBAAP,EAAO,4BAAP,YAAwB,UAAK,WAAL ,C;;MAE5B,OAAO,M;K;IAGX,0C;MAMwB,UACT,M;MAHX,aAAa,eAAW,OAAQ,KAAIB,C;MACb,kBAakB, C;MACE,yB;MAApB,OAAoB,cAApB,C;QAAoB,6B;QACHB,OAAO,oBAAP,EAAO,4BAAP,YAAwB,UAAK,W AAL,C;;MAE5B,OAAO,M;K;IAGX,0C;MAMwB,UACT,M;MAHX,aAAa,eAAS,OAAQ,KAAjB,C;MACb,kBAA kB,C;MACE,yB;MAApB,OAAoB,cAApB,C;QAAoB,6B;QACHB,OAAO,oBAAP,EAAO,4BAAP,YAAwB,UAAK ,WAAL,C;;MAE5B,OAAO,M;K;IAGX,0C;MAMwB,UACT,M;MAHX,aAAa,iBAAU,OAAQ,KAAIB,C;MACb,k BAAkB,C;MACE,yB;MAApB,OAAoB,cAApB,C;QAAoB,6B;QACHB,OAAO,oBAAP,EAAO,4BAAP,YAAwB,U AAK,WAAL,C;;MAE5B,OAAO,M;K;IAGX,0C;MAMwB,UACT,M;MAHX,aAAa,iBAAW,OAAQ,KAAIB,C;M ACb,kBAakB,C;MACE,yB;MAApB,OAAoB,cAApB,C;QAAoB,6B;QACHB,OAAO,oBAAP,EAAO,4BAAP,YA AwB,UAAK,WAAL,C;;MAE5B,OAAO,M;K;IAGX,0C;MAMwB,UACT,M;MAHX,aAAa,iBAAY,OAAQ,KAAp B,C;MACb,kBAakB,C;MACE,yB;MAApB,OAAoB,cAApB,C;QAAoB,6B;QACHB,OAAO,oBAAP,EAAO,4BAA P,YAAwB,UAAK,WAAL,C;;MAE5B,OAAO,M;K;IAGX,0C;MAMwB,UACT,M;MAHX,aAAa,oBAAa,OAAQ,K AArB,C;MACb,kBAakB,C;MACE,yB;MAApB,OAAoB,cAApB,C;QAAoB,6B;QACHB,OAAO,oBAAP,EAAO,4 BAAP,YAAwB,UAAK,WAAL,C;;MAE5B,OAAO,M;K;IAGX,0C;MAMwB,UACT,M;MAHX,aAAa,iBAAU,OA AQ,KAAIB,C;MACb,kBAakB,C;MACE,yB;MAApB,OAAoB,cAApB,C;QAAoB,6B;QACHB,OAAO,oBAAP,EA AO,4BAAP,YAAwB,UAAK,WAAL,C;;MAE5B,OAAO,M;K;IAGX,0C;MAIL,IAAI,OAAQ,UAAZ,C;QAAuB,OA AO,yBAAY,CAAZ,EAAe,CAAf,C;MAC9B,OAAO,yBAAY,OAAQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CA AvB,IAA3B,C;K;IAGX,0C;MAIL,IAAI,OAAQ,UAAZ,C;QAAuB,OAAO,cAAU,CAAV,C;MAC9B,OAAO,yBAA Y,OAAQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3B,C;K;IAGX,2C;MAIL,IAAI,OAAQ,UAAZ,C;Q AAuB,OAAO,eAAW,CAAX,C;MAC9B,OAAO,yBAAY,OAAQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB ,IAA3B,C;K;IAGX,2C;MAIL,IAAI,OAAQ,UAAZ,C;QAAuB,OAAO,eAAS,CAAT,C;MAC9B,OAAO,yBAAY,OA AQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3B,C;K;IAGX,2C;MAIL,IAAI,OAAQ,UAAZ,C;QAAuB ,OAAO,iBAAU,CAAV,C;MAC9B,OAAO,yBAAY,OAAQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA 3B,C;K;IAGX,2C;MAIL,IAAI,OAAQ,UAAZ,C;QAAuB,OAAO,iBAAW,CAAX,C;MAC9B,OAAO,yBAAY,OAA Q,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3B,C;K;IAGX,2C;MAIL,IAAI,OAAQ,UAAZ,C;QAAuB, OAAO,iBAAY,CAAZ,C;MAC9B,OAAO,yBAAY,OAAQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3 B,C;K;IAGX,2C;MAIL,IAAI,OAAQ,UAAZ,C;QAAuB,OAAO,oBAAa,CAAb,C;MAC9B,OAAO,0BAAAY,OAAQ, MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3B,C;K;IAGX,2C;MAIL,IAAI,OAAQ,UAAZ,C;QAAuB,OA AO,iBAAU,CAAV,C;MAC9B,OAAO,0BAAAY,OAAQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3B,C ;K;IAGX,4B;MAciB,Q;Mc3nJb,IAAI,EdqnJI,KAAK,CernJT,CAAJ,C;QACI,cdonJc,sD;QcnnJd,MAAM,gCAAYB, OAAQ,WAAjC,C;;MdonJV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,gBAAT,C;QAAY,OA AO,iB;MAcTb,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,CAAL,CAAP,C;MACnB,YAAY,C;MACZ, WAAW,iBAAa,CAAb,C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,IAAK,WAAL,IAAJ,C;QACL,I AAI,mCAAW,CAAf,C;UACI,K;;MAER,OAAO,I;K;IAGX,8B;MAciB,Q;McvpJb,IAAI,Ed2oJI,KAAK,Cc3oJT,CA AJ,C;QACI,cd0oJc,sD;QczoJd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Md0oJV,IAAI,MAAK,CAAT,C;QAAY,OA AO,W;MACnB,IAAI,KAAK,gBAAT,C;QAAY,OAAO,mB;MAcTb,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,U AAK,CAAL,CAAP,C;MACnB,YAAY,C;MACZ,WAAW,iBAAgB,CAAhB,C;MACX,wBAAa,SAAb,gB;QAAa, WAAA,SAAb,M;QACI,IAAK,WAAL,IAAJ,C;QACL,IAAI,mCAAW,CAAf,C;UACI,K;;MAER,OAAO,I;K;IAGX, 8B;MAciB,Q;McvqJb,IAAI,EdiqJI,KAAK,CcjqJT,CAAJ,C;QACI,cdgqJc,sD;Qc/pJd,MAAM,gCAAYB,OAAQ,W AAjC,C;;MdgqJV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,gBAAT,C;QAAY,OAAO,mB; MAcTb,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,CAAL,CAAP,C;MACnB,YAAY,C;MACZ,WAAW ,iBAAiB,CAAjB,C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,IAAK,WAAL,IAAJ,C;QACL,IAAI, mCAAW,CAAf,C;UACI,K;;MAER,OAAO,I;K;IAGX,8B;MAciB,Q;Mc7rJb,IAAI,EdurJI,KAAK,CcvtJT,CAAJ,C; QACI,cdsrJc,sD;QcrrJd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MdsrJV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;M ACnB,IAAI,KAAK,gBAAT,C;QAAY,OAAO,mB;MAcTb,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,C

AAL,CAAP,C;MACnB,YAAY,C;MACZ,WAAW,iBA Ae,CAAf,C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,IAAK,WAAI,IAAJ,C;QACL,IAAI,mCAAW,CAAf,C;UACI,K;;MAER,OAAO,I;K;IAGX,8B;MAciB,Q;McntJb,IAAI,Ed6sJI,KAAK,Cc7sJT,CAAJ,C;QACI,cd4sJc,sD;Qc3sJd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Md4sJV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,gBAAT,C;QAAe,OAAO,mB;MACTb,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,CAAL,CAAP,C;MACnB,YAAY,C;MACZ,WAAW,iBAAgB,CAAhB,C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,IAAK,WAAI,IAAJ,C;QACL,IAAI,mCAAW,CAAf,C;UACI,K;;MAER,OAAO,I;K;IAGX,8B;MAciB,Q;MczuJb,IAAI,EdmuJI,KAAK,CcnuJT,CAAJ,C;QACI,cdkuJc,sD;QcjuJd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Md4sJV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,gBAAT,C;QAAe,OAAO,mB;MACTb,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,CAAL,CAAP,C;MACnB,YAAY,C;MACZ,WAAW,iBAAiB,CAAjB,C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,IAAK,WAAI,IAAJ,C;QACL,IAAI,mCAAW,CAAf,C;UACI,K;;MAER,OAAO,I;K;IAGX,8B;MAciB,Q;McvJb,IAAI,EdyvJI,KAAK,CczvJT,CAAJ,C;QACI,cdwvJc,sD;QcqvJd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MdwwJV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,gBAAT,C;QAAe,OAAO,mB;MACTb,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,CAAL,CAAP,C;MACnB,YAAY,C;MACZ,WAAW,iBAAB,CAAIB,C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,IAAK,WAAI,IAAJ,C;QACL,IAAI,mCAAW,CAAf,C;UACI,K;;MAER,OAAO,I;K;IAGX,8B;MAciB,Q;McrxJb,IAAI,Ed+wJI,KAAK,Cc/wJT,CAAJ,C;QACI,cd8wJc,sD;Qc7wJd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Md8wJV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,gBAAT,C;QAAe,OAAO,mB;MACTb,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,CAAL,CAAP,C;MACnB,YAAY,C;MACZ,WAAW,iBAAmB,CAAnB,C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,IAAK,WAAI,IAAJ,C;QACL,IAAI,mCAAW,CAAf,C;UACI,K;;MAER,OAAO,I;K;IAGX,8B;MAciB,Q;Mc3yJb,IAAI,EdqyJI,KAAK,CcryJT,CAAJ,C;QACI,cdoyJc,sD;QcnyJd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MdoyJV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,gBAAT,C;QAAe,OAAO,mB;MACTb,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,sBAAB,CAAL,EAAP,C;MACnB,YAAY,C;MACZ,WAAW,iBAAgB,CAAhB,C;MACX,wBAAa,SAAb,gB;QAAa,WAAb,UAAa,SAAb,O;QACI,IAAK,WAAI,iBAAJ,C;QACL,IAAI,mCAAW,CAAf,C;UACI,K;;MAER,OAAO,I;K;IAGX,gC;McnzJI,IAAI,Ed2zJI,KAAK,Cc3zJT,CAAJ,C;QACI,cd0zJc,sD;QczzJd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Md0zJV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,gB;MACX,IAAI,KAAK,IAAT,C;QAAe,OAAO,iB;MACTb,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,OAAO,CAAP,IAAL,CAAP,C;MACnB,WAAW,iBAAb,CAAb,C;MACX,iBAAc,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,WAAI,UAAK,KAAL,CAAJ,C;MACT,OAAO,I;K;IAGX,kC;Mct0JI,IAAI,Ed80JI,KAAK,Cc90JT,CAAJ,C;QACI,cd60Jc,sD;Qc50Jd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Md60JV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,gB;MACX,IAAI,KAAK,IAAT,C;QAAe,OAAO,mB;MACTb,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,OAAO,CAAP,IAAL,CAAP,C;MACnB,WAAW,iBAAgB,CAAhB,C;MACX,iBAAc,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,WAAI,UAAK,KAAL,CAAJ,C;MACT,OAAO,I;K;IAGX,kC;Mcz1JI,IAAI,Edi2JI,KAAK,Ccj2JT,CAAJ,C;QACI,cdg2Jc,sD;Qc/1Jd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Mdg2JV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,gB;MACX,IAAI,KAAK,IAAT,C;QAAe,OAAO,mB;MACTb,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,OAAO,CAAP,IAAL,CAAP,C;MACnB,WAAW,iBAAiB,CAAjB,C;MACX,iBAAc,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,WAAI,UAAK,KAAL,CAAJ,C;MACT,OAAO,I;K;IAGX,kC;Mc52JI,IAAI,Edo3JI,KAAK,Ccp3JT,CAAJ,C;QACI,cdm3Jc,sD;Qcl3Jd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Mdm3JV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,gB;MACX,IAAI,KAAK,IAAT,C;QAAe,OAAO,mB;MACTb,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,OAAO,CAAP,IAAL,CAAP,C;MACnB,WAAW,iBA Ae,CAAf,C;MACX,iBAAc,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,WAAI,UAAK,KAAL,CAAJ,C;MACT,OAAO,I;K;IAGX,kC;Mc/3JI,IAAI,Ed4JI,KAAK,Ccv4JT,CAAJ,C;QACI,cds4Jc,sD;Qcr4Jd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Mds4JV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,gB;MACX,IAAI,KAAK,IAAT,C;QAAe,OAAO,mB;MACTb,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,OAAO,CAAP,IAAL,CAAP,C;MACnB,WAAW,iBAAgB,CAAhB,C;MACX,iBAAc,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,WAAI,UAAK,KAAL,CAAJ,C;MACT,OAAO,I;K;IAGX,kC;Mcl5JI,IAAI,Ed05JI,KAAK,Cc15JT,CAAJ,C;QACI,cdy5Jc,sD;Qcx5Jd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Mdy5JV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,gB;MACX,IAAI,KAAK,IAAT,C;QAAe,OAAO,mB;MA

CtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,OAAO,CAAP,IAAL,CAAP,C;MACnB,WAAW,iBAAi
B,CAAjB,C;MACX,iBAAc,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,WAAI,UAAK,KAAL,CAAJ,C;
MACT,OAAO,I;K;IAGX,kC;Mcr6JI,IAAI,Ed66JI,KAAK,Cc76JT,CAAJ,C;QACI,cd46Jc,sD;Qc36Jd,MAAM,gCA
AyB,OAAQ,WAAjC,C;;Md46JV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,gB;MACX,IAAI,KA
AK,IAAT,C;QAae,OAAO,mB;MACTB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,OAAO,CAAP,IAA
L,CAAP,C;MACnB,WAAW,iBAaKB,CAAIB,C;MACX,iBAAc,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IA
AK,WAAI,UAAK,KAAL,CAAJ,C;MACT,OAAO,I;K;IAGX,kC;Mcx7JI,IAAI,Edg8JI,KAAK,Cch8JT,CAAJ,C;Q
ACI,cd+7Jc,sD;Qc97Jd,MAAM,gCAAyB,OAAQ,WAAjC,C;;Md+7JV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;
MACnB,WAAW,gB;MACX,IAAI,KAAK,IAAT,C;QAae,OAAO,mB;MACTB,IAAI,MAAK,CAAT,C;QAAY,OA
AO,OAAO,UAAK,OAAO,CAAP,IAAL,CAAP,C;MACnB,WAAW,iBAAmB,CAAnB,C;MACX,iBAAc,OAAO,C
AAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,WAAI,UAAK,KAAL,CAAJ,C;MACT,OAAO,I;K;IAGX,kC;Mc38JI,
IAAI,Edm9JI,KAAK,Ccn9JT,CAAJ,C;QACI,cdk9Jc,sD;Qcj9Jd,MAAM,gCAAyB,OAAQ,WAAjC,C;;Mdk9JV,IAA
I,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,gB;MACX,IAAI,KAAK,IAAT,C;QAae,OAAO,mB;MACT
B,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,sBAAK,OAAO,CAAP,IAAL,EAAP,C;MACnB,WAAW,iBAAGB
,CAAhB,C;MACX,iBAAc,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,WAAI,sBAAK,KAAL,EAJ,C;
MACT,OAAO,I;K;gGAGX,yB;MAAA,8D;MAAA,4C;MAAA,gD;MAAA,uC;QAMI,iBAAc,wBAAd,WAA+B,C
AA/B,U;UACI,IAAI,CAAC,UAAU,UAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,
C;;;QAGf,OAAO,iB;O;KAXX,C;kGAcA,yB;MAAA,8D;MAAA,2C;MAAA,gD;MAAA,uC;QAMI,iBAAc,wBAA
d,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,UAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,QAAQ,C
AAR,IAAL,C;;;QAGf,OAAO,iB;O;KAXX,C;kGAcA,yB;MAAA,8D;MAAA,4C;MAAA,gD;MAAA,uC;QAMI,iB
AAc,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,UAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAA
K,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,iB;O;KAXX,C;kGAcA,yB;MAAA,8D;MAAA,4C;MAAA,gD;MAAA,u
C;QAMI,iBAAc,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,UAAK,KAAL,CAAV,CAAL,C;YACI,O
AAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,iB;O;KAXX,C;kGAcA,yB;MAAA,8D;MAAA,4C;MAAA,g
D;MAAA,uC;QAMI,iBAAc,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,UAAK,KAAL,CAAV,CAAL
,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,iB;O;KAXX,C;kGAcA,yB;MAAA,8D;MAAA,4
C;MAAA,gD;MAAA,uC;QAMI,iBAAc,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,UAAK,KAAL,C
AAV,CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,iB;O;KAXX,C;kGAcA,yB;MAAA,
8D;MAAA,4C;MAAA,gD;MAAA,uC;QAMI,iBAAc,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,UA
AK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,iB;O;KAXX,C;kGAcA
,yB;MAAA,8D;MAAA,4C;MAAA,gD;MAAA,uC;QAMI,iBAAc,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,
UAAU,UAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,iB;O;KAX
X,C;kGAcA,yB;MAAA,8D;MAAA,oC;MAAA,4C;MAAA,gD;MAAA,uC;QAMI,iBAAc,wBAAd,WAA+B,CAA/
B,U;UACI,IAAI,CAAC,UAAU,sBAAK,KAAL,EAIV,CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;
QAGf,OAAO,iB;O;KAXX,C;wFAcA,yB;MAAA,+D;MAAA,uC;QAOiB,Q;QADb,WAAW,gB;QACX,wBAAa,S
AAb,gB;UAAa,WAAA,SAAb,M;UACI,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACI,K;UACJ,IAAK,WAAI,IAAJ,
C;;QAET,OAAO,I;O;KAZX,C;0FAeA,yB;MAAA,+D;MAAA,uC;QAOiB,Q;QADb,WAAW,gB;QACX,wBAAa,S
AAb,gB;UAAa,WAAA,SAAb,M;UACI,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACI,K;UACJ,IAAK,WAAI,IAAJ,
C;;QAET,OAAO,I;O;KAZX,C;0FAeA,yB;MAAA,+D;MAAA,uC;QAOiB,Q;QADb,WAAW,gB;QACX,wBAAa,S
AAb,gB;UAAa,WAAA,SAAb,M;UACI,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACI,K;UACJ,IAAK,WAAI,IAAJ,
C;;QAET,OAAO,I;O;KAZX,C;0FAeA,yB;MAAA,+D;MAAA,uC;QAOiB,Q;QADb,WAAW,gB;QACX,wBAAa,S
AAb,gB;UAAa,WAAA,SAAb,M;UACI,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACI,K;UACJ,IAAK,WAAI,IAAJ,
C;;QAET,OAAO,I;O;KAZX,C;0FAeA,yB;MAAA,+D;MAAA,uC;QAOiB,Q;QADb,WAAW,gB;QACX,wBAAa,S
AAb,gB;UAAa,WAAA,SAAb,M;UACI,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACI,K;UACJ,IAAK,WAAI,IAAJ,
C;;QAET,OAAO,I;O;KAZX,C;0FAeA,yB;MAAA,+D;MAAA,uC;QAOiB,Q;QADb,WAAW,gB;QACX,wBAAa,S
AAb,gB;UAAa,WAAA,SAAb,M;UACI,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACI,K;UACJ,IAAK,WAAI,IAAJ,

C;;QAET,OAAO,I;O;KAZX,C;0FAeA,yB;MAAA,+D;MAAA,uC;QAOiB,Q;QADb,WAAW,gB;QACX,wBAAa,S
AAb,gB;UAAa,WAAA,SAAb,M;UACI,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACI,K;UACJ,IAAK,WAAI,IAAJ,
C;;QAET,OAAO,I;O;KAZX,C;0FAeA,yB;MAAA,+D;MAAA,oC;MAAA,gC;MAAA,uC;QAOiB,Q;QADb,WAA
W,gB;QACX,wBAAa,SAAb,gB;UAAa,WAAb,UAAa,SAAb,O;UACI,IAAI,CAAC,UAAU,iBAAV,CAAL,C;YAC
I,K;UACJ,IAAK,WAAI,iBAAJ,C;;QAET,OAAO,I;O;KAZX,C;IAeA,4B;MAII,eAAe,CAAC,mBAAO,CAAP,IAA
D,IAAa,CAAb,I;MACf,IAAI,WAAW,CAAf,C;QAAkB,M;MACIB,mBAAmB,wB;MACnB,iBAAc,CAAd,WAAi
B,QAAjB,U;QACI,UAAU,UAAK,KAAL,C;QACV,UAAK,KAAL,IAAc,UAAK,YAAL,C;QACd,UAAK,YAAL,I
AAqB,G;QACrB,mC;;K;IAIR,8B;MAII,eAAe,CAAC,mBAAO,CAAP,IAAD,IAAa,CAAb,I;MACf,IAAI,WAAW,
CAAf,C;QAAkB,M;MACIB,mBAAmB,0B;MACnB,iBAAc,CAAd,WAAiB,QAAjB,U;QACI,UAAU,UAAK,KAA
L,C;QACV,UAAK,KAAL,IAAc,UAAK,YAAL,C;QACd,UAAK,YAAL,IAAqB,G;QACrB,mC;;K;IAIR,8B;MAII,
eAAe,CAAC,mBAAO,CAAP,IAAD,IAAa,CAAb,I;MACf,IAAI,WAAW,CAAf,C;QAAkB,M;MACIB,mBAAmB,
0B;MACnB,iBAAc,CAAd,WAAiB,QAAjB,U;QACI,UAAU,UAAK,KAAL,C;QACV,UAAK,KAAL,IAAc,UAAK
,YAAL,C;QACd,UAAK,YAAL,IAAqB,G;QACrB,mC;;K;IAIR,8B;MAII,eAAe,CAAC,mBAAO,CAAP,IAAD,IA
Aa,CAAb,I;MACf,IAAI,WAAW,CAAf,C;QAAkB,M;MACIB,mBAAmB,0B;MACnB,iBAAc,CAAd,WAAiB,QA
AjB,U;QACI,UAAU,UAAK,KAAL,C;QACV,UAAK,KAAL,IAAc,UAAK,YAAL,C;QACd,UAAK,YAAL,IAAqB
,G;QACrB,mC;;K;IAIR,8B;MAII,eAAe,CAAC,mBAAO,CAAP,IAAD,IAAa,CAAb,I;MACf,IAAI,WAAW,CAAf,
C;QAAkB,M;MACIB,mBAAmB,0B;MACnB,iBAAc,CAAd,WAAiB,QAAjB,U;QACI,UAAU,UAAK,KAAL,C;Q
ACV,UAAK,KAAL,IAAc,UAAK,YAAL,C;QACd,UAAK,YAAL,IAAqB,G;QACrB,mC;;K;IAIR,8B;MAII,eAAe,
CAAC,mBAAO,CAAP,IAAD,IAAa,CAAb,I;MACf,IAAI,WAAW,CAAf,C;QAAkB,M;MACIB,mBAAmB,0B;M
ACnB,iBAAc,CAAd,WAAiB,QAAjB,U;QACI,UAAU,UAAK,KAAL,C;QACV,UAAK,KAAL,IAAc,UAAK,YAA
L,C;QACd,UAAK,YAAL,IAAqB,G;QACrB,mC;;K;IAIR,8B;MAII,eAAe,CAAC,mBAAO,CAAP,IAAD,IAAa,CA
Ab,I;MACf,IAAI,WAAW,CAAf,C;QAAkB,M;MACIB,mBAAmB,0B;MACnB,iBAAc,CAAd,WAAiB,QAAjB,U;
QACI,UAAU,UAAK,KAAL,C;QACV,UAAK,KAAL,IAAc,UAAK,YAAL,C;QACd,UAAK,YAAL,IAAqB,G;QA
CrB,mC;;K;IAIR,8B;MAII,eAAe,CAAC,mBAAO,CAAP,IAAD,IAAa,CAAb,I;MACf,IAAI,WAAW,CAAf,C;QA
AkB,M;MACIB,mBAAmB,0B;MACnB,iBAAc,CAAd,WAAiB,QAAjB,U;QACI,UAAU,UAAK,KAAL,C;QACV,
UAAK,KAAL,IAAc,UAAK,YAAL,C;QACd,UAAK,YAAL,IAAqB,G;QACrB,mC;;K;IAIR,8B;MAII,eAAe,CAA
C,mBAAO,CAAP,IAAD,IAAa,CAAb,I;MACf,IAAI,WAAW,CAAf,C;QAAkB,M;MACIB,mBAAmB,0B;MACnB,
iBAAc,CAAd,WAAiB,QAAjB,U;QACI,UAAU,UAAK,KAAL,C;QACV,UAAK,KAAL,IAAc,UAAK,YAAL,C;Q
ACd,UAAK,YAAL,IAAqB,G;QACrB,mC;;K;IAIR,kD;MAWI,oCAAA,2BAAkB,SAaIB,EAA6B,OAA7B,EAAsC,
gBAAtC,C;MACb,eAAe,CAAC,YAAY,OAAZ,IAAD,IAAwB,CAAxB,I;MACf,IAAI,cAAa,QAAjB,C;QAA2B,M;
MAC3B,mBAAmB,UAAU,CAAV,I;MACnB,iBAAc,SAAd,UAA8B,QAA9B,U;QACI,UAAU,UAAK,KAAL,C;Q
ACV,UAAK,KAAL,IAAc,UAAK,YAAL,C;QACd,UAAK,YAAL,IAAqB,G;QACrB,mC;;K;IAIR,kD;MAWI,oCA
Aa,2BAAkB,SAaIB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACb,eAAe,CAAC,YAAY,OAAZ,IAAD,IAAwB,CA
AxB,I;MACf,IAAI,cAAa,QAAjB,C;QAA2B,M;MAC3B,mBAAmB,UAAU,CAAV,I;MACnB,iBAAc,SAAd,UAA
8B,QAA9B,U;QACI,UAAU,UAAK,KAAL,C;QACV,UAAK,KAAL,IAAc,UAAK,YAAL,C;QACd,UAAK,YAAL,
IAAqB,G;QACrB,mC;;K;IAIR,mD;MAWI,oCAAA,2BAAkB,SAaIB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACb,
eAAe,CAAC,YAAY,OAAZ,IAAD,IAAwB,CAAxB,I;MACf,IAAI,cAAa,QAAjB,C;QAA2B,M;MAC3B,mBAAm
B,UAAU,CAAV,I;MACnB,iBAAc,SAAd,UAA8B,QAA9B,U;QACI,UAAU,UAAK,KAAL,C;QACV,UAAK,KAA
L,IAAc,UAAK,YAAL,C;QACd,UAAK,YAAL,IAAqB,G;QACrB,mC;;K;IAIR,mD;MAWI,oCAAA,2BAAkB,SAa
IB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACb,eAAe,CAAC,YAAY,OAAZ,IAAD,IAAwB,CAAxB,I;MACf,IAAI,
cAAa,QAAjB,C;QAA2B,M;MAC3B,mBAAmB,UAAU,CAAV,I;MACnB,iBAAc,SAAd,UAA8B,QAA9B,U;QAC
I,UAAU,UAAK,KAAL,C;QACV,UAAK,KAAL,IAAc,UAAK,YAAL,C;QACd,UAAK,YAAL,IAAqB,G;QACrB,
mC;;K;IAIR,mD;MAWI,oCAAA,2BAAkB,SAaIB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACb,eAAe,CAAC,YAA
Y,OAAZ,IAAD,IAAwB,CAAxB,I;MACf,IAAI,cAAa,QAAjB,C;QAA2B,M;MAC3B,mBAAmB,UAAU,CAAV,I;
MACnB,iBAAc,SAAd,UAA8B,QAA9B,U;QACI,UAAU,UAAK,KAAL,C;QACV,UAAK,KAAL,IAAc,UAAK,Y
AAL,C;QACd,UAAK,YAAL,IAAqB,G;QACrB,mC;;K;IAIR,mD;MAWI,oCAAA,2BAAkB,SAaIB,EAA6B,OAA7
B,EAAsC,gBAAtC,C;MACb,eAAe,CAAC,YAAY,OAAZ,IAAD,IAAwB,CAAxB,I;MACf,IAAI,cAAa,QAAjB,C;
QAA2B,M;MAC3B,mBAAmB,UAAU,CAAV,I;MACnB,iBAAc,SAAd,UAA8B,QAA9B,U;QACI,UAAU,UAAK,

KAAL,C;QACV,UAAK,KAAL,IAAc,UAAK,YAAL,C;QACd,UAAK,YAAL,IAAqB,G;QACrB,mC;;K;IAIR,mD; MAWI,oCAAA,2BAAkB,SAAlB,EAA6B,OAA7B,EAA5C,gBAAtC,C;MACb,eAAe,CAAC,YAAY,OAAZ,IAAD,IAAwB,CAAxB,I;MACf,IAAI,cAAa,QAAjB,C;QAA2B,M;MAC3B,mBAAmB,UAAU,CAAV,I;MACnB,iBAAc,SAAd,UAA8B,QAA9B,U;QACI,UAAU,UAAK,KAAL,C;QACV,UAAK,KAAL,IAAc,UAAK,YAAL,C;QACd,UAAK,YAAL,IAAqB,G;QACrB,mC;;K;IAIR,mD;MAWI,oCAAA,2BAAkB,SAAlB,EAA6B,OAA7B,EAA5C,gBAAtC,C;MACb,eAAe,CAAC,YAAY,OAAZ,IAAD,IAAwB,CAAxB,I;MACf,IAAI,cAAa,QAAjB,C;QAA2B,M;MAC3B,mBAAmB,UAAU,CAAV,I;MACnB,iBAAc,SAAd,UAA8B,QAA9B,U;QACI,UAAU,UAAK,KAAL,C;QACV,UAAK,KAAL,IAAc,UAAK,YAAL,C;QACd,UAAK,YAAL,IAAqB,G;QACrB,mC;;K;IAIR,mD;MAWI,oCAAA,2BAAkB,SAAlB,EAA6B,OAA7B,EAA5C,gBAAtC,C;MACb,eAAe,CAAC,YAAY,OAAZ,IAAD,IAAwB,CAAxB,I;MACf,IAAI,cAAa,QAAjB,C;QAA2B,M;MAC3B,mBAAmB,UAAU,CAAV,I;MACnB,iBAAc,SAAd,UAA8B,QAA9B,U;QACI,UAAU,UAAK,KAAL,C;QACV,UAAK,KAAL,IAAc,UAAK,YAAL,C;QACd,UAAK,YAAL,IAAqB,G;QACrB,mC;;K;IAIR,6B;MAII,IA+nEO,qBAAQ,CA/nEf,C;QAAe,OAAO,W;MACtB,WAAW,wB;MACN,WAAL,IAAK,C;MACL,OAAO,I;K;IAGX,+B;MAII,IA6nEO,qBAAQ,CA7nEf,C;QAAe,OAAO,W;MACtB,WAAW,0B;MACN,WAAL,IAAK,C;MACL,OAAO,I;K;IAGX,+B;MAII,IA2nEO,qBAAQ,CA3nEf,C;QAAe,OAAO,W;MACtB,WAAW,0B;MACN,WAAL,IAAK,C;MACL,OAAO,I;K;IAGX,+B;MAII,IAynEO,qBAAQ,CAznEf,C;QAAe,OAAO,W;MACtB,WAAW,0B;MACN,WAAL,IAAK,C;MACL,OAAO,I;K;IAGX,+B;MAII,IAunEO,qBAAQ,CAvnEf,C;QAAe,OAAO,W;MACtB,WAAW,0B;MACN,WAAL,IAAK,C;MACL,OAAO,I;K;IAGX,+B;MAII,IAqnEO,qBAAQ,CArnEf,C;QAAe,OAAO,W;MACtB,WAAW,0B;MACN,WAAL,IAAK,C;MACL,OAAO,I;K;IAGX,+B;MAII,IAmnEO,qBAAQ,CAnnEf,C;QAAe,OAAO,W;MACtB,WAAW,0B;MACN,WAAL,IAAK,C;MACL,OAAO,I;K;IAGX,+B;MAII,IAinEO,qBAAQ,CAjnEf,C;QAAe,OAAO,W;MACtB,WAAW,0B;MACN,WAAL,IAAK,C;MACL,OAAO,I;K;IAGX,+B;MAII,IA+mEO,qBAAQ,CA/mEf,C;QAAe,OAAO,W;MACtB,WAAW,0B;MACN,WAAL,IAAK,C;MACL,OAAO,I;K;IAGX,kC;MAII,IAqiEO,qBAAQ,CAriEf,C;QAAe,OAAO,S;MACtB,aAAa,aAAa,SAAb,EAAmB,gBAAnB,C;MACb,gBAAgB,wB;MACHb,aAAU,CAAV,OAAa,SAAb,M;QACI,OAAO,YAAY,CAAZ,IAAP,IAAwB,UAAK,CAAL,C;MAC5B,OAAO,M;K;IAGX,oC;MAII,IAiiEO,qBAAQ,CAjiEf,C;QAAe,OAAO,S;MACtB,aAAa,cAAU,gBAAV,C;MACb,gBAAgB,0B;MACHb,aAAU,CAAV,OAAa,SAAb,M;QACI,OAAO,YAAY,CAAZ,IAAP,IAAwB,UAAK,CAAL,C;MAC5B,OAAO,M;K;IAGX,oC;MAII,IA6hEO,qBAAQ,CA7hEf,C;QAAe,OAAO,S;MACtB,aAAa,eAAW,gBAAX,C;MACb,gBAAgB,0B;MACHb,aAAU,CAAV,OAAa,SAAb,M;QACI,OAAO,YAAY,CAAZ,IAAP,IAAwB,UAAK,CAAL,C;MAC5B,OAAO,M;K;IAGX,oC;MAII,IAyhEO,qBAAQ,CAzhEf,C;QAAe,OAAO,S;MACtB,aAAa,eAAS,gBAAT,C;MACb,gBAAgB,0B;MACHb,aAAU,CAAV,OAAa,SAAb,M;QACI,OAAO,YAAY,CAAZ,IAAP,IAAwB,UAAK,CAAL,C;MAC5B,OAAO,M;K;IAGX,oC;MAII,IAqhEO,qBAAQ,CArhEf,C;QAAe,OAAO,S;MACtB,aAAa,iBAAU,gBAAV,C;MACb,gBAAgB,0B;MACHb,aAAU,CAAV,OAAa,SAAb,M;QACI,OAAO,YAAY,CAAZ,IAAP,IAAwB,UAAK,CAAL,C;MAC5B,OAAO,M;K;IAGX,oC;MAII,IAihEO,qBAAQ,CAjhEf,C;QAAe,OAAO,S;MACtB,aAAa,iBAAW,gBAAX,C;MACb,gBAAgB,0B;MACHb,aAAU,CAAV,OAAa,SAAb,M;QACI,OAAO,YAAY,CAAZ,IAAP,IAAwB,UAAK,CAAL,C;MAC5B,OAAO,M;K;IAGX,oC;MAII,IA6gEO,qBAAQ,CA7gEf,C;QAAe,OAAO,S;MACtB,aAAa,iBAAU,gBAAV,C;MACb,gBAAgB,0B;MACHb,aAAU,CAAV,OAAa,SAAb,M;QACI,OAAO,YAAY,CAAZ,IAAP,IAAwB,UAAK,CAAL,C;MAC5B,OAAO,M;K;IAGX,oC;MAII,IAygEO,qBAAQ,CAzgEf,C;QAAe,OAAO,S;MACtB,aAAa,oBAAa,gBAAb,C;MACb,gBAAgB,0B;MACHb,aAAU,CAAV,OAAa,SAAb,M;QACI,OAAO,YAAY,CAAZ,IAAP,IAAwB,UAAK,CAAL,C;MAC5B,OAAO,M;K;IAGX,oC;MAII,IAqgEO,qBAAQ,CArgEf,C;QAAe,OAAO,S;MACtB,aAAa,iBAAU,gBAAV,C;MACb,gBAAgB,0B;MACHb,aAAU,CAAV,OAAa,SAAb,M;QACI,OAAO,YAAY,CAAZ,IAAP,IAAwB,UAAK,CAAL,C;MAC5B,OAAO,M;K;IAGX,4B;MAKI,qBAAQ,4BAAR,C;K;IAGJ,8B;MAKI,qBAAQ,4BAAR,C;K;IAGJ,8B;MAKI,sBAAQ,4BAAR,C;K;IAGJ,8B;MAKI,sBAAQ,4BAAR,C;K;IAGJ,8B;MAKI,sBAAQ,4BAAR,C;K;IAGJ,8B;MAKI,sBAAQ,4BAAR,C;K;IAGJ,8B;MAKI,sBAAQ,4BAAR,C;K;IAGJ,8B;MAKI,sBAAQ,4BAAR,C;K;IAGJ,8B;MAKI,sBAAQ,4BAAR,C;K;IAGJ,8B;MAKI,sBAAQ,4BAAR,C;K;IAGJ,sC;MAOI,aAAU,wBAAV,OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAAR,C;QACf,WAAW,UAAK,CAAL,C;QACX,UAAK,CAAL,IAAU,UAAK,CAAL,C;QACV,UAAK,CAAL,IAAU,I;;K;IAIIB,sC;MAOI,aAAU,0BAAV,OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAA

I,CAAJ,IAAR,C;QACf,WAAW,UAAK,CAAL,C;QACX,UAAK,CAAL,IAAU,UAAK,CAAL,C;QACV,UAAK,C
AAL,IAAU,I;;K;IAIIB,uC;MAOI,aAAU,0BAAV,OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,I
AAR,C;QACf,WAAW,UAAK,CAAL,C;QACX,UAAK,CAAL,IAAU,UAAK,CAAL,C;QACV,UAAK,CAAL,IAA
U,I;;K;IAIIB,uC;MAOI,aAAU,0BAAV,OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAAR,C;Q
ACf,WAAW,UAAK,CAAL,C;QACX,UAAK,CAAL,IAAU,UAAK,CAAL,C;QACV,UAAK,CAAL,IAAU,I;;K;IA
IIB,uC;MAOI,aAAU,0BAAV,OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAAR,C;QACf,WA
AW,UAAK,CAAL,C;QACX,UAAK,CAAL,IAAU,UAAK,CAAL,C;QACV,UAAK,CAAL,IAAU,I;;K;IAIIB,uC;M
AOI,aAAU,0BAAV,OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAAR,C;QACf,WAAW,UAA
K,CAAL,C;QACX,UAAK,CAAL,IAAU,UAAK,CAAL,C;QACV,UAAK,CAAL,IAAU,I;;K;IAIIB,uC;MAOI,aAA
U,0BAAV,OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAAR,C;QACf,WAAW,UAAK,CAAL,
C;QACX,UAAK,CAAL,IAAU,UAAK,CAAL,C;QACV,UAAK,CAAL,IAAU,I;;K;IAIIB,uC;MAOI,aAAU,0BAA
V,OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAAR,C;QACf,WAAW,UAAK,CAAL,C;QACX
,UAAK,CAAL,IAAU,UAAK,CAAL,C;QACV,UAAK,CAAL,IAAU,I;;K;IAIIB,yB;MAAA,oD;MgBn5LA,sC;M
AAA,oC;MAAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA5Dd,cAAc,SA4DgB,
CA5DhB,CAAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MhB44Lf,sC;QAMI,IAAI,mBAAO,CAAX,C
;UAAc,oBgB15Ld,eAAW,iBhBk5LsB,QgB15LtB,CAAX,ChBk5Lc,C;;O;KANIB,C;sGASA,yB;MAAA,oD;MgBz4
LA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA/Ed,cAAc,S
A+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN,CAA3B,C;W;S;OA+EI,C;MhBk4Lf,sC;QAMI,IAAI,mBAAO,C
AAX,C;UAAc,oBgBx4Ld,eAAW,2BhBw4LgC,QgBx4LhC,CAAX,ChBw4Lc,C;;O;KANIB,C;IASA,mC;MAMI,o
BAAS,cAAT,C;K;IAGJ,qC;MAII,IAAI,mBAAO,CAAX,C;QACI,e;QACA,oB;;K;IAIR,qC;MAII,IAAI,mBAAO,C
AAX,C;QACI,e;QACA,oB;;K;IAIR,qC;MAII,IAAI,mBAAO,CAAX,C;QACI,e;QACA,oB;;K;IAIR,qC;MAII,IAAI
,mBAAO,CAAX,C;QACI,iB;QACA,oB;;K;IAIR,qC;MAII,IAAI,mBAAO,CAAX,C;QACI,e;QACA,oB;;K;IAIR,q
C;MAII,IAAI,mBAAO,CAAX,C;QACI,e;QACA,oB;;K;IAIR,qC;MAII,IAAI,mBAAO,CAAX,C;QACI,e;QACA,o
B;;K;IAIR,2B;MAMI,OAAqB,OAAAd,sBAAC,C;K;IAGzB,6B;MAI0B,kBAAf,yB;MAAuB,mB;MAA9B,OAAuC,
OiB5gMhC,WjB4gMgC,C;K;IAG3C,6B;MAI0B,kBAAf,yB;MAAuB,mB;MAA9B,OAAuC,OiBnhMhC,WjBmhM
gC,C;K;IAG3C,6B;MAI0B,kBAAf,yB;MAAuB,mB;MAA9B,OAAuC,OiB1hMhC,WjB0hMgC,C;K;IAG3C,6B;M
AI0B,kBAAf,yB;MAAuB,mB;MAA9B,OAAuC,OiBjiMhC,WjBiiMgC,C;K;IAG3C,6B;MAI0B,kBAAf,yB;MAAu
B,mB;MAA9B,OAAuC,OiBxiMhC,WjBwiMgC,C;K;IAG3C,6B;MAI0B,kBAAf,yB;MAAuB,mB;MAA9B,OAAu
C,OiB/iMhC,WjB+iMgC,C;K;IAG3C,6B;MAI0B,kBAAf,0B;MAAuB,mB;MAA9B,OAAuC,OiBtjMhC,WjBsjMg
C,C;K;IAG3C,gC;MAMI,IA6kDO,qBAAQ,CA7kDf,C;QA Ae,OAAO,S;MACD,kBAAd,SejKiB,Q;MfojKK,mB;
MAA7B,OiBhkMO,W;K;IjBmkMX,kC;MAII,IA6kDO,qBAAQ,CA7kDf,C;QA Ae,OAAO,S;MACD,kBAAd,SejK
iB,Q;MfjkKK,iB;MAA7B,OiBxkMO,W;K;IjB2kMX,kC;MAII,IA6kDO,qBAAQ,CA7kDf,C;QA Ae,OAAO,S;MA
CD,kBAAd,SejKiB,Q;MfgjKK,iB;MAA7B,OiBhlMO,W;K;IjBmlMX,kC;MAII,IA6kDO,qBAAQ,CA7kDf,C;QA
Ae,OAAO,S;MACD,kBAAd,Se9iKiB,Q;Mf8iKK,iB;MAA7B,OiBxIMO,W;K;IjB2IMX,kC;MAII,IA6kDO,qBAA
Q,CA7kDf,C;QA Ae,OAAO,S;MACD,kBAAT,UAAAL,SAAK,C;MAAiB,mB;MAA7B,OiBhmMO,W;K;IjBmmMX
,kC;MAII,IA6kDO,qBAAQ,CA7kDf,C;QA Ae,OAAO,S;MACD,kBAAd,Se3iKiB,Q;Mf2iKK,iB;MAA7B,OiBxm
MO,W;K;IjB2mMX,kC;MAII,IA6kDO,qBAAQ,CA7kDf,C;QA Ae,OAAO,S;MACD,kBAAd,SeziKiB,Q;MfyiKK,i
B;MAA7B,OiBhnMO,W;K;IjBmnMX,kC;MAII,IAqlDO,qBAAQ,CArIdf,C;QA Ae,OAAO,S;MACD,kBAAT,UAA
AL,SAAK,C;MAAiB,iB;MAA7B,OiBxnMO,W;K;IjB2nMX,0C;MAMI,IA2gDO,qBAAQ,CA3gDf,C;QA Ae,OAA
O,S;MACD,kBAAd,SetnKiB,Q;MfsnKK,sBAAS,cAAT,C;MAA7B,OiBloMO,W;K;IjBqoMX,4C;MAII,IA2gDO,q
BAAQ,CA3gDf,C;QA Ae,OAAO,S;MACD,kBAAd,SejnKiB,Q;MfonKK,6B;MAA7B,OiBoMO,W;K;IjB6oMX,4
C;MAII,IA2gDO,qBAAQ,CA3gDf,C;QA Ae,OAAO,S;MACD,kBAAd,SelnKiB,Q;MfknKK,6B;MAA7B,OiBlpMO
,W;K;IjBqpMX,4C;MAII,IA2gDO,qBAAQ,CA3gDf,C;QA Ae,OAAO,S;MACD,kBAAd,SehnKiB,Q;MfgnKK,6B;
MAA7B,OiBlpMO,W;K;IjB6pMX,4C;MAII,IA2gDO,qBAAQ,CA3gDf,C;QA Ae,OAAO,S;MACD,kBAAT,UAA
L,SAAK,C;MAAiB,6B;MAA7B,OiBlqMO,W;K;IjBqqMX,4C;MAII,IA2gDO,qBAAQ,CA3gDf,C;QA Ae,OAAO,S
;MACD,kBAAd,Se7mKiB,Q;Mf6mKK,6B;MAA7B,OiBlqMO,W;K;IjB6qMX,4C;MAII,IA2gDO,qBAAQ,CA3gD
f,C;QA Ae,OAAO,S;MACD,kBAAd,Se3mKiB,Q;Mf2mKK,6B;MAA7B,OiBlrMO,W;K;IjBqrMX,4C;MAII,IAmh
DO,qBAAQ,CAhDf,C;QA Ae,OAAO,S;MACD,kBAAT,UAAAL,SAAK,C;MAAiB,6B;MAA7B,OiBlrMO,W;K;Ij

B6rMX,gD;MAMI,IAy8CO,qBAAQ,CAz8Cf,C;QAAe,OAAO,S;MACD,kBAAd,SexrKiB,Q;MfwrKK,iC;MAA7B ,OiBpsMO,W;K;sFjBusMX,yB;MAAA,wD;MgB5rMA,sC;MAAA,oC;MAAA,uBAOe,yB;QArEf,8D;eAqEe,4B;U AAA,uB;YAAU,eAAsB,gB;YAAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EAA2B,SA4DM,CA5DN,CAA3B,C ;W;S;OA4DI,C;MhBqrMf,sC;QAQI,OAAO,sBgB7rMP,eAAW,iBhB6rMiB,QgB7rMjB,CAAX,ChB6rMO,C;O;KA RX,C;wFAWA,yB;MAAA,wD;MgBvsMA,sC;MAAA,oC;MAAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;Y AAU,eAAsB,gB;YAAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4 DI,C;MhBgsMf,sC;QAMI,OAAO,sBgBtsMP,eAAW,iBhBssMiB,QgBtsMjB,CAAX,ChBssMO,C;O;KANX,C;wF ASA,yB;MAAA,wD;MgBhtMA,sC;MAAA,oC;MAAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAs B,gB;YAAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MhBy sMf,sC;QAMI,OAAO,sBgB/sMP,eAAW,iBhB+sMiB,QgB/sMjB,CAAX,ChB+sMO,C;O;KANX,C;wFASA,yB;M AAA,wD;MgBztMA,sC;MAAA,oC;MAAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAAt B,OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MhBktMf,sC;QA MI,OAAO,sBgBxtMP,eAAW,iBhBwtMiB,QgBxtMjB,CAAX,ChBwtMO,C;O;KANX,C;wFASA,yB;MAAA,wD; MgBluMA,sC;MAAA,oC;MAAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAAtB,OA5Dd ,cAAc,SA4DgB,CA5DhB,CAAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MhB2tMf,sC;QAMI,OAAO, sBgBjuMP,eAAW,iBhBiuMiB,QgBjuMjB,CAAX,ChBiuMO,C;O;KANX,C;wFASA,yB;MAAA,wD;MgB3uMA,s C;MAAA,oC;MAAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAAtB,OA5Dd,cAAc,SA4 DgB,CA5DhB,CAAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MhBouMf,sC;QAMI,OAAO,sBgB1uM P,eAAW,iBhB0uMiB,QgB1uMjB,CAAX,ChB0uMO,C;O;KANX,C;wFASA,yB;MAAA,wD;MgBpvMA,sC;MAA A,oC;MAAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAAtB,OA5Dd,cAAc,SA4DgB,CA 5DhB,CAAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MhB6uMf,sC;QAMI,OAAO,sBgBnvMP,eAAW ,iBhBmvMiB,QgBnvMjB,CAAX,ChBmvMO,C;O;KANX,C;wFASA,yB;MAAA,wD;MgB7vMA,sC;MAAA,oC;M AAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,C AAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MhBsvMf,sC;QAMI,OAAO,sBgB5vMP,eAAW,iBhB4v MiB,QgB5vMjB,CAAX,ChB4vMO,C;O;KANX,C;wFASA,yB;MAAA,wD;MgBtwMA,sC;MAAA,oC;MAAA,uB AOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EA A2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MhB+vMf,sC;QAMI,OAAO,sBgBrwMP,eAAW,iBhBqwMiB,Qg BrwMjB,CAAX,ChBqwMO,C;O;KANX,C;0GASA,yB;MAAA,wD;MgB5vMA,sC;MAAA,oC;MAAA,iCAOe,yB; QAxFf,8D;eAwFe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+ EM,CA/EN,CAA3B,C;W;S;OA+EI,C;MhBqvMf,sC;QAMI,OAAO,sBgB3vMP,eAAW,2BhB2vM2B,QgB3vM3B, CAAX,ChB2vMO,C;O;KANX,C;4GASA,yB;MAAA,wD;MgBrwMA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8 D;eAwFe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/E N,CAA3B,C;W;S;OA+EI,C;MhB8vMf,sC;QAI,OAAO,sBgBlwMP,eAAW,2BhBkwM2B,QgBlwM3B,CAAX,Ch BkwMO,C;O;KAJX,C;4GAOA,yB;MAAA,wD;MgB5wMA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe, 4B;UAAA,uB;YAAU,eAAsB,gB;YAAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN,CAA3 B,C;W;S;OA+EI,C;MhBqwMf,sC;QAI,OAAO,sBgBzwMP,eAAW,2BhBywM2B,QgBzwM3B,CAAX,ChBywMO ,C;O;KAJX,C;4GAOA,yB;MAAA,wD;MgBnxMA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe,4B;UAA A,uB;YAAU,eAAsB,gB;YAAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN,CAA3B,C;W;S; OA+EI,C;MhB4wMf,sC;QAI,OAAO,sBgBhxMP,eAAW,2BhBgxM2B,QgBhxM3B,CAAX,ChBgxMO,C;O;KAJ X,C;4GAOA,yB;MAAA,wD;MgB1xMA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe,4B;UAAA,uB;YAA U,eAAsB,gB;YAAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN,CAA3B,C;W;S;OA+EI,C; MhBmxMf,sC;QAI,OAAO,sBgBvxMP,eAAW,2BhBuxM2B,QgBvxM3B,CAAX,ChBuxMO,C;O;KAJX,C;4GAO A,yB;MAAA,wD;MgBjyMA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe,4B;UAAA,uB;YAAU,eAAsB,g B;YAAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN,CAA3B,C;W;S;OA+EI,C;MhB0xMf,s C;QAI,OAAO,sBgB9xMP,eAAW,2BhB8xM2B,QgB9xM3B,CAAX,ChB8xMO,C;O;KAJX,C;4GAOA,yB;MAA A,wD;MgBxyMA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAAtB, OA/Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN,CAA3B,C;W;S;OA+EI,C;MhBiyMf,sC;QAI,OAA O,sBgBryMP,eAAW,2BhBqyM2B,QgBryM3B,CAAX,ChBqyMO,C;O;KAJX,C;4GAOA,yB;MAAA,wD;MgB/y

A,wB;QAAW,yBAAK,KAAL,C;O;K;IAJvC,iC;MAII,OAAO,gCAAW,gBAAX,GAAiB,8BAAjB,C;K;wFA2CX,y
B;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,uC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAAkB,EAAC,EAAd,C;Q
AC1B,kBAAY,mBAAoB,QAApB,C;QAYqBH,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WA1qB8C,SA0qB/B,CAA
U,OAAV,C;UOx+QnB,wBAAI,IAAK,MAAT,EAAGB,IAAK,OAARb,C;;QP8zPA,OA4qBO,W;O;KAXrBX,C;0FA
eA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,uC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAAkB,EAAC,EAAd,
C;QAC1B,kBAAY,mBAAoB,QAApB,C;QAYqBH,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WA1qB8C,SA0qB/B,C
AAU,OAAV,C;UOv/QnB,wBAAI,IAAK,MAAT,EAAGB,IAAK,OAARb,C;;QP60PA,OA4qBO,W;O;KAXrBX,C;0
FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,uC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAAkB,EAAC,EA
Ad,C;QAC1B,kBAAY,mBAAoB,QAApB,C;QAYqBH,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WA1qB8C,SA0qB/
B,CAAU,OAAV,C;UOtgRnB,wBAAI,IAAK,MAAT,EAAGB,IAAK,OAARb,C;;QP41PA,OA4qBO,W;O;KAXrBX,
C;0FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,uC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAAkB,EAAC,
EAAd,C;QAC1B,kBAAY,mBAAoB,QAApB,C;QAYqBH,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WA1qB8C,SA0
qB/B,CAAU,OAAV,C;UOrhRnB,wBAAI,IAAK,MAAT,EAAGB,IAAK,OAARb,C;;QP22PA,OA4qBO,W;O;KAXr
BX,C;0FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,uC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAAkB,E
AAc,EAAd,C;QAC1B,kBAAY,mBAAoB,QAApB,C;QAYqBH,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WA1qB8C
,SA0qB/B,CAAU,OAAV,C;UOpiRnB,wBAAI,IAAK,MAAT,EAAGB,IAAK,OAARb,C;;QP03PA,OA4qBO,W;O;
KAXrBX,C;0FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,uC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAA
kB,EAAC,EAAd,C;QAC1B,kBAAY,mBAAoB,QAApB,C;QAYqBH,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WA1q
B8C,SA0qB/B,CAAU,OAAV,C;UOnjRnB,wBAAI,IAAK,MAAT,EAAGB,IAAK,OAARb,C;;QPpy4PA,OA4qBO,W
;O;KAXrBX,C;0FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,uC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,C
AAkB,EAAC,EAAd,C;QAC1B,kBAAY,mBAAoB,QAApB,C;QAYqBH,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,W
A1qB8C,SA0qB/B,CAAU,OAAV,C;UOlkRnB,wBAAI,IAAK,MAAT,EAAGB,IAAK,OAARb,C;;QPw5PA,OA4qB
O,W;O;KAXrBX,C;0FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,uC;QAWI,eAAiC,cAAIB,YAAY,gBAA
Z,CAAkB,EAAC,EAAd,C;QAC1B,kBAAY,mBAAoB,QAApB,C;QAYqBH,Q;QAAhB,iD;UAAgB,cAAhB,e;UAC
I,WA1qB8C,SA0qB/B,CAAU,OAAV,C;UOjlRnB,wBAAI,IAAK,MAAT,EAAGB,IAAK,OAARb,C;;QPu6PA,OA4
qBO,W;O;KAXrBX,C;0FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,gC;MA5qBA,uC;QAW
I,eAAiC,cAAIB,YAAY,gBAAZ,CAAkB,EAAC,EAAd,C;QAC1B,kBAAY,mBAAoB,QAApB,C;QAYqBH,Q;QAA
hB,iD;UAAgB,cAAhB,0B;UACI,WA1qB8C,SA0qB/B,CAAU,oBAAV,C;UOhmRnB,wBAAI,IAAK,MAAT,EA
AGB,IAAK,OAARb,C;;QPs7PA,OA4qBO,W;O;KAXrBX,C;4FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,y
C;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAAkB,EAAC,EAAd,C;QAC1B,kBAAY,mBAAoB,QAApB,C;QAmQL,
Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WAAy,aApQoC,WaOqHc,CAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;QAp
QhB,OAsQO,W;O;KAIRX,C;8FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yC;QAWI,eAAiC,cAAIB,YA
AY,gBAAZ,CAAkB,EAAC,EAAd,C;QAC1B,kBAAY,mBAAuB,QAAvB,C;QAoQL,Q;QAAhB,iD;UAAgB,cAAh
B,e;UACI,WAAy,aArQuC,WaQqnC,CAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;QArQhB,OAuQO,W;O;KAnRX,
C;8FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAAkB,EAAC,
EAAd,C;QAC1B,kBAAY,mBAAwB,QAAxB,C;QAqQL,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WAAy,aAtQwC,
WAsQpC,CAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;QAtQhB,OAwQO,W;O;KApRX,C;8FAeA,yB;MAAA,0D;M
AAA,yD;MAAA,uE;MAAA,yC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAAkB,EAAC,EAAd,C;QAC1B,kBAAY,m
BAAbB,QAAtB,C;QAsQL,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WAAy,aAvQsC,WaUqIC,CAAY,OAAZ,CAA
J,EAA0B,OAA1B,C;;QAvQhB,OAyQO,W;O;KArRX,C;8FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yC
;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAAkB,EAAC,EAAd,C;QAC1B,kBAAY,mBAAuB,QAAvB,C;QAuQL,Q;
QAAhB,iD;UAAgB,cAAhB,e;UACI,WAAy,aAxQuC,WaWqnC,CAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;QAxQ
hB,OA0QO,W;O;KAtRX,C;8FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yC;QAWI,eAAiC,cAAIB,YAA
Y,gBAAZ,CAAkB,EAAC,EAAd,C;QAC1B,kBAAY,mBAAwB,QAAxB,C;QAwQL,Q;QAAhB,iD;UAAgB,cAAhB
,e;UACI,WAAy,aAzQwC,WaYqPc,CAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;QAZhB,OA2QO,W;O;KAvRX,C
;8FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAAkB,EAAC,E
AAd,C;QAC1B,kBAAY,mBAAyB,QAazB,C;QAYQL,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WAAy,aA1QyC,W
A0QrC,CAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;QA1QhB,OA4QO,W;O;KAXrX,C;8FAeA,yB;MAAA,0D;MAA

A,yD;MAAA,uE;MAAA,yC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAAkB,EAAC,EAAD,C;QAC1B,kBAAC,mBA
A0B,QAA1B,C;QA0QL,Q;QAaHb,iD;UAAgB,cAAhB,e;UACI,WAAy,aA3Q0C,WA2QtC,CAAY,OAAZ,CAAJ,
EAA0B,OAA1B,C;;QA3QhB,OA6QO,W;O;KAZRX,C;8FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MA6QA,oC
;MAAA,gC;MA7QA,yC;QAWI,eAAiC,cAAIB,YAAY,gBAAZ,CAAkB,EAAC,EAAD,C;QAC1B,kBAAC,mBAAu
B,QAAvB,C;QA2QL,Q;QAaHb,iD;UAAgB,cAAhB,0B;UACI,WAAy,aA5QuC,WA4QnC,CAAY,oBAAZ,CAAJ,
EAA0B,oBAA1B,C;;QA5QhB,OA8QO,W;O;KA1RX,C;8FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yD
;QAUI,eAAiC,cAAIB,YAAY,gBAAZ,CAAkB,EAAC,EAAD,C;QAC1B,kBAAC,mBAAoB,QAApB,C;QA6QL,Q;Q
AAhB,iD;UAAgB,cAAhB,e;UACI,WAAy,aA9QoC,WA8QhC,CAAY,OAAZ,CAAJ,EA9QiD,cA8QvB,CAAe,OA
Af,CAA1B,C;;QA9QhB,OAgRO,W;O;KA3RX,C;8FAcA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yD;QAUI
,eAAiC,cAAIB,YAAY,gBAAZ,CAAkB,EAAC,EAAD,C;QAC1B,kBAAC,mBAAoB,QAApB,C;QA+QL,Q;QAaHb
,iD;UAAgB,cAAhB,e;UACI,WAAy,aAhRoC,WAgRhC,CAAY,OAAZ,CAAJ,EAhRiD,cAgRvB,CAAe,OAAf,CA
A1B,C;;QAhRhB,OAKRO,W;O;KA7RX,C;+FAcA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yD;QAUI,eAAi
C,cAAIB,YAAY,gBAAZ,CAAkB,EAAC,EAAD,C;QAC1B,kBAAC,mBAAoB,QAApB,C;QAiRL,Q;QAaHb,iD;U
AAgB,cAAhB,e;UACI,WAAy,aAIrOC,WAKRhC,CAAY,OAAZ,CAAJ,EAIRiD,cAKRvB,CAAe,OAAf,CAA1B,C
;;QAIRhB,OAoRO,W;O;KA/RX,C;+FAcA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yD;QAUI,eAAiC,cAAI
B,YAAY,gBAAZ,CAAkB,EAAC,EAAD,C;QAC1B,kBAAC,mBAAoB,QAApB,C;QAmRL,Q;QAaHb,iD;UAAgB,
cAAhB,e;UACI,WAAy,aApRoC,WAoRhC,CAAY,OAAZ,CAAJ,EApRiD,cAoRvB,CAAe,OAAf,CAA1B,C;;QAp
RhB,OAsRO,W;O;KAjSX,C;+FAcA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yD;QAUI,eAAiC,cAAIB,YA
AY,gBAAZ,CAAkB,EAAC,EAAD,C;QAC1B,kBAAC,mBAAoB,QAApB,C;QAqRL,Q;QAaHb,iD;UAAgB,cAAh
B,e;UACI,WAAy,aAtRoC,WAsRhC,CAAY,OAAZ,CAAJ,EAtrID,cAsRvB,CAAe,OAAf,CAA1B,C;;QAtRhB,O
AwRO,W;O;KAnSX,C;+FAcA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yD;QAUI,eAAiC,cAAIB,YAAY,gB
AAZ,CAAkB,EAAC,EAAD,C;QAC1B,kBAAC,mBAAoB,QAApB,C;QAUrL,Q;QAaHb,iD;UAAgB,cAAhB,e;UA
CI,WAAy,aAxRoC,WAwRhC,CAAY,OAAZ,CAAJ,EAxRiD,cAwRvB,CAAe,OAAf,CAA1B,C;;QAxRhB,OA0R
O,W;O;KArSX,C;+FAcA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yD;QAUI,eAAiC,cAAIB,YAAY,gBAAZ,
CAAkB,EAAC,EAAD,C;QAC1B,kBAAC,mBAAoB,QAApB,C;QAYrL,Q;QAaHb,iD;UAAgB,cAAhB,e;UACI,W
AAy,aA1RoC,WA0RhC,CAAY,OAAZ,CAAJ,EA1RiD,cA0RvB,CAAe,OAAf,CAA1B,C;;QA1RhB,OA4RO,W;O
;KAvSX,C;+FAcA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yD;QAUI,eAAiC,cAAIB,YAAY,gBAAZ,CAAk
B,EAAC,EAAD,C;QAC1B,kBAAC,mBAAoB,QAApB,C;QA2RL,Q;QAaHb,iD;UAAgB,cAAhB,e;UACI,WAAy,a
A5RoC,WA4RhC,CAAY,OAAZ,CAAJ,EA5RiD,cA4RvB,CAAe,OAAf,CAA1B,C;;QA5RhB,OA8RO,W;O;KAZS
X,C;+FAcA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yD;QAUI,eAAiC,cAAIB,YAAY,gBAAZ,CAAk
B,EAAC,EAAD,C;QAC1B,kBAAC,mBAAoB,QAApB,C;QA6RL,Q;QAaHb,iD;UAAgB,cAAh
B,0B;UACI,WAAy,aA9RoC,WA8RhC,CAAY,oBAAZ,CAAJ,EA9RiD,cA8RvB,CAAe,oBAAf,CAA1B,C;;QA9R
hB,OAgSO,W;O;KA3SX,C;gGAcA,+C;MAUoB,Q;MAAhB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QA
CI,WAAy,aAAI,YAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;MAEhB,OAAO,W;K;kGAGX,+C;MAUoB,Q;MAAhB
,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,WAAy,aAAI,YAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;
MAEhB,OAAO,W;K;kGAGX,+C;MAUoB,Q;MAAhB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,W
AAy,aAAI,YAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;MAEhB,OAAO,W;K;iGAGX,+C;MAUoB,Q;MAAhB,wBA
AGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,WAAy,aAAI,YAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;MAEh
B,OAAO,W;K;kGAGX,+C;MAUoB,Q;MAAhB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,WAAy,a
AAI,YAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;MAEhB,OAAO,W;K;kGAGX,+C;MAUoB,Q;MAAhB,wBAAGB,S
AAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,WAAy,aAAI,YAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;MAEhB,OA
AO,W;K;kGAGX,+C;MAUoB,Q;MAAhB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,WAAy,aAAI,
YAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;MAEhB,OAAO,W;K;kGAGX,+C;MAUoB,Q;MAAhB,wBAAGB,SA
AhB,gB;QAAGB,cAAA,SAAhB,M;QACI,WAAy,aAAI,YAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;MAEhB,OAAO,
W;K;iGAGX,yB;MAAA,oC;MAAA,gC;MAAA,sD;QAUoB,Q;QAaHb,wBAAGB,SAAhB,gB;UAAgB,cAAhB,U
AAgB,SAAhB,O;UACI,WAAy,aAAI,YAAY,oBAAZ,CAAJ,EAA0B,oBAA1B,C;;QAEhB,OAAO,W;O;KAbX,C;
kGAgBA,+D;MAUoB,Q;MAAhB,wBAAGB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,WAAy,aAAI,YAAY,O
AAZ,CAAJ,EAA0B,eAAe,OAAf,CAA1B,C;;MAEhB,OAAO,W;K;kGAGX,+D;MAUoB,Q;MAAhB,wBAAGB,S

AAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,WAAy,aAAI,YAAy,OAAZ,CAAJ,EAA0B,eAAe,OAAf,CAA1B,C;;
MAEhB,OAAO,W;K;mGAGX,+D;MAUoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,W
AAy,aAAI,YAAy,OAAZ,CAAJ,EAA0B,eAAe,OAAf,CAA1B,C;;MAEhB,OAAO,W;K;mGAGX,+D;MAUoB,Q;
MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,WAAy,aAAI,YAAy,OAAZ,CAAJ,EAA0B,eAA
e,OAAf,CAA1B,C;;MAEhB,OAAO,W;K;mGAGX,+D;MAUoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,
SAAhB,M;QACI,WAAy,aAAI,YAAy,OAAZ,CAAJ,EAA0B,eAAe,OAAf,CAA1B,C;;MAEhB,OAAO,W;K;mGA
GX,+D;MAUoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,WAAy,aAAI,YAAy,OAAZ,
CAAJ,EAA0B,eAAe,OAAf,CAA1B,C;;MAEhB,OAAO,W;K;mGAGX,+D;MAUoB,Q;MAAhB,wBAAgB,SAAhB
,gB;QAAgB,cAAA,SAAhB,M;QACI,WAAy,aAAI,YAAy,OAAZ,CAAJ,EAA0B,eAAe,OAAf,CAA1B,C;;MAEh
B,OAAO,W;K;mGAGX,+D;MAUoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,WAAy,
aAAI,YAAy,OAAZ,CAAJ,EAA0B,eAAe,OAAf,CAA1B,C;;MAEhB,OAAO,W;K;mGAGX,yB;MAAA,oC;MAA
A,gC;MAAA,sE;QAUoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UACI,WAAy,aA
AI,YAAy,oBAAZ,CAAJ,EAA0B,eAAe,oBAAf,CAA1B,C;;QAEhB,OAAO,W;O;KAbX,C;2FAgBA,6C;MASoB,
Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,WAAe,UAAU,OAAV,C;QOx+QnB,wBAAL
IAAK,MAAT,EAAGB,IAAK,OAArB,C;;MP0+QA,OAAO,W;K;8FAGX,6C;MASoB,Q;MAAhB,wBAAgB,SAAhB
,gB;QAAgB,cAAA,SAAhB,M;QACI,WAAe,UAAU,OAAV,C;QOv/QnB,wBAAL,IAAK,MAAT,EAAGB,IAAK,O
AArB,C;;MPy/QA,OAAO,W;K;8FAGX,6C;MASoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;
QACI,WAAe,UAAU,OAAV,C;QOtgRnB,wBAAL,IAAK,MAAT,EAAGB,IAAK,OAArB,C;;MPwgRA,OAAO,W;
K;8FAGX,6C;MASoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,WAAe,UAAU,OAAV,
C;QOrhRnB,wBAAL,IAAK,MAAT,EAAGB,IAAK,OAArB,C;;MPuhRA,OAAO,W;K;8FAGX,6C;MASoB,Q;MA
AhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,WAAe,UAAU,OAAV,C;QOpiRnB,wBAAL,IAAK,M
AAT,EAAGB,IAAK,OAArB,C;;MPsiRA,OAAO,W;K;8FAGX,6C;MASoB,Q;MAAhB,wBAAgB,SAAhB,gB;QA
AgB,cAAA,SAAhB,M;QACI,WAAe,UAAU,OAAV,C;QOnjRnB,wBAAL,IAAK,MAAT,EAAGB,IAAK,OAArB,C
;;MPqjRA,OAAO,W;K;8FAGX,6C;MASoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,
WAAe,UAAU,OAAV,C;QOlkRnB,wBAAL,IAAK,MAAT,EAAGB,IAAK,OAArB,C;;MPokRA,OAAO,W;K;8FAG
X,6C;MASoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,WAAe,UAAU,OAAV,C;QOjlR
nB,wBAAL,IAAK,MAAT,EAAGB,IAAK,OAArB,C;;MPmlRA,OAAO,W;K;8FAGX,yB;MAAA,oC;MAAA,gC;M
AAA,oD;QASoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UACI,WAAe,UAAU,oBA
AV,C;UOhmRnB,wBAAL,IAAK,MAAT,EAAGB,IAAK,OAArB,C;;QPkmRA,OAAO,W;O;KAZX,C;gGAeA,yB;
MAAA,0D;MAAA,yD;MAAA,uE;MAAA,2C;QAYI,aAAa,mBAAAs,cAAIB,YAAy,gBAAZ,CAAkB,EAAC,EA
Ad,CAATC,C;QAsJG,Q;QAAhB,iD;UAAgB,cAAhB,e;UArJuB,MAsJP,aAAI,OAAJ,EATJe,aAsJF,CAAc,OAAd,CA
Ab,C;;QAtJhB,OAAuB,M;O;KAb3B,C;kGAgBA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,2C;QAaI,aAAa,m
BAAyC,cAAIB,YAAy,gBAAZ,CAAkB,EAAC,EAAd,CAAzC,C;QAsJG,Q;QAAhB,iD;UAAgB,cAAhB,e;UArJuB
,MAsJP,aAAI,OAAJ,EATJe,aAsJF,CAAc,OAAd,CAAb,C;;QAtJhB,OAAuB,M;O;KAd3B,C;kGaiBA,yB;MAAA,0
D;MAAA,yD;MAAA,uE;MAAA,2C;QAaI,aAAa,mBAA0C,cAAIB,YAAy,gBAAZ,CAAkB,EAAC,EAAd,CAA1C
,C;QAsJG,Q;QAAhB,iD;UAAgB,cAAhB,e;UArJuB,MAsJP,aAAI,OAAJ,EATJe,aAsJF,CAAc,OAAd,CAAb,C;;QA
tJhB,OAAuB,M;O;KAd3B,C;kGaiBA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,2C;QAaI,aAAa,mBAAwC,c
AAIB,YAAy,gBAAZ,CAAkB,EAAC,EAAd,CAAxC,C;QAsJG,Q;QAAhB,iD;UAAgB,cAAhB,e;UArJuB,MAsJP,a
AAI,OAAJ,EATJe,aAsJF,CAAc,OAAd,CAAb,C;;QAtJhB,OAAuB,M;O;KAd3B,C;kGaiBA,yB;MAAA,0D;MAA
A,yD;MAAA,uE;MAAA,2C;QAaI,aAAa,mBAAyC,cAAIB,YAAy,gBAAZ,CAAkB,EAAC,EAAd,CAAzC,C;QAsJ
G,Q;QAAhB,iD;UAAgB,cAAhB,e;UArJuB,MAsJP,aAAI,OAAJ,EATJe,aAsJF,CAAc,OAAd,CAAb,C;;QAtJhB,OA
AuB,M;O;KAd3B,C;kGaiBA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,2C;QAaI,aAAa,mBAA0C,cAAIB,Y
AAy,gBAAZ,CAAkB,EAAC,EAAd,CAA1C,C;QAsJG,Q;QAAhB,iD;UAAgB,cAAhB,e;UArJuB,MAsJP,aAAI,O
AAJ,EATJe,aAsJF,CAAc,OAAd,CAAb,C;;QAtJhB,OAAuB,M;O;KAd3B,C;kGaiBA,yB;MAAA,0D;MAAA,yD;M
AAA,uE;MAAA,2C;QAaI,aAAa,mBAA2C,cAAIB,YAAy,gBAAZ,CAAkB,EAAC,EAAd,CAA3C,C;QAsJG,Q;Q
AAhB,iD;UAAgB,cAAhB,e;UArJuB,MAsJP,aAAI,OAAJ,EATJe,aAsJF,CAAc,OAAd,CAAb,C;;QAtJhB,OAAuB,
M;O;KAd3B,C;kGaiBA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,2C;QAaI,aAAa,mBAA4C,cAAIB,YAAy,
gBAAZ,CAAkB,EAAC,EAAd,CAA5C,C;QAsJG,Q;QAAhB,iD;UAAgB,cAAhB,e;UArJuB,MAsJP,aAAI,OAAJ,E

AAK,C;UAHV,K;;MAAP,W;K;IAOJ,kC;MAII,OAAO,iBA Ae,aAAL,SAAK,CAAf,C;K;IAGX,oC;MAKiB,Q;MA Db,WAAW,iBAAgB,gBAAhB,C;MACX,wBA Aa,SAAb,gB;QAAa,WAAA,SAAb,M;QAAmB,IAAK,WAAI,IAAJ ,C;;MACxB,OAAO,I;K;IAGX,oC;MAKiB,Q;MADb,WAAW,iBA AiB,gBAAjB,C;MACX,wBA Aa,SAAb,gB;QAA a,WAAA,SAAb,M;QAAmB,IAAK,WAAI,IAAJ,C;;MACxB,OAAO,I;K;IAGX,oC;MAKiB,Q;MADb,WAAW,iBA Ae,gBA Af,C;MACX,wBA Aa,SAAb,gB;QAAa,WAAA,SAAb,M;QAAmB,IAAK,WAAI,IAAJ,C;;MACxB,OAAO, I;K;IAGX,oC;MAKiB,Q;MADb,WAAW,iBA AgB,gBAAhB,C;MACX,wBA Aa,SAAb,gB;QAAa,WAAA,SAAb,M ;QAAmB,IAAK,WAAI,IAAJ,C;;MACxB,OAAO,I;K;IAGX,oC;MAKiB,Q;MADb,WAAW,iBA AiB,gBA AjB,C;M ACX,wBA Aa,SAAb,gB;QAAa,WAAA,SAAb,M;QAAmB,IAAK,WAAI,IAAJ,C;;MACxB,OAAO,I;K;IAGX,oC; MAKiB,Q;MADb,WAAW,iBA AkB,gBA AiB,C;MACX,wBA Aa,SAAb,gB;QAAa,WAAA,SAAb,M;QAAmB,IAA K,WAAI,IAAJ,C;;MACxB,OAAO,I;K;IAGX,oC;MAKiB,Q;MADb,WAAW,iBA AmB,gBA AnB,C;MACX,wBAA a,SAAb,gB;QAAa,WAAA,SAAb,M;QAAmB,IAAK,WAAI,IAAJ,C;;MACxB,OAAO,I;K;IAGX,oC;MAKiB,Q;M ADb,WAAW,iBA AgB,gBA AhB,C;MACX,wBA Aa,SAAb,gB;QAAa,WAAb,UAAa,SAAb,O;QAAmB,IAAK,WA AI,iBA AJ,C;;MACxB,OAAO,I;K;IAGX,0B;MAMiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,iB;UAA L,K;aACA,C;UAAK,aAM,UAAK,CAAL,CAAN,C;UAAL,K;;UACQ,+BA Aa,qBA AiB,YAAY,gBA AZ,CAAjB, CAAb,C;UAHL,K;;MAAP,W;K;IAOJ,4B;MAMiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,iB;UAA L, K;aACA,C;UAAK,aAM,UAAK,CAAL,CAAN,C;UAAL,K;;UACQ,iCAAa,qBA AoB,YAAY,gBA AZ,CAApB,C AA b,C;UAHL,K;;MAAP,W;K;IAOJ,4B;MAMiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,iB;UAA L,K;a ACA,C;UAAK,aAM,UAAK,CAAL,CAAN,C;UAAL,K;;UACQ,iCAAa,qBA AqB,YAAY,gBA AZ,CAArB,CA Ab,C;UAHL,K;;MAAP,W;K;IAOJ,4B;MAMiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,iB;UAA L,K;a ACA,C;UAAK,aAM,UAAK,CAAL,CAAN,C;UAAL,K;;UACQ,iCAAa,qBA AmB,YAAY,gBA AZ,CAAnB,CAA b,C;UAHL,K;;MAAP,W;K;IAOJ,4B;MAMiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,iB;UAA L,K;a ACA,C;UAAK,aAM,UAAK,CAAL,CAAN,C;UAAL,K;;UACQ,iCAAa,qBA AoB,YAAY,gBA AZ,CAApB,CAAb, C;UAHL,K;;MAAP,W;K;IAOJ,4B;MAMiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,iB;UAA L,K;a AC A,C;UAAK,aAM,UAAK,CAAL,CAAN,C;UAAL,K;;UACQ,iCAAa,qBA AqB,YAAY,gBA AZ,CAArB,CAAb,C; UAHL,K;;MAAP,W;K;IAOJ,4B;MAMiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,iB;UAA L,K;aACA, C;UAAK,aAM,UAAK,CAAL,CAAN,C;UAAL,K;;UACQ,iCAAa,qBA AsB,YAAY,gBA AZ,CAAtB,CAAb,C;U AHL,K;;MAAP,W;K;IAOJ,4B;MAMiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,iB;UAA L,K;aACA,C ;UAAK,aAM,UAAK,CAAL,CAAN,C;UAAL,K;;UACQ,iCAAa,qBA AuB,YAAY,gBA AZ,CAAvB,CAAb,C;UA HL,K;;MAAP,W;K;IAOJ,4B;MAMiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,iB;UAA L,K;aACA,C; UAAK,aAM,sBA AK,CAAL,EAAN,C;UAAL,K;;UACQ,iCAAa,qBA AoB,YAAiB,eAAL,gBA AK,EAAa,GAAb, CAAjB,CAApB,CAAb,C;UAHL,K;;MAAP,W;K;oFAOJ,yB;MAAA,+D;MAwaA,gD;MAxaA,uC;QAMW,kBAA U,gB;QAwaD,Q;QA AhB,iD;UAAgB,cAAhB,e;UACI,WAv6B,SAuAlB,CAAU,OAAV,C;UACC,OAAZ,WAA Y,EAAO,IAAP,C;;QA xahB,OA0aO,W;O;KAhbX,C;sFASA,yB;MAAA,+D;MA0aA,gD;MA1aA,uC;QAMW,kBAAU, gB;QAwaD,Q;QA AhB,iD;UAAgB,cAAhB,e;UACI,WAZa6B,SAyalB,CAAU,OAAV,C;UACC,OAAZ,WAA Y,EA AO,IAAP,C;;QA1ahB,OA4aO,W;O;KA1bX,C;sFASA,yB;MAAA,+D;MA4aA,gD;MA5aA,uC;QAMW,kBAAU,g B;QA0aD,Q;QA AhB,iD;UAAgB,cAAhB,e;UACI,WA3a6B,SA2alB,CAAU,OAAV,C;UACC,OAAZ,WAA Y,EAA O,IAAP,C;;QA5ahB,OA8aO,W;O;KApbX,C;sFASA,yB;MAAA,+D;MA8aA,gD;MA9aA,uC;QAMW,kBAAU,gB; QA4aD,Q;QA AhB,iD;UAAgB,cAAhB,e;UACI,WA7a6B,SA6alB,CAAU,OAAV,C;UACC,OAAZ,WAA Y,EAAO, IAAP,C;;QA9ahB,OAgbO,W;O;KAtbX,C;sFASA,yB;MAAA,+D;MAGbA,gD;MAhbA,uC;QAMW,kBAAU,gB;Q A8aD,Q;QA AhB,iD;UAAgB,cAAhB,e;UACI,WA/a6B,SA+alB,CAAU,OAAV,C;UACC,OAAZ,WAA Y,EAAO,IA AP,C;;QA hbhB,OAkbO,W;O;KAxbX,C;sFASA,yB;MAAA,+D;MAkbA,gD;MA1bA,uC;QAMW,kBAAU,gB;QA g bD,Q;QA AhB,iD;UAAgB,cAAhB,e;UACI,WAjb6B,SAi1B,CAAU,OAAV,C;UACC,OAAZ,WAA Y,EAAO,IAAP ,C;;QA1bhB,OAobO,W;O;KA1bX,C;sFASA,yB;MAAA,+D;MAobA,gD;MApbA,uC;QAMW,kBAAU,gB;QAkbD, Q;QA AhB,iD;UAAgB,cAAhB,e;UACI,WAnb6B,SAm1B,CAAU,OAAV,C;UACC,OAAZ,WAA Y,EAAO,IAAP, C;;QApbhB,OAsbO,W;O;KA5bX,C;sFASA,yB;MAAA,+D;MAsbA,gD;MAtbA,uC;QAMW,kBAAU,gB;QAobD, Q;QA AhB,iD;UAAgB,cAAhB,e;UACI,WArb6B,SAqb1B,CAAU,OAAV,C;UACC,OAAZ,WAA Y,EAAO,IAAP,C; ;QAtbhB,OAwbO,W;O;KA9bX,C;sFASA,yB;MAAA,+D;MAwbA,oC;MAAA,gD;MAAA,gC;MAxbA,uC;QAMW ,kBAAU,gB;QAsbD,Q;QA AhB,iD;UAAgB,cAAhB,0B;UACI,WAvb6B,SAub1B,CAAU,oBAAV,C;UACC,OAAZ,

WAAY,EAAO,IAAP,C;;QAxhbB,OA0bO,W;O;KAhcX,C;sFASA,yB;MAAA,+D;MA0bA,gD;MA1bA,uC;QAUW ,kBAAU,gB;QAwbD,Q;QAaHb,iD;UAAgB,cAAhB,e;UACI,Wazb6B,SAyblB,CAAU,OAAV,C;UACC,OAAZ,W AAY,EAAO,IAAP,C;;QA1bhB,OA4bO,W;O;KAtcX,C;kGAaA,yB;MAAA,+D;MA5JA,gD;MATJA,uC;QAYW,kB AAIb,gB;QAqJR,gB;QADhb,YAAY,C;QACZ,iD;UAAgB,cAAhB,e;UACI,WatJoC,SAsJzB,EAAU,cAAV,EAA U,sBAAV,WAAmB,OAAAnB,C;UACC,OAAZ,WAAAY,EAAO,IAAP,C;;QAvJhB,OAYJO,W;O;KArKX,C;oGAeA, yB;MAAA,+D;MAyJA,gD;MAzJA,uC;QAYW,kBAAiB,gB;QAwJR,gB;QADhb,YAAY,C;QACZ,iD;UAAgB,cA AhB,e;UACI,WazJoC,SAyJzB,EAAU,cAAV,EAAU,sBAAV,WAAmB,OAAAnB,C;UACC,OAAZ,WAAAY,EAAO,I AAP,C;;QA1JhB,OA4JO,W;O;KAxKX,C;oGAeA,yB;MAAA,+D;MA4JA,gD;MA5JA,uC;QAYW,kBAAiB,gB;Q A2JR,gB;QADhb,YAAY,C;QACZ,iD;UAAgB,cAAhB,e;UACI,WA5JoC,SA4JzB,EAAU,cAAV,EAAU,sBAAV, WAAmB,OAAAnB,C;UACC,OAAZ,WAAAY,EAAO,IAAP,C;;QA7JhB,OA+JO,W;O;KA3KX,C;oGAeA,yB;MAAA ,+D;MA+JA,gD;MA/JA,uC;QAYW,kBAAiB,gB;QA8JR,gB;QADhb,YAAY,C;QACZ,iD;UAAgB,cAAhB,e;UAC I,WA/JoC,SA+JzB,EAAU,cAAV,EAAU,sBAAV,WAAmB,OAAAnB,C;UACC,OAAZ,WAAAY,EAAO,IAAP,C;;QA hKhB,OAKKO,W;O;KA9KX,C;oGAeA,yB;MAAA,+D;MAkKA,gD;MAIKa,uC;QAYW,kBAAiB,gB;QAIKR,gB; QADhb,YAAY,C;QACZ,iD;UAAgB,cAAhB,e;UACI,WAlKoC,SakKzB,EAAU,cAAV,EAAU,sBAAV,WAAmB, OAAAnB,C;UACC,OAAZ,WAAAY,EAAO,IAAP,C;;QAnKhB,OAqKO,W;O;KAjLX,C;oGAeA,yB;MAAA,+D;MAq KA,gD;MArKA,uC;QAYW,kBAAiB,gB;QAoKR,gB;QADhb,YAAY,C;QACZ,iD;UAAgB,cAAhB,e;UACI,WAr KoC,SAqKzB,EAAU,cAAV,EAAU,sBAAV,WAAmB,OAAAnB,C;UACC,OAAZ,WAAAY,EAAO,IAAP,C;;QAtKh B,OAwKO,W;O;KApLX,C;oGAeA,yB;MAAA,+D;MAwKA,gD;MAxKA,uC;QAYW,kBAAiB,gB;QauKR,gB;Q ADhb,YAAY,C;QACZ,iD;UAAgB,cAAhB,e;UACI,WaxKoC,SAwKzB,EAAU,cAAV,EAAU,sBAAV,WAAmB, OAAAnB,C;UACC,OAAZ,WAAAY,EAAO,IAAP,C;;QazKhB,OA2KO,W;O;KAvLX,C;oGAeA,yB;MAAA,+D;MA 2KA,gD;MA3KA,uC;QAYW,kBAAiB,gB;QA0KR,gB;QADhb,YAAY,C;QACZ,iD;UAAgB,cAAhB,e;UACI,WA 3KoC,SA2KzB,EAAU,cAAV,EAAU,sBAAV,WAAmB,OAAAnB,C;UACC,OAAZ,WAAAY,EAAO,IAAP,C;;QA5K hB,OA8KO,W;O;KA1LX,C;oGAeA,yB;MAAA,+D;MA8KA,oC;MAAA,gD;MAAA,gC;MA9KA,uC;QAYW,kBA AiB,gB;QA6KR,gB;QADhb,YAAY,C;QACZ,iD;UAAgB,cAAhB,0B;UACI,WA9KoC,SA8KzB,EAAU,cAAV,EA AU,sBAAV,WAAmB,oBAAnB,C;UACC,OAAZ,WAAAY,EAAO,IAAP,C;;QA/KhB,OaiLO,W;O;KA7LX,C;oGAe A,yB;MAAA,+D;MAiLA,gD;MAjLA,uC;QAYW,kBAAiB,gB;QAgLR,gB;QADhb,YAAY,C;QACZ,iD;UAAgB,c AAhB,e;UACI,WajLoC,SAiLzB,EAAU,cAAV,EAAU,sBAAV,WAAmB,OAAAnB,C;UACC,OAAZ,WAAAY,EAA O,IAAP,C;;QAILhB,OAoLO,W;O;KAhMX,C;sGAeA,yB;MAAA,gD;MAAA,oD;QAWoB,UACS,M;QAFzB,YAA Y,C;QACZ,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,WAAW,WAAU,cAAV,EAAU,sBAAV,WAA mB,OAAAnB,C;UACC,OAAZ,WAAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KafX,C;wGakBA,yB;MAAA,gD;M AAA,oD;QAWoB,UACS,M;QAFzB,YAAY,C;QACZ,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,WA AAW,WAAU,cAAV,EAAU,sBAAV,WAAmB,OAAAnB,C;UACC,OAAZ,WAAAY,EAAO,IAAP,C;;QAEhB,OAAO, W;O;KafX,C;wGakBA,yB;MAAA,gD;MAAA,oD;QAWoB,UACS,M;QAFzB,YAAY,C;QACZ,wBAAgB,SAAh B,gB;UAAgB,cAAA,SAAhB,M;UACI,WAAW,WAAU,cAAV,EAAU,sBAAV,WAAmB,OAAAnB,C;UACC,OAA Z,WAAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KafX,C;wGakBA,yB;MAAA,gD;MAAA,oD;QAWoB,UACS,M ;QAFzB,YAAY,C;QACZ,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,WAAW,WAAU,cAAV,EAAU,s BAAV,WAAmB,OAAAnB,C;UACC,OAAZ,WAAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KafX,C;wGakBA,yB; MAAA,gD;MAAA,oD;QAWoB,UACS,M;QAFzB,YAAY,C;QACZ,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB ,M;UACI,WAAW,WAAU,cAAV,EAAU,sBAAV,WAAmB,OAAAnB,C;UACC,OAAZ,WAAAY,EAAO,IAAP,C;;QA EhB,OAAO,W;O;KafX,C;wGakBA,yB;MAAA,gD;MAAA,oD;QAWoB,UACS,M;QAFzB,YAAY,C;QACZ,wB AAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,WAAW,WAAU,cAAV,EAAU,sBAAV,WAAmB,OAAAnB,C; UACC,OAAZ,WAAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KafX,C;wGakBA,yB;MAAA,gD;MAAA,oD;QAW oB,UACS,M;QAFzB,YAAY,C;QACZ,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,WAAW,WAAU,c AAV,EAAU,sBAAV,WAAmB,OAAAnB,C;UACC,OAAZ,WAAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KafX,C; wGakBA,yB;MAAA,gD;MAAA,oD;QAWoB,UACS,M;QAFzB,YAAY,C;QACZ,wBAAgB,SAAhB,gB;UAAgB, cAAA,SAAhB,M;UACI,WAAW,WAAU,cAAV,EAAU,sBAAV,WAAmB,OAAAnB,C;UACC,OAAZ,WAAAY,EAA O,IAAP,C;;QAEhB,OAAO,W;O;KafX,C;wGakBA,yB;MAAA,oC;MAAA,gD;MAAA,gC;MAAA,oD;QAWoB,U ACS,M;QAFzB,YAAY,C;QACZ,wBAAgB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UACI,WAAW,WAA

U,cAAV,EAAU,sBAAV,WAAmB,oBAAnB,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KAF X,C;wGAKBA,yB;MAAA,gD;MAAA,oD;QAWoB,UACS,M;QAFzB,YAAy,C;QACZ,wBAAgB,SAAhB,gB;UA AgB,cAAA,SAAhB,M;UACI,WAAW,WAAU,cAAV,EAAU,sBAAV,WAAmB,OAAnB,C;UACC,OAAZ,WAAy, EAAO,IAAP,C;;QAEhB,OAAO,W;O;KAFx,C;uFakBA,yB;MAAA,gD;MAAA,oD;QAIoB,Q;QAAhB,wBAAgB, SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;Q AEhB,OAAO,W;O;KARX,C;0FAWA,yB;MAAA,gD;MAAA,oD;QAIoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAg B,cAAA,SAAhB,M;UACI,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QAEhB,OAAO,W;O ;KARX,C;0FAWA,yB;MAAA,gD;MAAA,oD;QAIoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M ;UACI,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KARX,C;0FAWA, yB;MAAA,gD;MAAA,oD;QAIoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,WAAW,UA AU,OAAV,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KARX,C;0FAWA,yB;MAAA,gD;MA AA,oD;QAIoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,WAAW,UAAU,OAAV,C;UAC C,OAAZ,WAAy,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KARX,C;0FAWA,yB;MAAA,gD;MAAA,oD;QAIoB,Q;Q AAhB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAy,E AAO,IAAP,C;;QAEhB,OAAO,W;O;KARX,C;0FAWA,yB;MAAA,gD;MAAA,oD;QAIoB,Q;QAAhB,wBAAgB,S AAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QA EhB,OAAO,W;O;KARX,C;0FAWA,yB;MAAA,gD;MAAA,oD;QAIoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB, cAAA,SAAhB,M;UACI,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QAEhB,OAAO,W;O;K ARX,C;0FAWA,yB;MAAA,oC;MAAA,gD;MAAA,gC;MAAA,oD;QAIoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAA gB,cAAhB,UAAgB,SAAhB,O;UACI,WAAW,UAAU,oBAAV,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QAEhB, OAAO,W;O;KARX,C;0FAWA,yB;MAAA,gD;MAAA,oD;QAQoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cA AA,SAAhB,M;UACI,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KA ZX,C;oFAeA,yB;MAAA,wE;MAiOA,+D;MAjOA,yC;QASW,kBAAU,oB;QAIoD,Q;QAAhB,iD;UAAgB,cAAhB, e;UACI,UAIoID,WakOvC,CAAY,OAAZ,C;UOp5UP,U;UADP,YPs5Ue,Wot5UH,WPs5UwB,Got5UxB,C;UAC L,IAAI,aAAJ,C;YACH,aPo5UuC,gB;YAA5B,WOn5UX,aPm5UgC,GOn5UhC,EAAS,MAAT,C;YACA,e;;YAEA, c;;UPg5UA,iB;UACA,IAAK,WAAI,OAAJ,C;;QApOT,OAsOO,W;O;KA/OX,C;sFAYA,yB;MAAA,wE;MAsoA,+ D;MAtoA,yC;QASW,kBAAU,oB;QAsOD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,UAvOoD,WauO1C,CAAY,O AAZ,C;UOr6UP,U;UADP,YPu6Ue,Wov6UH,Wpu6UwB,GOv6UxB,C;UACL,IAAI,aAAJ,C;YACH,aPq6UuC,gB ;YAA5B,WOp6UX,aPo6UgC,GOp6UhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UPi6UA,iB;UACA,IAAK,WAAI,O AAJ,C;;QAzOT,OA2OO,W;O;KApPX,C;sFAYA,yB;MAAA,wE;MA2OA,+D;MA3OA,yC;QASW,kBAAU,oB;Q A2OD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,UA5OqD,WA4O3C,CAAY,OAAZ,C;UOt7UP,U;UADP,YPw7Ue, Wox7UH,WPw7UwB,GOx7UxB,C;UACL,IAAI,aAAJ,C;YACH,aPs7UuC,gB;YAA5B,WOr7UX,aPq7UgC,GOr7 UhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UPk7UA,iB;UACA,IAAK,WAAI,OAAJ,C;;QA9OT,OAgPO,W;O;KAz PX,C;sFAYA,yB;MAAA,wE;MAgPA,+D;MAhPA,yC;QASW,kBAAU,oB;QAgPD,Q;QAAhB,iD;UAAgB,cAAhB ,e;UACI,UAjPmD,WaiPzC,CAAY,OAAZ,C;UOv8UP,U;UADP,YPy8Ue,Woz8UH,Wpy8UwB,GoZ8UxB,C;UA CL,IAAI,aAAJ,C;YACH,aPu8UuC,gB;YAA5B,Wot8UX,aPs8UgC,Got8UhC,EAAS,MAAT,C;YACA,e;;YAEA,c ;;UPm8UA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAnPT,OAgPO,W;O;KA9PX,C;sFAYA,yB;MAAA,wE;MAqPA,+ D;MArPA,yC;QASW,kBAAU,oB;QAqPD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,UAtPoD,WAsP1C,CAAY,OA AZ,C;UOx9UP,U;UADP,YP09Ue,Wo19UH,WP09UwB,GO19UxB,C;UACL,IAAI,aAAJ,C;YACH,aPw9UuC,gB; YAA5B,Wov9UX,aPu9UgC,GOv9UhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UPo9UA,iB;UACA,IAAK,WAAI,O AAJ,C;;QAxPT,OA0PO,W;O;KAnQX,C;sFAYA,yB;MAAA,wE;MA0PA,+D;MA1PA,yC;QASW,kBAAU,oB;QA 0PD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,UA3PqD,WA2P3C,CAAY,OAAZ,C;UOz+UP,U;UADP,YP2+Ue,Wo 3+UH,WP2+UwB,GO3+UxB,C;UACL,IAAI,aAAJ,C;YACH,aPy+UuC,gB;YAA5B,Wox+UX,aPw+UgC,GOx+U hC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UPq+UA,iB;UACA,IAAK,WAAI,OAAJ,C;;QA7PT,OA+PO,W;O;KAxQ X,C;sFAYA,yB;MAAA,wE;MA+PA,+D;MA/PA,yC;QASW,kBAAU,oB;QA+PD,Q;QAAhB,iD;UAAgB,cAAhB,e ;UACI,UAhQsD,WAgQ5C,CAAY,OAAZ,C;UO1/UP,U;UADP,YP4/Ue,Wo5/UH,WP4/UwB,GO5/UxB,C;UACL, IAAI,aAAJ,C;YACH,aP0/UuC,gB;YAA5B,Woz/UX,aPy/UgC,GOz/UhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UP s/UA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAIQT,OAoQO,W;O;KA7QX,C;sFAYA,yB;MAAA,wE;MAoQA,+D;M

ApQA,yC;QASW,kBAAU,oB;QAoQD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,UArQuD,WAqQ7C,CAAY,OAAZ,C;UO3gVP,U;UADP,YP6gVe,WO7gVH,WP6gVwB,GO7gVxB,C;UACL,IAAI,aAAJ,C;YACH,aP2gVuC,gB;YA A5B,WO1gVX,aP0gVgC,GO1gVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UPugVA,iB;UACA,IAAK,WAAI,OAA J,C;;QAvQT,OAYQO,W;O;KAIRX,C;sFAYa,yB;MAAA,wE;MAyQA,oC;MAAA,+D;MAAA,gC;MAzQA,yC;QA SW,kBAAU,oB;QAYQD,Q;QAAhB,iD;UAAgB,cAAhB,0B;UACI,UA1QoD,WA0Q1C,CAAY,oBAAZ,C;UO5hV P,U;UADP,YP8hVe,WO9hVH,WP8hVwB,GO9hVxB,C;UACL,IAAI,aAAJ,C;YACH,aP4hVuC,gB;YAA5B,WO3 hVX,aP2hVgC,GO3hVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UPwhVA,iB;UACA,IAAK,WAAI,oBAAJ,C;;QA5 QT,OA8QO,W;O;KAvRX,C;sFAYa,yB;MAAA,wE;MA8QA,+D;MA9QA,yD;QAUW,kBAAU,oB;QA8QD,Q;Q AAhB,iD;UAAgB,cAAhB,e;UACI,UA/QiD,WA+QvC,CAAY,OAAZ,C;UO9iVP,U;UADP,YPgjVe,WOhjVH,WP gjVwB,GOhjVxB,C;UACL,IAAI,aAAJ,C;YACH,aP8iVuC,gB;YAA5B,WO7iVX,aP6iVgC,GO7iVhC,EAAS,MA AT,C;YACA,e;;YAEA,c;;UP0iVA,iB;UACA,IAAK,WajRyD,cAiRrD,CAAe,OAAf,CAAJ,C;;QAJRT,OAmRO,W ;O;KA7RX,C;sFAaA,yB;MAAA,wE;MAmRA,+D;MAnRA,yD;QAUW,kBAAU,oB;QAmRD,Q;QAAhB,iD;UAA gB,cAAhB,e;UACI,UAPriD,WAoRvC,CAAY,OAAZ,C;UOhkVP,U;UADP,YPkkVe,WolkVH,WPkkVwB,GOlk VxB,C;UACL,IAAI,aAAJ,C;YACH,aPgkVuC,gB;YAA5B,WO/jVX,aP+jVgC,GO/jVhC,EAAS,MAAT,C;YACA,e ;;YAEA,c;;UP4jVA,iB;UACA,IAAK,WAtRyD,cAsRrD,CAAe,OAAf,CAAJ,C;;QATRT,OAwRO,W;O;KAISX,C;u FaaA,yB;MAAA,wE;MAwRA,+D;MAxRA,yD;QAUW,kBAAU,oB;QAwRD,Q;QAAhB,iD;UAAgB,cAAhB,e;U ACI,UazRiD,WAYRvC,CAAY,OAAZ,C;UOiiVP,U;UADP,YPoiVe,WOpIVH,WPoiVwB,GOpiVxB,C;UACL,IA AI,aAAJ,C;YACH,aPkiVuC,gB;YAA5B,WOjIVX,aPiiVgC,GOjIVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UP8kV A,iB;UACA,IAAK,WA3RyD,cA2RrD,CAAe,OAAf,CAAJ,C;;QA3RT,OA6RO,W;O;KAvSX,C;uFAaA,yB;MAA A,wE;MA6RA,+D;MA7RA,yD;QAUW,kBAAU,oB;QA6RD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,UA9RiD,W A8RvC,CAAY,OAAZ,C;UOpmVP,U;UADP,YPsmVe,WotmVH,WPsmVwB,GOtmVxB,C;UACL,IAAI,aAAJ,C; YACH,aPomVuC,gB;YAA5B,WOnmVX,aPmmVgC,GOnmVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UPgmVA,i B;UACA,IAAK,WAhSyD,cAgSrD,CAAe,OAAf,CAAJ,C;;QAhST,OAKSO,W;O;KA5SX,C;uFAaA,yB;MAAA,wE ;MAkSA,+D;MAISA,yD;QAUW,kBAAU,oB;QAKSD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,UAnSiD,WAmSvC, CAAY,OAAZ,C;UOtnVP,U;UADP,YPwnVe,WoxnVH,WPwnVwB,GOxnVxB,C;UACL,IAAI,aAAJ,C;YACH,aP snVuC,gB;YAA5B,WOmVX,aPqnVgC,GOmVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UPknVA,iB;UACA,IAAK ,WArSyD,cAqSrD,CAAe,OAAf,CAAJ,C;;QArST,OAuSO,W;O;KAjTX,C;uFAaA,yB;MAAA,wE;MAuSA,+D;M AvSA,yD;QAUW,kBAAU,oB;QAUd,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,UAXSiD,WAwSvC,CAAY,OAAZ, C;UOxoVP,U;UADP,YP0oVe,WO1oVH,WP0oVwB,GO1oVxB,C;UACL,IAAI,aAAJ,C;YACH,aPwoVuC,gB;YA A5B,WOvoVX,aPuoVgC,GOvoVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UPooVA,iB;UACA,IAAK,WA1SyD,cA 0SrD,CAAe,OAAf,CAAJ,C;;QA1ST,OA4SO,W;O;KAiTX,C;uFAaA,yB;MAAA,wE;MA4SA,+D;MA5SA,yD;QA UW,kBAAU,oB;QA4SD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,UA7SiD,WA6SvC,CAAY,OAAZ,C;UO1pVP,U; UADP,YP4pVe,WO5pVH,WP4pVwB,GO5pVxB,C;UACL,IAAI,aAAJ,C;YACH,aP0pVuC,gB;YAA5B,WOzpVX, aPypVgC,GOzpVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UPspVA,iB;UACA,IAAK,WA/SyD,cA+SrD,CAAe,OA Af,CAAJ,C;;QA/ST,OaiTO,W;O;KA3TX,C;uFAaA,yB;MAAA,wE;MAiTA,+D;MAjTA,yD;QAUW,kBAAU,oB; QAiTD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,UAlTiD,WaktvC,CAAY,OAAZ,C;UO5qVP,U;UADP,YP8qVe, WO9qVH,WP8qVwB,GO9qVxB,C;UACL,IAAI,aAAJ,C;YACH,aP4qVuC,gB;YAA5B,WO3qVX,aP2qVgC,GO3q VhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UPwqVA,iB;UACA,IAAK,WApTyD,cAoTrD,CAAe,OAAf,CAAJ,C;;Q ApTT,OAsTO,W;O;KAhUX,C;uFAaA,yB;MAAA,wE;MAStA,oC;MAAA,+D;MAAA,gC;MATa,yD;QAUW,kB AAU,oB;QAsTD,Q;QAAhB,iD;UAAgB,cAAhB,0B;UACI,UAvTiD,WAutvC,CAAY,oBAAZ,C;UO9rVP,U;UAD P,YPgsVe,WOhsvH,WPgsVwB,GOhsVxB,C;UACL,IAAI,aAAJ,C;YACH,aP8rVuC,gB;YAA5B,WO7rVX,aP6rV gC,GO7rVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UP0rVA,iB;UACA,IAAK,WazTyD,cAyTrD,CAAe,oBAAf,C AAJ,C;;QAzTT,OA2TO,W;O;KArUX,C;uFAaA,yB;MAAA,+D;MAAA,sD;QASoB,Q;QAAhB,wBAAGB,SAAhB ,gB;UAAgB,cAAA,SAAhB,M;UACI,UAAU,YAAY,OAAZ,C;UOp5UP,U;UADP,YP5Ue,WOt5UH,WP5UwB, GOt5UxB,C;UACL,IAAI,aAAJ,C;YACH,aPo5UuC,gB;YAA5B,WOn5UX,aPm5UgC,GOt5UhC,EAAS,MAAT,C; YACA,e;;YAEA,c;;UPg5UA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA, +D;MAAA,sD;QASoB,Q;QAAhB,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,UAAU,YAAY,OAAZ, C;UOr6UP,U;UADP,YPu6Ue,Wov6UH,WPu6UwB,GOv6UxB,C;UACL,IAAI,aAAJ,C;YACH,aPq6UuC,gB;YA

A5B,WOp6UX,aPo6UgC,GOp6UhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UPi6UA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,+D;MAAA,sD;QASoB,Q;QAAhB,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,UAAU,YAAY,OAAZ,C;UOt7UP,U;UADP,YPw7Ue,Wox7UH,WPw7UwB,GOx7UxB,C;UACL,IAAI,aAAJ,C;YACH,aPs7UuC,gB;YAA5B,WOr7UX,aPq7UgC,GOr7UhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UPk7UA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,+D;MAAA,sD;QASoB,Q;QAAhB,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,UAAU,YAAY,OAAZ,C;UOv8UP,U;UADP,YPy8Ue,Woz8UH,WPy8UwB,GOz8UxB,C;UACL,IAAI,aAAJ,C;YACH,aPu8UuC,gB;YAA5B,WOt8UX,aPs8UgC,GOt8UhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UPm8UA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,+D;MAAA,sD;QASoB,Q;QAAhB,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,UAAU,YAAY,OAAZ,C;UOx9UP,U;UADP,YP09Ue,Wo19UH,WP09UwB,GO19UxB,C;UACL,IAAI,aAAJ,C;YACH,aPw9UuC,gB;YAA5B,Wov9UX,aPu9UgC,GOv9UhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UPo9UA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,+D;MAAA,sD;QASoB,Q;QAAhB,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,UAAU,YAAY,OAAZ,C;UOz+UP,U;UADP,YP2+Ue,Wo3+UH,WP2+UwB,GO3+UxB,C;UACL,IAAI,aAAJ,C;YACH,aPy+UuC,gB;YAA5B,Wox+UX,aPw+UgC,GOx+UhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UPq+UA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,+D;MAAA,sD;QASoB,Q;QAAhB,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,UAAU,YAAY,OAAZ,C;UO1/UP,U;UADP,YP4/Ue,Wo5/UH,WP4/UwB,GO5/UxB,C;UACL,IAAI,aAAJ,C;YACH,aP0/UuC,gB;YAA5B,Woz/UX,aPy/UgC,GOz/UhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UPs/UA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,+D;MAAA,sD;QASoB,Q;QAAhB,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,UAAU,YAAY,OAAZ,C;UO3gVP,U;UADP,YP6gVe,Wo7gVH,WP6gVwB,GO7gVxB,C;UACL,IAAI,aAAJ,C;YACH,aP2gVuC,gB;YAA5B,Wo1gVX,aP0gVgC,GO1gVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UPugVA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,+D;MAAA,sD;QASoB,Q;QAAhB,wBAAGB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UACI,UAAU,YAAY,oBAAZ,C;UO5hVP,U;UADP,YP8hVe,Wo9hVH,WP8hVwB,GO9hVxB,C;UACL,IAAI,aAAJ,C;YACH,aP4hVuC,gB;YAA5B,Wo3hVX,aP2hVgC,GO3hVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UPwhVA,iB;UACA,IAAK,WAAI,oBAAJ,C;;QAET,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,+D;MAAA,sE;QAUoB,Q;QAAhB,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,UAAU,YAAY,OAAZ,C;UO9iVP,U;UADP,YPgjVe,WohjVH,WPgjVwB,GOhjVxB,C;UACL,IAAI,aAAJ,C;YACH,aP8iVuC,gB;YAA5B,Wo7iVX,aP6iVgC,GO7iVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UPoiVA,iB;UACA,IAAK,WAAI,eAAe,OAAf,CAAJ,C;;QAET,OAAO,W;O;KafX,C;0FakBA,yB;MAAA,+D;MAAA,sE;QAUoB,Q;QAAhB,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,UAAU,YAAY,OAAZ,C;UOhkVP,U;UADP,YPkkVe,WoIkVH,WPkkVwB,GOIkVxB,C;UACL,IAAI,aAAJ,C;YACH,aPkgVuC,gB;YAA5B,Wo/jVX,aP+jVgC,GO/jVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UP4jVA,iB;UACA,IAAK,WAAI,eAAe,OAAf,CAAJ,C;;QAET,OAAO,W;O;KafX,C;2FakBA,yB;MAAA,+D;MAAA,sE;QAUoB,Q;QAAhB,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,UAAU,YAAY,OAAZ,C;UOIIVP,U;UADP,YPoiVe,WoPlVH,WPoIVwB,GOPlVxB,C;UACL,IAAI,aAAJ,C;YACH,aPkIVuC,gB;YAA5B,WoJlVX,aPilVgC,GOJlVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UP8kVA,iB;UACA,IAAK,WAAI,eAAe,OAAf,CAAJ,C;;QAET,OAAO,W;O;KafX,C;2FakBA,yB;MAAA,+D;MAAA,sE;QAUoB,Q;QAAhB,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,UAAU,YAAY,OAAZ,C;UOpMVP,U;UADP,YPsmVe,WoTmVH,WPsmVwB,GOtmVxB,C;UACL,IAAI,aAAJ,C;YACH,aPomVuC,gB;YAA5B,WonmVX,aPmmVgC,GOnmVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UPgmVA,iB;UACA,IAAK,WAAI,eAAe,OAAf,CAAJ,C;;QAET,OAAO,W;O;KafX,C;2FakBA,yB;MAAA,+D;MAAA,sE;QAUoB,Q;QAAhB,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,UAAU,YAAY,OAAZ,C;UOtnVP,U;UADP,YPwnVe,WoxnVH,WPwnVwB,GOxnVxB,C;UACL,IAAI,aAAJ,C;YACH,aPsnVuC,gB;YAA5B,WOrnVX,aPqnVgC,GOrnVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UPknVA,iB;UACA,IAAK,WAAI,eAAe,OAAf,CAAJ,C;;QAET,OAAO,W;O;KafX,C;2FakBA,yB;MAAA,+D;MAAA,sE;QAUoB,Q;QAAhB,wBAAGB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,UAAU,YAAY,OAAZ,C;UOxoVP,U;UADP,YP0oVe,Wo1oVH,WP0oVwB,GO1oVxB,C;UACL,IAAI,aAAJ,C;YACH,aPwoVuC,gB;YAA5B,WovoVX,aPuoVgC,GOvoVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UPooVA,iB;UACA,IAAK,WAAI,eAAe,OAAf,CAAJ,C;;QAET,OAAO,W;O;KafX,C;2FakBA,yB;MAAA,+D;MAAA,sE;QAUoB,Q;QAAhB,wBAAGB,SAAhB,gB;

AA,sC;QAYc,Q;QAFV,UAAU,c;QACV,WAAW,gB;QACX,wBAAU,SAAV,gB;UAAU,QAAA,SAAV,M;UACI,
UAAU,SAAS,CAAT,C;UACV,IAAI,GAAI,WAAI,GAJ,CAAR,C;YACI,IAAK,WAAI,CAAJ,C;;QAEb,OAAO,I;
O;KAjBX,C;4FAoBA,yB;MAAA,2D;MAAA,+D;MAAA,sC;QAWc,Q;QAFV,UAAU,c;QACV,WAAW,gB;QAC
X,wBAAU,SAAV,gB;UAAU,QAAA,SAAV,M;UACI,UAAU,SAAS,CAAT,C;UACV,IAAI,GAAI,WAAI,GAJ,C
AAR,C;YACI,IAAK,WAAI,CAAJ,C;;QAEb,OAAO,I;O;KAhBX,C;4FAmBA,yB;MAAA,2D;MAAA,+D;MAAA,s
C;QAWc,Q;QAFV,UAAU,c;QACV,WAAW,gB;QACX,wBAAU,SAAV,gB;UAAU,QAAA,SAAV,M;UACI,UAA
U,SAAS,CAAT,C;UACV,IAAI,GAAI,WAAI,GAJ,CAAR,C;YACI,IAAK,WAAI,CAAJ,C;;QAEb,OAAO,I;O;K
AhBX,C;4FAmBA,yB;MAAA,2D;MAAA,+D;MAAA,sC;QAWc,Q;QAFV,UAAU,c;QACV,WAAW,gB;QACX,w
BAAU,SAAV,gB;UAAU,QAAA,SAAV,M;UACI,UAAU,SAAS,CAAT,C;UACV,IAAI,GAAI,WAAI,GAJ,CAA
R,C;YACI,IAAK,WAAI,CAAJ,C;;QAEb,OAAO,I;O;KAhBX,C;4FAmBA,yB;MAAA,2D;MAAA,+D;MAAA,sC;
QAWc,Q;QAFV,UAAU,c;QACV,WAAW,gB;QACX,wBAAU,SAAV,gB;UAAU,QAAA,SAAV,M;UACI,UAAU,
SAAS,CAAT,C;UACV,IAAI,GAAI,WAAI,GAJ,CAAR,C;YACI,IAAK,WAAI,CAAJ,C;;QAEb,OAAO,I;O;K
AhBX,C;4FAmBA,yB;MAAA,2D;MAAA,+D;MAAA,sC;QAWc,Q;QAFV,UAAU,c;QACV,WAAW,gB;QACX,wB
AAU,SAAV,gB;UAAU,QAAA,SAAV,M;UACI,UAAU,SAAS,CAAT,C;UACV,IAAI,GAAI,WAAI,GAJ,CAAR,C
;YACI,IAAK,WAAI,CAAJ,C;;QAEb,OAAO,I;O;KAhBX,C;4FAmBA,yB;MAAA,2D;MAAA,+D;MAAA,sC;QA
Wc,Q;QAFV,UAAU,c;QACV,WAAW,gB;QACX,wBAAU,SAAV,gB;UAAU,QAAA,SAAV,M;UACI,UAAU,SA
AS,CAAT,C;UACV,IAAI,GAAI,WAAI,GAJ,CAAR,C;YACI,IAAK,WAAI,CAAJ,C;;QAEb,OAAO,I;O;KAhBX
,C;4FAmBA,yB;MAAA,2D;MAAA,+D;MAAA,sC;QAWc,Q;QAFV,UAAU,c;QACV,WAAW,gB;QACX,wBAA
U,SAAV,gB;UAAU,QAAA,SAAV,M;UACI,UAAU,SAAS,CAAT,C;UACV,IAAI,GAAI,WAAI,GAJ,CAAR,C;
YACI,IAAK,WAAI,CAAJ,C;;QAEb,OAAO,I;O;KAhBX,C;4FAmBA,yB;MAAA,2D;MAAA,+D;MAAA,oC;MA
AA,gC;MAAA,sC;QAWc,Q;QAFV,UAAU,c;QACV,WAAW,gB;QACX,wBAAU,SAAV,gB;UAAU,QAAV,UAA
U,SAAV,O;UACI,UAAU,SAAS,cAAT,C;UACV,IAAI,GAAI,WAAI,GAJ,CAAR,C;YACI,IAAK,WAAI,cAAJ,
C;;QAEb,OAAO,I;O;KAhBX,C;IAmBA,qC;MAQI,UAAe,aAAL,SAAK,C;MACX,YAAJ,GAAL,EAAU,KAAV,C;
MACJ,OAAO,G;K;IAGX,uC;MAQI,UAAe,eAAL,SAAK,C;MACX,YAAJ,GAAL,EAAU,KAAV,C;MACJ,OAAO,
G;K;IAGX,uC;MAQI,UAAe,eAAL,SAAK,C;MACX,YAAJ,GAAL,EAAU,KAAV,C;MACJ,OAAO,G;K;IAGX,uC;
MAQI,UAAe,eAAL,SAAK,C;MACX,YAAJ,GAAL,EAAU,KAAV,C;MACJ,OAAO,G;K;IAGX,uC;MAQI,UAAe,
eAAL,SAAK,C;MACX,YAAJ,GAAL,EAAU,KAAV,C;MACJ,OAAO,G;K;IAGX,uC;MAQI,UAAe,eAAL,SAAK,
C;MACX,YAAJ,GAAL,EAAU,KAAV,C;MACJ,OAAO,G;K;IAGX,uC;MAQI,UAAe,eAAL,SAAK,C;MACX,YA
AJ,GAAL,EAAU,KAAV,C;MACJ,OAAO,G;K;IAGX,uC;MAQI,UAAe,eAAL,SAAK,C;MACX,YAAJ,GAAL,EAA
U,KAAV,C;MACJ,OAAO,G;K;IAGX,uC;MAQI,UAAe,eAAL,SAAK,C;MACX,YAAJ,GAAL,EAAU,KAAV,C;M
ACJ,OAAO,G;K;IAGX,oC;MAMI,UAAe,aAAL,SAAK,C;MACX,YAAJ,GAAL,EAAU,KAAV,C;MACJ,OAAO,G
;K;IAGX,sC;MAMI,UAAe,eAAL,SAAK,C;MACX,YAAJ,GAAL,EAAU,KAAV,C;MACJ,OAAO,G;K;IAGX,sC;
MAMI,UAAe,eAAL,SAAK,C;MACX,YAAJ,GAAL,EAAU,KAAV,C;MACJ,OAAO,G;K;IAGX,sC;MAMI,UAAe,
eAAL,SAAK,C;MACX,YAAJ,GAAL,EAAU,KAAV,C;MACJ,OAAO,G;K;IAGX,sC;MAMI,UAAe,eAAL,SAAK,
C;MACX,YAAJ,GAAL,EAAU,KAAV,C;MACJ,OAAO,G;K;IAGX,sC;MAMI,UAAe,eAAL,SAAK,C;MACX,YA
AJ,GAAL,EAAU,KAAV,C;MACJ,OAAO,G;K;IAGX,sC;MAMI,UAAe,eAAL,SAAK,C;MACX,YAAJ,GAAL,EA
AU,KAAV,C;MACJ,OAAO,G;K;IAGX,sC;MAMI,UAAe,eAAL,SAAK,C;MACX,YAAJ,GAAL,EAAU,KAAV,C;
MACJ,OAAO,G;K;IAGX,iC;MAMI,OAAO,wBAAa,qBAaIB,YAAY,gBAAZ,CAAJB,CAAb,C;K;IAGX,mC;MAMI,OAAO,0B
AAa,qBAAoB,YAAY,gBAAZ,CAApB,CAAb,C;K;IAGX,mC;MAMI,OAAO,0BAAa,qBAAqB,YAAY,gBAAZ,C
AArB,CAAb,C;K;IAGX,mC;MAMI,OAAO,0BAAa,qBAAmB,YAAY,gBAAZ,CAAnB,CAAb,C;K;IAGX,mC;M
AMI,OAAO,0BAAa,qBAAoB,YAAY,gBAAZ,CAApB,CAAb,C;K;IAGX,mC;MAMI,OAAO,0BAAa,qBAAqB,Y
AAY,gBAAZ,CAArB,CAAb,C;K;IAGX,mC;MAMI,OAAO,0BAAa,qBAAsB,YAAY,gBAAZ,CAAtB,CAAb,C;K;
IAGX,mC;MAMI,OAAO,0BAAa,qBAAuB,YAAY,gBAAZ,CAAvB,CAAb,C;K;IAGX,mC;MAMI,OAAO,0BAAa
,qBAAoB,YAAiB,eAAL,gBAAK,EAAa,GAAb,CAAJB,CAApB,CAAb,C;K;IAGX,iC;MAUI,UAAe,aAAL,SAAK,
C;MACX,OAAJ,GAAL,EAAO,KAAP,C;MACJ,OAAO,G;K;IAGX,mC;MAUI,UAAe,eAAL,SAAK,C;MACX,OA
AJ,GAAL,EAAO,KAAP,C;MACJ,OAAO,G;K;IAGX,mC;MAUI,UAAe,eAAL,SAAK,C;MACX,OAAJ,GAAL,EA
AO,KAAP,C;MACJ,OAAO,G;K;IAGX,mC;MAUI,UAAe,eAAL,SAAK,C;MACX,OAAJ,GAAL,EAAO,KAAP,C;

QAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IA/xLO,qBAAQ,CA+xLf,C;UAAe,OA
AO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OA
AO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR
,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OA
AO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IA9yLO,qBAAQ,CA8yLf,C;UAAe,OAAO,I;QACt
B,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC
3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SA
AS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KA
pBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IA7zLO,qBAAQ,CA6zLf,C;UAAe,OAAO,I;QACtB,cAAc,UA
AK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,S
AAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,
C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;8F
AuBA,yB;MAAA,8D;MAAA,sC;QAOI,IA5LO,qBAAQ,CA40Lf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL
,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT
,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IA
AI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;M
AAA,8D;MAAA,sC;QAOI,IA31LO,qBAAQ,CA21Lf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gB
AAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aA
AU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,
CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;M
AAA,sC;QAOI,IA12LO,qBAAQ,CA02Lf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAA
L,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,
OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KA
AJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QA
OI,IAz3LO,qBAAQ,CAy3Lf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;
QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SA
Ab,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,
UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,oC;MAAA,sC;QA
OI,IAx4LO,qBAAQ,CAw4Lf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;
QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,oBAAT,C;QACf,aAAU,CAAV,OAAa,SA
Ab,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,cAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI
,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;gFAuBA,yB;MAAA,sE;MAAA,8D;MkBhnbA,iB;
MIBgnbA,sC;QAEiB,Q;QAFb,IAr+LO,qBAAQ,CAq+Lf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL
,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBznbG,MA
AO,KIBynbO,QkBznbP,EIBynbiB,CkBznbjB,C;;QIB2nbd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,
8D;MkBtobA,iB;MIBsobA,sC;QAEiB,Q;QAFb,IAN/LO,qBAAQ,CAm/Lf,C;UAAe,MAAM,6B;QACrB,eAAe,SA
AS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR
,WkB/obG,MAAO,KIB+obO,QkB/obP,EIB+obiB,CkB/objB,C;;QIBipbd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MA
AA,sE;MAAA,8D;MkB5pbA,iB;MIB4pbA,sC;QAEiB,Q;QAFb,IAjgMO,qBAAQ,CAigMf,C;UAAe,MAAM,6B;Q
ACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAA
L,CAAT,C;UACR,WkBrqbG,MAAO,KIBqqbO,QkBrqbP,EIBqqbiB,CkBBrqbjB,C;;QIBuqbd,OAAO,Q;O;KAnBX,
C;kFAsBA,yB;MAAA,sE;MAAA,8D;MkB1rbA,iB;MIBkrbA,sC;QAEiB,Q;QAFb,IA/gMO,qBAAQ,CA+gMf,C;U
AAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,S
AAS,UAAK,CAAL,CAAT,C;UACR,WkB3rbG,MAAO,KIB2rbO,QkB3rbP,EIB2rbiB,CkB3rbjB,C;;QIB6rbd,OAA
O,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MkBxsbA,iB;MIBwsbA,sC;QAEiB,Q;QAFb,IA7hMO,qBA
AQ,CA6hMf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,i
B;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBjtbG,MAAO,KIBitbO,QkBjtbP,EIBitbiB,CkBjtbjB,C;;
QIBmtbd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MkB9tbA,iB;MIB8tbA,sC;QAEiB,Q;QAFb,I

A3iMO,qBAAQ,CA2iMf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBvubG,MAAO,KlBuubO,QkBvubP,ElBuu biB,CkBvubjB,C;;QlByubd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MkBpva,iB;MIBovbA,sC ;QAEiB,Q;QAFb,IAzjMO,qBAAQ,CAyjMf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;Q ACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkB7vbG,MAAO,KIB6vb O,QkB7vbp,ElB6vbiB,CkB7vbjB,C;;QlB+vbd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MkB1w bA,iB;MIB0wbA,sC;QAEiB,Q;QAFb,IAvkMO,qBAAQ,CAukMf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAA K,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBn xbg,MAAO,KlBmxbO,QkBnxbP,ElBmxbiB,CkBnxbjB,C;;QlBqxbd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,s E;MAAA,oC;MAAA,8D;MkBhybA,iB;MlBgybA,sC;QAEiB,Q;QAFb,IArlMO,qBAAQ,CAqlMf,C;UAAe,MAAM, 6B;QACrB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAA K,CAAL,EAAT,C;UACR,WkBzybG,MAAO,KlByybO,QkBzybP,ElByybiB,CkBzybjB,C;;QlB2ybd,OAAO,Q;O;K AnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MkBj0bA,iB;MlBi0bA,sC;QAEiB,Q;QAFb,IA3qMO,qBAAQ,CA2q Mf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI, QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkB10bG,MAAO,KlB00bO,QkB10bp,ElB00biB,CkB10bjB,C;;Ql B40bd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MkBv1bA,iB;MlBu1bA,sC;QAEiB,Q;QAFb,IA zrMO,qBAAQ,CAyrMf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAA U,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBh2bG,MAAO,KlBg2bO,QkBh2bp,ElBg2bi B,CkBh2bjB,C;;QlBk2bd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MkB72bA,iB;MlB62bA,sC; QAEiB,Q;QAFb,IAvsMO,qBAAQ,CAusMf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;Q ACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBt3bG,MAAO,KlB3b O,QkBt3bp,ElB3biB,CkBt3bjB,C;;QlBw3bd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MkBn4 bA,iB;MlBm4bA,sC;QAEiB,Q;QAFb,IArtMO,qBAAQ,CAqtMf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK ,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkB54 bG,MAAO,KlB44bO,QkB54bp,ElB44biB,CkB54bjB,C;;QlB84bd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE ;MAAA,8D;MkBz5bA,iB;MlBy5bA,sC;QAEiB,Q;QAFb,IAnuMO,qBAAQ,CAmuMf,C;UAAe,MAAM,6B;QACrB ,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CA AT,C;UACR,WkB16bG,MAAO,KlBk6bO,QkB16bp,ElBk6biB,CkB16bjB,C;;QlBo6bd,OAAO,Q;O;KAnBX,C;mFA sBA,yB;MAAA,sE;MAAA,8D;MkB/6bA,iB;MlB+6bA,sC;QAEiB,Q;QAFb,IAjvMO,qBAAQ,CAivMf,C;UAAe,M AAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,U AAK,CAAL,CAAT,C;UACR,WkBx7bG,MAAO,KlBw7bO,QkBx7bp,ElBw7biB,CkBx7bjB,C;;QlB07bd,OAAO,Q ;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MkB8bA,iB;MlBq8bA,sC;QAEiB,Q;QAFb,IA/vMO,qBAAQ,C A+vMf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UA CI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkB98bG,MAAO,KlB88bO,QkB98bp,ElB88biB,CkB98bjB,C;; QlBg9bd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MkB39bA,iB;MlB29bA,sC;QAEiB,Q;QAFb, IA7wMO,qBAAQ,CA6wMf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb, aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBp+bG,MAAO,KlBo+bO,QkBp+bp,El Bo+biB,CkBp+bjB,C;;QlBs+bd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,oC;MAAA,8D;MkBj/bA, iB;MlBi/bA,sC;QAEiB,Q;QAFb,IA3xMO,qBAAQ,CA2xMf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,C AAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,WkB1/bG ,MAAO,KlB0/bO,QkB1/bP,ElB0/biB,CkB1/bjB,C;;QlB4/bd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MA AA,8D;MAAA,sC;QAaiB,Q;QAFb,IA/2MO,qBAAQ,CA+2Mf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK, CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2B AAW,CAAX,KAAJ,C;YACI,WAAW,C;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;M AAA,sC;QAaiB,Q;QAFb,IA73MO,qBAAQ,CA63Mf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,C AAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,C AAX,KAAJ,C;YACI,WAAW,C;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC; QAaiB,Q;QAFb,IA34MO,qBAAQ,CA24Mf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;Q

ACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KA
AJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q
;QAFb,IAz5MO,qBAAQ,CAy5Mf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;
QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YA
CI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,I
Av6MO,qBAAQ,CAu6Mf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,a
AAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAA
W,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IAR7MO,q
BAAQ,CAq7Mf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAA
V,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QA
GnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IAn8MO,qBAAQ,C
Am8Mf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;U
ACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OA
AO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IAj9MO,qBAAQ,CAi9Mf,C;
UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ
,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;K
AnBX,C;mFAsBA,yB;MAAA,sE;MAAA,oC;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IA/9MO,qBAAQ,CA+9Mf,
C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QA
AQ,SAAS,sBAK,CAAL,EAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;
O;KAnBX,C;4FAsBA,yB;MAAA,8D;MkBlscA,iB;MIBkscA,sC;QAaiB,Q;QAFb,IARjNO,qBAAQ,CAqjNf,C;UAA
e,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,
UAAK,CAAL,CAAT,C;UACR,WkBzscG,MAAO,KlByscO,QkBzscP,ElBysciB,CkBzscjB,C;;QIB2scd,OAAO,Q;
O;KAjBX,C;8FAoBA,yB;MAAA,8D;MkBttcA,iB;MIBstcA,sC;QAaiB,Q;QAFb,IAjkNO,qBAAQ,CAikNf,C;UAA
e,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,
UAAK,CAAL,CAAT,C;UACR,WkB7tcG,MAAO,KIB6tcO,QkB7tcP,ElB6tciB,CkB7tcjB,C;;QIB+tcd,OAAO,Q;O;
KAjBX,C;8FAoBA,yB;MAAA,8D;MkB1ucA,iB;MIB0ucA,sC;QAaiB,Q;QAFb,IA7kNO,qBAAQ,CA6kNf,C;UAA
e,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,
UAAK,CAAL,CAAT,C;UACR,WkBjvcG,MAAO,KlBivcO,QkBjvcP,ElBivciB,CkBjvcjB,C;;QlBmvcd,OAAO,Q;
O;KAjBX,C;8FAoBA,yB;MAAA,8D;MkB9vcA,iB;MIB8vcA,sC;QAaiB,Q;QAFb,IAzlnO,qBAAQ,CAylNf,C;UA
Ae,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS
,UAAK,CAAL,CAAT,C;UACR,WkBrwcG,MAAO,KlBqwcO,QkBrwcP,ElBqwciB,CkBrwcjB,C;;QlBuwcd,OAA
O,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MkBlxcA,iB;MlBkxcA,sC;QAaiB,Q;QAFb,IArmNO,qBAAQ,CAqmNf,
C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,
SAAS,UAAK,CAAL,CAAT,C;UACR,WkBzxcG,MAAO,KlByxcO,QkBzxcP,ElByxciB,CkBzxcjB,C;;QIB2xcd,O
AAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MkBtycA,iB;MlBsyscA,sC;QAaiB,Q;QAFb,IAjnNO,qBAAQ,CAinN
f,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAA
Q,SAAS,UAAK,CAAL,CAAT,C;UACR,WkB7ycG,MAAO,KlB6ycO,QkB7ycP,ElB6yciB,CkB7ycjB,C;;QIB+ycd,
OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MkB1zcA,iB;MlB0zcA,sC;QAaiB,Q;QAFb,IA7nNO,qBAAQ,CA
6nNf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,Q
AAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBj0cG,MAAO,KlBi0cO,QkBj0cP,ElBi0ciB,CkBj0cjB,C;;QIBm0c
d,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MkB90cA,iB;MlB80cA,sC;QAaiB,Q;QAFb,IAzoNO,qBAAQ,C
AyoNf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,
QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBr1cG,MAAO,KlBq1cO,QkBr1cP,ElBq1ciB,CkBr1cjB,C;;QIBu
1cd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,oC;MAAA,8D;MkB12cA,iB;MlBk2cA,sC;QAaiB,Q;QAFb,IARpN
O,qBAAQ,CAqpNf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CA
AV,iB;UACI,QAAQ,SAAS,sBAK,CAAL,EAAT,C;UACR,WkBz2cG,MAAO,KlBy2cO,QkBz2cP,ElBy2ciB,Ck
Bz2cjB,C;;QIB22cd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MkBj4cA,iB;MlBi4cA,sC;QAaiB,Q;QAFb,IA
zuNO,qBAAQ,CAyuNf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,

CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBx4cG,MAAO,KIBw4cO,QkBx4cP,ElBw4ciB,CkBx4cjB,C;;QIB04cd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MkBr5cA,iB;MIBq5cA,sC;QAaiB,Q;QAFb,IArvNO,qBAAQ,CAqvNf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkB55cG,MAAO,KIB45cO,QkB55cP,ElB45ciB,CkB55cjB,C;;QIB85cd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MkBz6cA,iB;MIBy6cA,sC;QAaiB,Q;QAFb,IAjwNO,qBAAQ,CAiwNf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBh7cG,MAAO,KIBg7cO,QkBh7cP,ElBg7ciB,CkBh7cjB,C;;QIBk7cd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MkB77cA,iB;MIB67cA,sC;QAaiB,Q;QAFb,IA7wNO,qBAAQ,CA6wNf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBp8cG,MAAO,KIBo8cO,QkBp8cP,ElBo8ciB,CkBp8cjB,C;;QIBs8cd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MkBj9cA,iB;MIBi9cA,sC;QAaiB,Q;QAFb,IAzxNO,qBAAQ,CAyxNf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBx9cG,MAAO,KIBw9cO,QkBx9cP,ElBw9ciB,CkBx9cjB,C;;QIB09cd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MkBr+cA,iB;MIBq+cA,sC;QAaiB,Q;QAFb,IAryNO,qBAAQ,CAqyNf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkB5+cG,MAAO,KIB4+cO,QkB5+cP,ElB4+ciB,CkB5+cjB,C;;QIB8+cd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MkBz/cA,iB;MIBy/cA,sC;QAaiB,Q;QAFb,IAjzNO,qBAAQ,CAizNf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBhgdG,MAAO,KIBggdO,QkBhgdP,ElBggdiB,CkBhgdjB,C;;QIBk added,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MkB7gdA,iB;MIB6gdA,sC;QAaiB,Q;QAFb,IA7zNO,qBAAQ,CA6zNf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBphdG,MAAO,KIBohdO,QkBphdP,ElBohdiB,CkBphdjB,C;;QIBshdd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,oC;MAAA,8D;MkBjida,iB;MIBiida,sC;QAaiB,Q;QAFb,IAz0NO,qBAAQ,CAy0Nf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,WkBxidG,MAAO,KIBwidO,QkBxidP,ElBwidiB,CkBxidjB,C;;QIB0idd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IA35NO,qBAAQ,CA25Nf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IAv6NO,qBAAQ,CAu6Nf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IAN7NO,qBAAQ,CAm7Nf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IA/7NO,qBAAQ,CA+7Nf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IA38NO,qBAAQ,CA28Nf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IAv9NO,qBAAQ,CAu9Nf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IAN+NO,qBAAQ,CAm+Nf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IA/+NO,qBAAQ,CA++Nf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAj

BX,C;+FAoBA,yB;MAAA,oC;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IA3/NO,qBAAQ,CA2/Nf,C;UAAe,OAA
O,I;QACtB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAA
K,CAAL,EAAT,C;UACR,IAAI,2BAAW,CAAX,KAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;wFA
oBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IAj100,qBAAQ,CAi1Of,C;UAAe,MAAM,6B;QACrB
,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CA
AT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,
OAAO,Q;O;KAnBX,C;0FAsBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IA/100,qBAAQ,CA+1Of,
C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QA
AQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C
;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;0FAsBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QAaiB,Q;QA
Fb,IA7m00,qBAAQ,CA6mOf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QA
Ab,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,C
AAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;0FAsBA,yB;MAAA,sE;MAA
A,8D;MAAA,kD;QAaiB,Q;QAFb,IA3n00,qBAAQ,CA2nOf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,C
AAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UA
AW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;
0FAsBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IAzo00,qBAAQ,CAYoOf,C;UAAe,MAAM,6B;Q
ACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAA
L,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QA
GnB,OAAO,Q;O;KAnBX,C;0FAsBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IAvp00,qBAAQ,C
AupOf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UA
CI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,C
AAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;0FAsBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QAai
B,Q;QAFb,IArq00,qBAAQ,CAqqOf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+
B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAa
kB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;0FAsBA,yB;MAAA,sE;
MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IANr00,qBAAQ,CAMrOf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UA
AK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAA
I,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAn
BX,C;0FAsBA,yB;MAAA,sE;MAAA,oC;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IAjs00,qBAAQ,CAisOf,C;UA
Ae,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,S
AAS,sBAAK,CAAL,EAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YA
CI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;oGAsBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IArx00,qB
AAQ,CAqxOf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB
;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAak
C,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;Q
AFb,IAjy00,qBAAQ,CAiyOf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,a
AAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB
,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD
;QAWiB,Q;QAFb,IA7y00,qBAAQ,CA6yOf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QA
CF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,
EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAA
A,8D;MAAA,kD;QAWiB,Q;QAFb,IAzz00,qBAAQ,CAYzOf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CA
AL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UA
W,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;s
GAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IAr000,qBAAQ,CAq0Of,C;UAAe,OAAO,I;QACtB,eAAe,
SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;U
ACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,

Q;O;KAjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IAj100,qBAAQ,CAi1Of,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IA7100,qBAAQ,CA61Of,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IAz200,qBAAQ,CAy2Of,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IAr300,qBAAQ,CAq3Of,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;IAoBA,8B;MASiB,Q;MAFb,IAv800,qBAAQ,CAu8Of,C;QAAe,OAAO,I;MACTB,UAAU,UAAK,CAAL,C;MACG,+B;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,MkB3leG,MAAO,KIB2leE,GkB3leF,EIB2leO,Ck3leP,C;;MIB6led,OAAO,G;K;IAGX,gC;MASiB,Q;MAFb,IAv900,qBAAQ,CAu9Of,C;QAAe,OAAO,I;MACTB,UAAU,UAAK,CAAL,C;MACG,+B;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,MkbtneG,MAAO,KIBsneE,GkbtneF,EIBsneO,CkbtneP,C;;MIBwned,OAAO,G;K;IAGX,gC;MAOiB,Q;MAFb,IAr+00,qBAAQ,CAq+Of,C;QAAe,OAAO,I;MACTB,UAAU,UAAK,CAAL,C;MACG,+B;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,sBAAM,CAAN,KAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,gC;MAOiB,Q;MAFb,IA3+00,qBAAQ,CA2+Of,C;QAAe,OAAO,I;MACTB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,gC;MAOiB,Q;MAFb,IAj/00,qBAAQ,CAi/Of,C;QAAe,OAAO,I;MACTB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,gC;MAOiB,Q;MAFb,IAv/00,qBAAQ,CAu/Of,C;QAAe,OAAO,I;MACTB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,gC;MAOiB,Q;MAFb,IA7/00,qBAAQ,CA6/Of,C;QAAe,OAAO,I;MACTB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,oBAAM,CAAN,KAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,gC;MASiB,Q;MAFb,IArPO,qBAAQ,CAqPf,C;QAAe,OAAO,I;MACTB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,MkB5seG,MAAO,KIB4seF,Gk5seF,EIB4seO,Ck5seP,C;;MIB8sed,OAAO,G;K;IAGX,gC;MASiB,Q;MAFb,IA7gPO,qBAAQ,CA6gPf,C;QAAe,OAAO,I;MACTB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,wC;MAGI,OAAO,yBAAc,UAAd,C;K;IAGX,0C;MAGI,OAAO,2BAAc,UAAd,C;K;IAGX,0C;MAGI,OAAO,2BAAc,UAAd,C;K;IAGX,0C;MAGI,OAAO,2BAAc,UAAd,C;K;IAGX,0C;MAGI,OAAO,2BAAc,UAAd,C;K;IAGX,0C;MAGI,OAAO,2BAAc,UAAd,C;K;IAGX,0C;MAGI,OAAO,2BAAc,UAAd,C;K;IAGX,8C;MAOiB,Q;MAFb,IA/oPO,qBAAQ,CA+oPf,C;QAAe,OAAO,I;MACTB,UAAU,UAAK,CAAL,C;MACG,+B;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAaA,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,gD;MAOiB,Q;MAFb,IArpPO,qBAAQ,CAqpPf,C;QAAe,OAAO,I;MACTB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAaA,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,gD;MAOiB,Q;MAFb,IAjqPO,qBAAQ,CAiqPf,C;QAAe,OAAO,I;MACTB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;Q

f,C;YAAe,qBAAO,I;YAAP,uB;;UACf,cAAc,UAAK,CAAL,C;UACd,gBAAqB,wB;UACrB,IAAI,cAAa,CAAjB,C;YAAoB,qBAAO,O;YAAP,uB;;UACpB,eArLmB,QAqLJ,CAAS,OAAT,C;UACf,aAAU,CAAV,OAAa,SAAb,M;YACI,QAAQ,UAAK,CAAL,C;YACR,QAxLe,QAwLP,CAAS,CAAT,C;YACR,IAAI,2BAAW,CAAX,KAAJ,C;cACI,UAAU,C;cACV,WAAW,C;;;UAGnB,qBAAO,O;;;QA9LP,yB;O;KAHJ,C;kFAMA,yB;MA8LA,8D;MAAA,oC;MA9LA,sC;QAGW,sB;;UAkMP,IA3/PO,qBAAQ,CA2/Pf,C;YAAe,qBAAO,I;YAAP,uB;;UACf,cAAc,UAAK,CAAL,C;UACd,gBAAqB,wB;UACrB,IAAI,cAAa,CAAjB,C;YAAoB,qBAAO,O;YAAP,uB;;UACpB,eAtMmB,QAsMJ,CAAS,oBAAT,C;UACf,aAAU,CAAV,OAAa,SAAb,M;YACI,QAAQ,UAAK,CAAL,C;YACR,QAzMe,QAyMP,C;AAS,cAAT,C;YACR,IAAI,2BAAW,CAAX,KAAJ,C;cACI,UAAU,C;cACV,WAAW,C;;;UAGnB,qBAAO,O;;;QA/MP,yB;O;KAHJ,C;4FAMA,yB;MAAA,8D;MAAA,sC;QAOI,IA4PO,qBAAQ,CAM4Pf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;YAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IA5PO,qBAAQ,CAk5Pf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;YAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IAj6PO,qBAAQ,CAi6Pf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;YAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IAh7PO,qBAAQ,CAg7Pf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;YAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IA7PO,qBAAQ,CA+7Pf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;YAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IA9PO,qBAAQ,CA88Pf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;YAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IA79PO,qBAAQ,CA69Pf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;YAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IA5+PO,qBAAQ,CA4+Pf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;YAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,cAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,oC;MAAA,sC;QAOI,IA3/PO,qBAAQ,CA2/Pf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;YAAoB,OAAO,O;QAC3B,eAAe,SAAS,oBAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,cAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;gFAuBA,yB;MAAA,sE;MAAA,8D;MkB/gfA,iB;MIB+gfA,sC;QAEiB,Q;QAFb,IAxlQO,qBAAQ,CAwlQf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBXhfG,MAAO,KIBwhfO,QkBXhfP,EIBwhfiB,CkBXhfjB,C;;QIB0hfd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MkBrifA,iB;MIBqifA,sC;QAEiB,Q;QAFb,IAtmQO,qBAAQ,CAsmQf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;Q

ACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkB9ifG,MAAO,KIB8ifO, QkB9ifP,EIB8ifiB,CkB9ifjB,C;;QIBgjfD,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MkB3jfA,iB; MIB2jfA,sC;QAeiB,Q;QAFb,IApnQO,qBAAQ,CAonQf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL, CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBpkfG,MA AAO,KIBokfO,QkBpkfP,EIBokfiB,CkBpkfjB,C;;QIBskfd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8 D;MkBjlfA,iB;MIBilfA,sC;QAeiB,Q;QAFb,IAloQO,qBAAQ,CAkoQf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS, UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR, WkB1lfG,MAAO,KIB0lfO,QkB1lfP,EIB0lfiB,CkB1lfjB,C;;QIB4lfd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,s E;MAAA,8D;MkBvmfA,iB;MIBumfA,sC;QAeiB,Q;QAFb,IAhpQO,qBAAQ,CAgpQf,C;UAAe,MAAM,6B;QACr B,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,C AAT,C;UACR,WkBhnfG,MAAO,KIBgnfO,QkBhnfP,EIBgnfiB,CkBhnfjB,C;;QIBknfd,OAAO,Q;O;KAnBX,C;kF AsBA,yB;MAAA,sE;MAAA,8D;MkB7nfA,iB;MIB6nfA,sC;QAeiB,Q;QAFb,IA9pQO,qBAAQ,CA8pQf,C;UAAe, MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS ,UAAK,CAAL,CAAT,C;UACR,WkBtofG,MAAO,KIBsofO,QkBtofP,EIBsofiB,CkBtofjB,C;;QIBwofd,OAAO,Q;O ;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MkBnfpA,iB;MIBmpfA,sC;QAeiB,Q;QAFb,IA5qQO,qBAAQ,CA 4qQf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI ,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkB5pfG,MAAO,KIB4pfO,QkB5pfP,EIB4pfiB,CkB5pfjB,C;;QIB 8pfd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MkBzqfA,iB;MIByqfA,sC;QAeiB,Q;QAFb,IA1r QO,qBAAQ,CA0rQf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU, CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkB1rfG,MAAO,KIBkrfO,QkB1rfP,EIBkrfiB,CkB 1rfjB,C;;QIBorfd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,oC;MAAA,8D;MkB/rfA,iB;MIB+rfA,sC; QAeiB,Q;QAFb,IAxsQO,qBAAQ,CAwsQf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;Q ACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,WkBxsfG,MAAO,KIBwsf O,QkBxsfP,EIBwsfiB,CkBxsfjB,C;;QIB0sfd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MkBhufA ,iB;MIBgufA,sC;QAeiB,Q;QAFb,IA9xQO,qBAAQ,CA8xQf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,C AAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBzufG, MAAO,KIByufO,QkBzufP,EIByufiB,CkBzufjB,C;;QIB2ufd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAA A,8D;MkBtvfA,iB;MIBsvfA,sC;QAeiB,Q;QAFb,IA5yQO,qBAAQ,CA4yQf,C;UAAe,MAAM,6B;QACrB,eAAe,S AAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;U ACR,WkB/vfG,MAAO,KIB+vfO,QkB/vfP,EIB+vfjB,CkB/vfjB,C;;QIBiwfd,OAAO,Q;O;KAnBX,C;mFAsBA,yB; MAAA,sE;MAAA,8D;MkB5wfA,iB;MIB4wfA,sC;QAeiB,Q;QAFb,IA1zQO,qBAAQ,CA0zQf,C;UAAe,MAAM,6 B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,C AAL,CAAT,C;UACR,WkBxrfG,MAAO,KIBqxfO,QkBxrfP,EIBqxfiB,CkBxrfjB,C;;QIBuxfd,OAAO,Q;O;KAnBX, C;mFAsBA,yB;MAAA,sE;MAAA,8D;MkBlyfA,iB;MIBkyfA,sC;QAeiB,Q;QAFb,IAx0QO,qBAAQ,CAw0Qf,C;U AAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,S AAS,UAAK,CAAL,CAAT,C;UACR,WkB3yfG,MAAO,KIB2yfO,QkB3yfP,EIB2yfiB,CkB3yfjB,C;;QIB6yfd,OAA O,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MkBxzfA,iB;MIBwzfA,sC;QAeiB,Q;QAFb,IA1QO,qBAA Q,CAs1Qf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB; UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBj0fG,MAAO,KIBi0fO,QkBj0fP,EIBi0fiB,CkBj0fjB,C;;Q IBm0fd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MkB90fA,iB;MIB80fA,sC;QAeiB,Q;QAFb,IA p2QO,qBAAQ,CAo2Qf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aA AU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBv1fG,MAAO,KIBu1fO,QkBv1fP,EIBu1fi B,CkBv1fjB,C;;QIBy1fd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MkBp2fA,iB;MIBo2fA,sC;Q AeiB,Q;QAFb,IA13QO,qBAAQ,CAk3Qf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QAC F,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkB72fG,MAAO,KIB62fO,Q kB72fP,EIB62fiB,CkB72fjB,C;;QIB+2fd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MkB13fA,iB ;MIB03fA,sC;QAeiB,Q;QAFb,IAh4QO,qBAAQ,CAg4Qf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAA L,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBn4fG,M

AAO,KlBm4fO,Qkbn4fP,ElBm4fiB,Ckbn4fjB,C;;QlBq4fd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAA
A,oC;MAAA,8D;MkBs5fA,iB;MlBg5fA,sC;QAeiB,Q;QAFb,IA94QO,qBAAQ,CA84Qf,C;UAAe,MAAM,6B;QA
CrB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAA
L,EAAT,C;UACR,WkBs5fG,MAAO,KlBy5fO,QkBs5fP,ElBy5fiB,CkBs5fjB,C;;QlB25fd,OAAO,Q;O;KAnBX,C;
mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IAI+QO,qBAAQ,CAk+Qf,C;UAAe,MAAM,6B;Q
ACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAA
L,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,y
B;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IAh/QO,qBAAQ,CAg/Qf,C;UAAe,MAAM,6B;QACrB,eAA
e,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;
UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,s
E;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IA9/QO,qBAAQ,CA8/Qf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UA
AK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAA
I,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8
D;MAAA,sC;QAaiB,Q;QAFb,IA5gRO,qBAAQ,CA4gRf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL
,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,
CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,s
C;QAaiB,Q;QAFb,IA1hRO,qBAAQ,CA0hRf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;
QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,K
AAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,
Q;QAFb,IAxiRO,qBAAQ,CAwiRf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;
QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YA
CI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,I
AtjRO,qBAAQ,CAsjRf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAA
U,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,
C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IApkRO,qBA
AQ,CAokRf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,i
B;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB
,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,oC;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IALlRO,qBA
AQ,CAklRf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,i
B;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGn
B,OAAO,Q;O;KAnBX,C;4FAsBA,yB;MAAA,8D;MkBjmgBA,iB;MlBimgBA,sC;QAaiB,Q;QAFb,IAxqRO,qBAA
Q,CAwqRf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;U
ACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBsimgBG,MAAO,KlBwmgBO,QkBsimgBP,ElBwmgBiB,C
kBsimgBjB,C;;QlB0mgBd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MkBrngBA,iB;MlBqngBA,sC;QAaiB,
Q;QAFb,IAprRO,qBAAQ,CAorRf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QA
Ab,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBsngBG,MAAO,KlB4ngBO,QkB5n
gBP,ElB4ngBiB,CkB5ngBjB,C;;QlB8ngBd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MkBzogBA,iB;MlByo
gBA,sC;QAaiB,Q;QAFb,IAhsRO,qBAAQ,CagsRf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,
C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBhpgBG,MAAO,Kl
BgpgBO,QkBhpgBP,ElBgpgBiB,CkBhpgBjB,C;;QlBkpgBd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MkBs
7pgBA,iB;MlB6pgBA,sC;QAaiB,Q;QAFb,IA5sRO,qBAAQ,CA4sRf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAA
K,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBs
qgBG,MAAO,KlBoqgBO,QkBpqgBP,ElBoqgBiB,CkBpqgBjB,C;;QlBsqqBd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;
MAAA,8D;MkBsrgBA,iB;MlBimgBA,sC;QAaiB,Q;QAFb,IAxtRO,qBAAQ,CAwtRf,C;UAAe,OAAO,I;QACtB,eA
Ae,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,
C;UACR,WkBsrgBG,MAAO,KlBwrgBO,QkBxrgBP,ElBwrgBiB,CkBxrgBjB,C;;QlB0rgBd,OAAO,Q;O;KAjBX,C
;8FAoBA,yB;MAAA,8D;MkBrsgBA,iB;MlBqsgBA,sC;QAaiB,Q;QAFb,IApuRO,qBAAQ,CAouRf,C;UAAe,OAA
O,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,

CAAL,CAAT,C;UACR,WkB5sgBG,MAAO,KIB4sgBO,QkB5sgBP,EIB4sgBiB,CkB5sgBjB,C,;QIB8sgBd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MkBztgBA,iB;MIBytgBA,sC;QAaiB,Q;QAFb,IAhvRO,qBAAQ,CAGvRf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBhugBG,MAAO,KIBgugBO,QkBhugBP,EIBgugBiB,CkBhugBjB,C,;QIBkugBd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MkB7ugBA,iB;MIB6ugBA,sC;QAaiB,Q;QAFb,IA5vRO,qBAAQ,CA4vRf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBpvgBG,MAAO,KIBovgBO,QkBpvgBP,EIBovgBiB,CkBpvgBjB,C,;QIBsvgBd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,oC;MAAA,8D;MkBjwgBA,iB;MIBiwgBA,sC;QAaiB,Q;QAFb,IAxwRO,qBAAQ,CAwwRf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,WkBxwgBG,MAAO,KIBwwgBO,QkBxwgBP,EIBwwgBiB,CkBxwgBjB,C,;QIB0wgBd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MkBhygBA,iB;MIBgygBA,sC;QAaiB,Q;QAFb,IA51RO,qBAAQ,CA41Rf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBvygBG,MAAO,KIBuygBO,QkBvygBP,EIBuygBiB,CkBvygBjB,C,;QIByygBd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MkBpzgBA,iB;MIBozgBA,sC;QAaiB,Q;QAFb,IAx2RO,qBAAQ,CAw2Rf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkB3zgBG,MAAO,KIB2zgBO,QkB3zgBP,EIB2zgBiB,CkB3zgBjB,C,;QIB6zgBd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MkBx0gBA,iB;MIBw0gBA,sC;QAaiB,Q;QAFb,IAp3RO,qBAAQ,CAo3Rf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkB/0gBG,MAAO,KIB+0gBO,QkB/0gBP,EIB+0gBiB,CkB/0gBjB,C,;QIBi1gBd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MkB51gBA,iB;MIB41gBA,sC;QAaiB,Q;QAFb,IAh4RO,qBAAQ,CAg4Rf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBn2gBG,MAAO,KIBm2gBO,QkBn2gBP,EIBm2gBiB,CkBn2gBjB,C,;QIBq2gBd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MkBh3gBA,iB;MIBg3gBA,sC;QAaiB,Q;QAFb,IA54RO,qBAAQ,CA44Rf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBv3gBG,MAAO,KIBu3gBO,QkBv3gBP,EIBu3gBiB,CkBv3gBjB,C,;QIBy3gBd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MkBp4gBA,iB;MIBo4gBA,sC;QAaiB,Q;QAFb,IAx5RO,qBAAQ,CAw5Rf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkB34gBG,MAAO,KIB24gBO,QkB34gBP,EIB24gBiB,CkB34gBjB,C,;QIB64gBd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MkBx5gBA,iB;MIBw5gBA,sC;QAaiB,Q;QAFb,IAp6RO,qBAAQ,CAo6Rf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkB/5gBG,MAAO,KIB+5gBO,QkB/5gBP,EIB+5gBiB,CkB/5gBjB,C,;QIBi6gBd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MkB56gBA,iB;MIB46gBA,sC;QAaiB,Q;QAFb,IAh7RO,qBAAQ,CAg7Rf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WkBn7gBG,MAAO,KIBm7gBO,QkBn7gBP,EIBm7gBiB,CkBn7gBjB,C,;QIBq7gBd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,oC;MAAA,8D;MkBh8gBA,iB;MIBg8gBA,sC;QAaiB,Q;QAFb,IA57RO,qBAAQ,CA47Rf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,WkBv8gBG,MAAO,KIBu8gBO,QkBv8gBP,EIBu8gBiB,CkBv8gBjB,C,;QIBy8gBd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IA9gSO,qBAAQ,CA8gSf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C,;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IA1hSO,qBAAQ,CA0hSf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C,;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IAtiSO,qBAAQ,CAsiSf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C,;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,

Q;QAFb,IAIjSO,qBAAQ,CAkjSf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAA b,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,W AAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IA9jSO,qBAAQ, CA8jSf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI, QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO, Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IA1kSO,qBAAQ,CA0kSf,C;UAAe,OAAO,I; QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CA AL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA, yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IA1lSO,qBAAQ,CAslSf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAA K,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI, 2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC; QAWiB,Q;QAFb,IAImSO,qBAAQ,CAkmSf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QAC F,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ, C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,oC;MAAA,8D;MAAA,sC;QAWiB,Q;Q AFb,IA9mSO,qBAAQ,CA8mSf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;QAA b,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,W AAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;wFAoBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IApsS O,qBAAQ,CAosSf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,C AAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX ,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;0FAsBA,yB;MAAA,sE;MAAA,8D;MAA A,kD;QAaiB,Q;QAFb,IAItSO,qBAAQ,CAktSf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C ;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QA AR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;0FAsBA,yB;M AAA,sE;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IAhuSO,qBAAQ,CAGuSf,C;UAAe,MAAM,6B;QACrB,eAAe,SA AS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UA CR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q; O;KAnBX,C;0FAsBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IA9uSO,qBAAQ,CA8uSf,C;UAAe, MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS ,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,W AAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;0FAsBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IA5vS O,qBAAQ,CA4vSf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,C AAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX ,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;0FAsBA,yB;MAAA,sE;MAAA,8D;MAA A,kD;QAaiB,Q;QAFb,IA1wSO,qBAAQ,CA0wSf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT ,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ, QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;0FAsBA,yB ;MAAA,sE;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IAxxSO,qBAAQ,CAwxSf,C;UAAe,MAAM,6B;QACrB,eAAe ,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C; UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO ,Q;O;KAnBX,C;0FAsBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IAtySO,qBAAQ,CAsySf,C;UAA e,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAA S,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI, WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;0FAsBA,yB;MAAA,sE;MAAA,oC;MAAA,8D;MAAA,kD;QAaiB,Q; QAFb,IApzSO,qBAAQ,CAozSf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;Q AAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB, CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;oGAsBA,yB;MAAA,8D;M AAA,kD;QAWiB,Q;QAFb,IAx4SO,qBAAQ,CAw4Sf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAA T,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,

QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB ;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IAp5SO,qBAAQ,CAo5Sf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAA K,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI, UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX ,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IAh6SO,qBAAQ,CAg6Sf,C;UAAe,OAAO,I;QACtB,eA Ae,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT, C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OA AO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IA56SO,qBAAQ,CA46Sf,C;UAAe,OA AO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAA K,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW, C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IAx7SO,qBAAQ,CAw7 Sf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAA Q,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C; YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IAp8SO, qBAAQ,CAo8Sf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV, iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAA kC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q; QAFb,IAh9SO,qBAAQ,CAg9Sf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAA b,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CA AIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,8D;MAA A,kD;QAWiB,Q;QAFb,IA59SO,qBAAQ,CA49Sf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C; QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QA AR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;M AAA,oC;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IAx+SO,qBAAQ,CAw+Sf,C;UAAe,OAAO,I;QACtB,eAAe,SA AS,sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UA CR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q; O;KAjBX,C;IAoBA,8B;MASiB,Q;MAFb,IA1jTO,qBAAQ,CA0jTf,C;QAAe,OAAO,I;MACTB,UAAU,UAAK,CA AL,C;MACG,+B;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,MkB1/hBG,MAAO,KIB0/hBE, GkB1/hBF,EIB0/hBO,CkB1/hBP,C;;;MIB4/hBd,OAAO,G;K;IAGX,gC;MASiB,Q;MAFb,IA1kTO,qBAAQ,CA0kTf ,C;QAAe,OAAO,I;MACTB,UAAU,UAAK,CAAL,C;MACG,+B;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,C AAL,C;QACR,MkBrhiBG,MAAO,KIBqhiBE,GkBrhiBF,EIBqhiBO,CkBrihiBP,C;;;MIBuhiBd,OAAO,G;K;IAGX,g C;MAOiB,Q;MAFb,IAx1TO,qBAAQ,CAw1Tf,C;QAAe,OAAO,I;MACTB,UAAU,UAAK,CAAL,C;MACG,+B;MA Ab,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,sBAAM,CAAN,KAAJ,C;UAAa,MAAM,C;;;MA EvB,OAAO,G;K;IAGX,gC;MAOiB,Q;MAFb,IA91TO,qBAAQ,CA81Tf,C;QAAe,OAAO,I;MACTB,UAAU,UAAK, CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAA a,MAAM,C;;;MAEvB,OAAO,G;K;IAGX,gC;MAOiB,Q;MAFb,IApmTO,qBAAQ,CAomTf,C;QAAe,OAAO,I;MA CTB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,MA AM,CAAV,C;UAAa,MAAM,C;;;MAEvB,OAAO,G;K;IAGX,gC;MAOiB,Q;MAFb,IA1mTO,qBAAQ,CA0mTf,C; QAAe,OAAO,I;MACTB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL ,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;;MAEvB,OAAO,G;K;IAGX,gC;MAOiB,Q;MAFb,IAhnTO,q BAAQ,CAgnTf,C;QAAe,OAAO,I;MACTB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QA AQ,UAAK,CAAL,C;QACR,IAAI,oBAAM,CAAN,KAAJ,C;UAAa,MAAM,C;;;MAEvB,OAAO,G;K;IAGX,gC;M ASiB,Q;MAFb,IAxnTO,qBAAQ,CAwnTf,C;QAAe,OAAO,I;MACTB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,a AAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,MkB3miBG,MAAO,KIB2miBE,GkB3miBF,EIB2miBO,Ck B3miBP,C;;;MIB6miBd,OAAO,G;K;IAGX,gC;MASiB,Q;MAFb,IAhoTO,qBAAQ,CAgoTf,C;QAAe,OAAO,I;MA CTB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,MkBhniB G,MAAO,KIBgniBE,GkBhniBF,EIBgniBO,CkBhniBP,C;;;MIBkniBd,OAAO,G;K;IAGX,gC;MAOiB,Q;MAFb,IA9 nTO,qBAAQ,CA8nTf,C;QAAe,OAAO,I;MACTB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QA

Q;MAAhB,iD;QAAGB,cAAhB,e;QAAsB,OAAO,OAAP,C;;MAArC,gB;K;oFAGJ,6B;MAMmC,Q;MAAhB,iD;QAAGB,cAAhB,e;QAAsB,OAAO,OAAP,C;;MAArC,gB;K;oFAGJ,6B;MAMmC,Q;MAAhB,iD;QAAGB,cAAhB,e;QAAsB,OAAO,OAAP,C;;MAArC,gB;K;oFAGJ,6B;MAMmC,Q;MAAhB,iD;QAAGB,cAAhB,e;QAAsB,OAAO,OAAP,C;;MAArC,gB;K;oFAGJ,yB;MAAA,oC;MAAA,gC;MAAA,oC;QAMmC,Q;QAAGB,iD;UAAgB,cAAhB,0B;UAAAsB,OAAO,oBAAP,C;;QAArC,gB;O;KANJ,C;gGASA,6B;MARjJiB,gB;MADb,YAAY,C;MACZ,iD;QAAa,WAAb,e;QAAMb,QAAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;MA8jJnB,gB;K;kGAGJ,6B;MAVjJiB,gB;MADb,YAAY,C;MACZ,iD;QAAa,WAAb,e;QAAMb,QAAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;MAGkJnB,gB;K;kGAGJ,6B;MAZjJiB,gB;MADb,YAAY,C;MACZ,iD;QAAa,WAAb,e;QAAMb,QAAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;MAkkJnB,gB;K;kGAGJ,6B;MA3jJiB,gB;MADb,YAAY,C;MACZ,iD;QAAa,WAAb,e;QAAMb,QAAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;MAokJnB,gB;K;kGAGJ,6B;MA7jJiB,gB;MADb,YAAY,C;MACZ,iD;QAAa,WAAb,e;QAAMb,QAAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;MAskJnB,gB;K;kGAGJ,6B;MA/jJiB,gB;MADb,YAAY,C;MACZ,iD;QAAa,WAAb,e;QAAMb,QAAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;MAwkJnB,gB;K;kGAGJ,6B;MAjkJiB,gB;MADb,YAAY,C;MACZ,iD;QAAa,WAAb,e;QAAMb,QAAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;MA0kJnB,gB;K;kGAGJ,6B;MANkJiB,gB;MADb,YAAY,C;MACZ,iD;QAAa,WAAb,e;QAAMb,QAAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;MA4kJnB,gB;K;kGAGJ,yB;MAAA,6B;MAAA,sC;MA5kJA,oC;MAAA,gC;MA4kJA,2BASiB,yB;QArLjJiB,oC;QAAA,gC;eAqLjJiB,0B;UAAA,4B;YAAE,aAAe,c;YA9kJjB,gB;YADb,YAAY,C;YACZ,iD;cAAa,WAAb,0B;cAAmB,QAAO,cAAP,EAAO,sBAAP,WAAgB,iBAAhB,C;;YA8kJmB,W;W;S;OAAzB,C;MATjJiB,oC;QArkJiB,gB;QADb,YAAY,C;QACZ,iD;UAAA,WAAb,0B;UAAmB,QAAO,cAAP,EAAO,sBAAP,WAAgB,iBAAhB,C;;QA8kJnB,gB;O;KATJ,C;kFAYA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAhvUO,qBAAQ,CAgvUf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAAqB,UAAK,CAAL,C;QACJ,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,UAAK,KAAL,CAAvB,C;;QAEIB,OAAO,W;O;KANBX,C;oFAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IA9vUO,qBAAQ,CA8vUf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAAkB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,UAAK,KAAL,CAAvB,C;;QAEIB,OAAO,W;O;KANBX,C;oFAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IA5wUO,qBAAQ,CA4wUf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAAkB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,UAAK,KAAL,CAAvB,C;;QAEIB,OAAO,W;O;KANBX,C;oFAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IA1xUO,qBAAQ,CA0xUf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAAkB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,UAAK,KAAL,CAAvB,C;;QAEIB,OAAO,W;O;KANBX,C;oFAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAxyUO,qBAAQ,CAwyUf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAAkB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,UAAK,KAAL,CAAvB,C;;QAEIB,OAAO,W;O;KANBX,C;oFAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAtzUO,qBAAQ,CAszUf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAAkB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,UAAK,KAAL,CAAvB,C;;QAEIB,OAAO,W;O;KANBX,C;oFAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAp0UO,qBAAQ,CAo0Uf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAAkB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,oBAAU,wBAAV,EAAuB,sBAAK,KAAL,EAAvB,E;;QAEIB,OAAO,W;O;KANBX,C;gGAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IA7UO,qBAAQ,CA7Uf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAAqB,UAAK,CAAL,C;QACJ,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAY,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KANBX,C;kGAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAp8UO,qBAAQ,CAo8Uf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAAkB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAY,EAAiB,WAAjB,EAA8B,U

AAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;kGAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAI9UO,qBAAQ,CAk9Uf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;kGAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAh+UO,qBAAQ,CAG+Uf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;kGAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IA9+UO,qBAAQ,CA8+Uf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;kGAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IA5/UO,qBAAQ,CA4/Uf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;kGAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IA1gVO,qBAAQ,CA0gVf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;kGAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAxhVO,qBAAQ,CAwhVf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;kGAsBA,yB;MAAA,4F;MAAA,8D;MAAA,oC;MAAA,gC;MAAA,uC;QAgBqB,Q;QAHjB,IAtiVO,qBAAQ,CAsiVf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,oBAAU,KAAV,EAAiB,wBAAjB,EAA8B,sBAAK,KAAAL,EAA9B,E;;QAEIB,OAAO,W;O;KAnBX,C;4GAsBA,yB;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IA5nVO,qBAAQ,CA4nVf,C;UACI,OAAO,I;QACX,kBAAqB,UAAK,CAAL,C;QACJ,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;8GAsBA,yB;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IA1oVO,qBAAQ,CA0oVf,C;UACI,OAAO,I;QACX,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;8GAsBA,yB;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAxpVO,qBAAQ,CAwpVf,C;UACI,OAAO,I;QACX,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;8GAsBA,yB;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAtqVO,qBAAQ,CAsqVf,C;UACI,OAAO,I;QACX,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;8GAsBA,yB;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAIsVO,qBAAQ,CAksVf,C;UACI,OAAO,I;QACX,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;8GAsBA,yB;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAhtVO,qBAAQ,CAgtVf,C;UACI,OAAO,I;QACX,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;8GAsBA,yB;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IA9tVO,qBAAQ,CA8tVf,C;UACI,OAAO,I;QACX,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAnBX,C;8GAsBA,yB;MAAA,8D;MAAA,oC;MAAA,gC;MAAA,uC;QAgBqB,Q;QAHjB,IA5uVO,qBAAQ,CA4uVf,C;UACI,OAAO,I;QACX,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,oBAAU,KAAV,EAAiB,wBAAjB,EAA8B,sBAAK,KAAAL,EAA9B,E;;QAEIB,OAAO,W;O;KAnBX,C;8FAsBA,yB;MAAA,8D;MAAA,uC;QAIbqB,Q;QAHjB,IAN0VO,qBAAQ,CAM0Vf,C;UACI,OAAO,I;QACX,kBAAqB,UAAK,CAAL,C;QACJ,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KApBX,C;gGAuBA,yB;MAAA,8D;MAAA,uC;QAIbqB,Q;QAHjB,IAI1VO,qBAAQ,CAk1Vf,C;UACI,OAAO,I;QACX,kBAakB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,UAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KApBX

AJ,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,oBAAU,KAAV,EAAiB,sBAAI,KAAJ,EAAjB,EAA6B,wBAA
7B,E;UACd,qB;;QAEJ,OAAO,W;O;KApBX,C;wGAuBA,yB;MAAA,8D;MAAA,uC;QAgB6B,UAE0,M;QAJhC,
YAAy,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAaQb,UAAI,YAAJ,EAAI,oBAAJ,O;QACrB
,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,cAAJ,EAAI,sBAAJ,SAAV,EAAwB,WAAxB,C;;QAEIB,OA
AO,W;O;KApBX,C;0GAuBA,yB;MAAA,8D;MAAA,uC;QAgB0B,UAEU,M;QAJhC,YAAy,wB;QACZ,IAAI,QA
AQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAakB,UAAI,YAAJ,EAAI,oBAAJ,O;QACIB,OAAO,SAAS,CAAhB,C;U
ACI,cAAc,UAAU,UAAI,cAAJ,EAAI,sBAAJ,SAAV,EAAwB,WAAxB,C;;QAEIB,OAAO,W;O;KApBX,C;0GAuB
A,yB;MAAA,8D;MAAA,uC;QAgB0B,UAEU,M;QAJhC,YAAy,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,
I;QACtB,kBAakB,UAAI,YAAJ,EAAI,oBAAJ,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,cA
AJ,EAAI,sBAAJ,SAAV,EAAwB,WAAxB,C;;QAEIB,OAAO,W;O;KApBX,C;0GAuBA,yB;MAAA,8D;MAAA,uC
;QAgB0B,UAEU,M;QAJhC,YAAy,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAakB,UAAI,Y
AAJ,EAAI,oBAAJ,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,cAAJ,EAAI,sBAAJ,SAAV,EA
AwB,WAAxB,C;;QAEIB,OAAO,W;O;KApBX,C;0GAuBA,yB;MAAA,8D;MAAA,uC;QAgB0B,UAEU,M;QAJh
C,YAAy,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAakB,UAAI,YAAJ,EAAI,oBAAJ,O;QAC
IB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,cAAJ,EAAI,sBAAJ,SAAV,EAAwB,WAAxB,C;;QAEIB,O
AAO,W;O;KApBX,C;0GAuBA,yB;MAAA,8D;MAAA,uC;QAgB0B,UAEU,M;QAJhC,YAAy,wB;QACZ,IAAI,Q
AAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAakB,UAAI,YAAJ,EAAI,oBAAJ,O;QACIB,OAAO,SAAS,CAAhB,C;
UACI,cAAc,UAAU,UAAI,cAAJ,EAAI,sBAAJ,SAAV,EAAwB,WAAxB,C;;QAEIB,OAAO,W;O;KApBX,C;0GAu
BA,yB;MAAA,8D;MAAA,uC;QAgB0B,UAEU,M;QAJhC,YAAy,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,OAA
O,I;QACtB,kBAakB,UAAI,YAAJ,EAAI,oBAAJ,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,c
AAJ,EAAI,sBAAJ,SAAV,EAAwB,WAAxB,C;;QAEIB,OAAO,W;O;KApBX,C;0GAuBA,yB;MAAA,8D;MAAA,
uC;QAgB0B,UAEU,M;QAJhC,YAAy,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAakB,UAAI
,YAAJ,EAAI,oBAAJ,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,cAAJ,EAAI,sBAAJ,SAAV,E
AAwB,WAAxB,C;;QAEIB,OAAO,W;O;KApBX,C;0GAuBA,yB;MAAA,8D;MAAA,oC;MAAA,gC;MAAA,uC;Q
AgB0B,UAEU,M;QAJhC,YAAy,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAakB,UAAI,YA
AJ,EAAI,oBAAJ,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,oBAAU,sBAAI,cAAJ,EAAI,sBAAJ,UAAV,EA
AwB,wBAAxB,E;;QAEIB,OAAO,W;O;KApBX,C;4FAuBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAgBoB,Q;Q
AHhB,IAP0XO,qBAAQ,CAo0Xf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,
C;QAA+B,8B;QAA5C,aiBj9mBO,W;QjBk9mBP,kBAakB,O;QACIB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAh
B,M;UACI,cAAc,UAAU,WAAV,EAAuB,OAavB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KApB
X,C;8FAuBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAIBoB,Q;QAHhB,IAP1XO,qBAAQ,CAo1Xf,C;UAAe,OA
AO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,aiBz+mBO,W;QjB0+m
BP,kBAakB,O;QACIB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,cAAc,UAAU,WAAV,EAAuB,OA
AvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KArBX,C;8FAwBA,yB;MAAA,gD;MAAA,gE;MAA
A,gD;QAIBoB,Q;QAHhB,IAP2XO,qBAAQ,CAo2Xf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mB
AAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,aiBjgnBO,W;QjBkgnBP,kBAakB,O;QACIB,wBAAgB,SAAhB,gB;UA
AgB,cAAA,SAAhB,M;UACI,cAAc,UAAU,WAAV,EAAuB,OAavB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,O
AAO,M;O;KArBX,C;8FAwBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAIBoB,Q;QAHhB,IAP3XO,qBAAQ,CAo
3Xf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,aiBzh
nBO,W;QjB0hnBP,kBAakB,O;QACIB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,cAAc,UAAU,WA
AV,EAAuB,OAavB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KArBX,C;8FAwBA,yB;MAAA,gD;
MAAA,gE;MAAA,gD;QAIBoB,Q;QAHhB,IAP4XO,qBAAQ,CAo4Xf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,k
BAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,aiBjgnBO,W;QjBkgnBP,kBAakB,O;QACIB,wBAAg
B,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,cAAc,UAAU,WAAV,EAAuB,OAavB,C;UACd,MAAO,WAAI,W
AAJ,C;;QAEX,OAAO,M;O;KArBX,C;8FAwBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAIBoB,Q;QAHhB,IAP5
XO,qBAAQ,CAo5Xf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8
B;QAA5C,aiBzknBO,W;QjB0knBP,kBAakB,O;QACIB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,c
AAc,UAAU,WAAV,EAAuB,OAavB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KArBX,C;8FAwBA

,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAIBoB,Q;QAHhB,IAp6XO,qBAAQ,CAo6Xf,C;UAAe,OAAO,OAAO,OAAO,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,aiBjmnBO,W;QjBkmmBP,kBAAkB,O;QACIB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KArBX,C;8FAwBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAIBoB,Q;QAHhB,IAp7XO,qBAAQ,CAo7Xf,C;UAAe,OAAO,OAAO,OAAO,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,aiBzmnBO,W;QjB0nnBP,kBAAkB,O;QACIB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KArBX,C;8FAwBA,yB;MAAA,gD;MAAA,gE;MAAA,oC;MAAA,gC;MAAA,gD;QAIBoB,Q;QAHhB,IAp8XO,qBAAQ,CAo8Xf,C;UAAe,OAAO,OAAO,OAAO,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,aiBjpnBO,W;QjBkpnBP,kBAAkB,O;QACIB,wBAAgB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UACI,cAAc,UAAU,WAAV,EAAuB,oBAAvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KArBX,C;OGAwBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAcI,IA5hYO,qBAAQ,CA4hYf,C;UAAe,OAAO,OAAO,OAAO,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,aiBzqnBO,W;QjB0qnBP,kBAAkB,O;QACIB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KArBX,C;4GAWBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAEI,IA7iYO,qBAAQ,CA6iYf,C;UAAe,OAAO,OAAO,OAAO,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,aiBlnBO,W;QjBmsnBP,kBAAkB,O;QACIB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KATBX,C;4GAYBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAEI,IA9jYO,qBAAQ,CA8jYf,C;UAAe,OAAO,OAAO,OAAO,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,aiB3tnBO,W;QjB4tnBP,kBAAkB,O;QACIB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KATBX,C;4GAYBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAEI,IA/kYO,qBAAQ,CA+kYf,C;UAAe,OAAO,OAAO,OAAO,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,aiBpvnBO,W;QjBqvnBP,kBAAkB,O;QACIB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KATBX,C;4GAYBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAEI,IAhmYO,qBAAQ,CAgmYf,C;UAAe,OAAO,OAAO,OAAO,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,aiB7wnBO,W;QjB8wnBP,kBAAkB,O;QACIB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KATBX,C;4GAYBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAEI,IAjnYO,qBAAQ,CAinYf,C;UAAe,OAAO,OAAO,OAAO,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,aiBtynBO,W;QjBuynBP,kBAAkB,O;QACIB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KATBX,C;4GAYBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAEI,IAloYO,qBAAQ,CAkoYf,C;UAAe,OAAO,OAAO,OAAO,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,aiB/znBO,W;QjBg0nBP,kBAAkB,O;QACIB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KATBX,C;4GAYBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAEI,IANpYO,qBAAQ,CAnpYf,C;UAAe,OAAO,OAAO,OAAO,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,aiBx1nBO,W;QjBy1nBP,kBAAkB,O;QACIB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KATBX,C;4GAYBA,yB;MAAA,gD;MAAA,gE;MAAA,oC;MAAA,gD;QAEI,IApqYO,qBAAQ,CAoqYf,C;UAAe,OAAO,OAAO,OAAO,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,aiBj3nBO,W;QjBk3nBP,kBAAkB,O;QACIB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,EAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KATBX,C;gGayBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAcI,IA5vYO,qBAAQ,CA4vYf,C;UAAe,OAAO,W;QACtB,sBAaqB,UAAK,CAAL,CAArB,C;QACgC,kBAAnB,eAAa,gBAAb,C;QAA2B,sBAAL,aAAJ,C;QAAxC,aiB14nBO,W;QjB24nBP,iBAAC,CAAd,UAAsB,gBAAtB,U;UACI,gBAAC,UAAU,aAAV,EAAuB,UAAK,KAAL,CAAvB,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KArBX,C;kGawBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAWI,IAzwYO,qBAAQ,CaywYf,C;UAAe,OAAO,W;QACtB,sBAakB,UAAK,CAAL,CAAIB,C;QACmC,kBAAtB,eAAgB,gBAAhB,C;QAA8B,sBAAI,aAAJ,C;QAA3C,aiB/5nBO,W;QjBg6nBP,iBAAC,CAAd,UAAsB,gBAAtB,U;UACI,gBAAC,UAAU,aAAV,EAAuB,UAAK,KAAL,CAAvB,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAlB

X,C;kGAqBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAWI,IAtxYO,qBAAQ,CAsxYf,C;UAAe,OAAO,W;QACtB ,sBAakB,UAAK,CAAL,CAAIB,C;QACoC,kBAAvB,eAAiB,gBAAjB,C;QAA+B,sBAAI,aAAJ,C;QAA5C,aiBp7n BO,W;QjBq7nBP,iBAAc,CAAd,UAAAsB,gBAAtB,U;UACI,gBAAc,UAAU,aAAV,EAAuB,UAAK,KAAL,CAAvB ,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAIBX,C;kGAqBA,yB;MAAA,qD;MAAA,gE;MAAA,uC; QAWI,IAAnyYO,qBAAQ,CAmyYf,C;UAAe,OAAO,W;QACtB,sBAakB,UAAK,CAAL,CAAIB,C;QACkC,kBAAr B,eAAe,gBAaf,C;QAA6B,sBAAI,aAAJ,C;QAA1C,aiBz8nBO,W;QjB08nBP,iBAAc,CAAd,UAAAsB,gBAAtB,U;U ACI,gBAAc,UAAU,aAAV,EAAuB,UAAK,KAAL,CAAvB,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O; KAIBX,C;kGAqBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAWI,IAhzYO,qBAAQ,CAgzYf,C;UAAe,OAAO,W; QACtB,sBAakB,UAAK,CAAL,CAAIB,C;QACmC,kBAAtB,eAAgB,gBAAhB,C;QAA8B,sBAAI,aAAJ,C;QAA3 C,aiB99nBO,W;QjB+9nBP,iBAAc,CAAd,UAAAsB,gBAAtB,U;UACI,gBAAc,UAAU,aAAV,EAAuB,UAAK,KAA L,CAAvB,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAIBX,C;kGAqBA,yB;MAAA,qD;MAAA,gE;M AAA,uC;QAWI,IA7zYO,qBAAQ,CA6zYf,C;UAAe,OAAO,W;QACtB,sBAakB,UAAK,CAAL,CAAIB,C;QACoC ,kBAAvB,eAAiB,gBAAjB,C;QAA+B,sBAAI,aAAJ,C;QAA5C,aiBn/nBO,W;QjBo/nBP,iBAAc,CAAd,UAAAsB,gB AAtB,U;UACI,gBAAc,UAAU,aAAV,EAAuB,UAAK,KAAL,CAAvB,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,O AAO,M;O;KAIBX,C;kGAqBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAWI,IA10YO,qBAAQ,CA00Yf,C;UAAe, OAAO,W;QACtB,sBAakB,UAAK,CAAL,CAAIB,C;QACqC,kBAAxB,eAAkB,gBAaIB,C;QAAgC,sBAAI,aAAJ, C;QAA7C,aiBxgoBO,W;QjBygoBP,iBAAc,CAAd,UAAAsB,gBAAtB,U;UACI,gBAAc,UAAU,aAAV,EAAuB,UAA K,KAAL,CAAvB,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAIBX,C;kGAqBA,yB;MAAA,qD;MAA A,gE;MAAA,uC;QAWI,IAv1YO,qBAAQ,CAu1Yf,C;UAAe,OAAO,W;QACtB,sBAakB,UAAK,CAAL,CAAIB,C ;QACsC,kBAAzB,eAAmB,gBAAnB,C;QAAiC,sBAAI,aAAJ,C;QAA9C,aiB7hoBO,W;QjB8hoBP,iBAAc,CAAd,U AAsB,gBAAtB,U;UACI,gBAAc,UAAU,aAAV,EAAuB,UAAK,KAAL,CAAvB,C;UACd,MAAO,WAAI,aAAJ,C;; QAEX,OAAO,M;O;KAIBX,C;kGAqBA,yB;MAAA,qD;MAAA,gE;MAAA,oC;MAAA,gC;MAAA,uC;QAWI,IAp 2YO,qBAAQ,CAo2Yf,C;UAAe,OAAO,W;QACtB,sBAakB,UAAK,CAAL,CAAIB,C;QACmC,kBAAtB,eAAgB,g BAAhB,C;QAA8B,sBAAI,0BAAJ,C;QAA3C,aiBljoBO,W;QjBmjoBP,iBAAc,CAAd,UAAAsB,gBAAtB,U;UACI,g BAAc,oBAAU,0BAAV,EAAuB,sBAAK,KAAL,EAAvB,E;UACd,MAAO,WAAI,0BAAJ,C;;QAEX,OAAO,M;O; KAIBX,C;8GAqBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QACI,IA57YO,qBAAQ,CA47Yf,C;UAAe,OAAO,W;Q ACtB,sBAaqB,UAAK,CAAL,CAArB,C;QACgC,kBAAnB,eAAa,gBAAb,C;QAA2B,sBAAI,aAAJ,C;QAAxC,aiB lkoBO,W;QjB2koBP,iBAAc,CAAd,UAAAsB,gBAAtB,U;UACI,gBAAc,UAAU,KAHV,EAAiB,aAAjB,EAA8B,UA AK,KAAL,CAA9B,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KArBX,C;gHawBA,yB;MAAA,qD;M AAA,gE;MAAA,uC;QAYI,IA18YO,qBAAQ,CA08Yf,C;UAAe,OAAO,W;QACtB,sBAakB,UAAK,CAAL,CAA I B,C;QACmC,kBAAtB,eAAgB,gBAAhB,C;QAA8B,sBAAI,aAAJ,C;QAA3C,aiBhmoBO,W;QjBimoBP,iBAAc,CA Ad,UAAAsB,gBAAtB,U;UACI,gBAAc,UAAU,KAHV,EAAiB,aAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,M AAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAnBX,C;gHAsBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAYI,IAx9 YO,qBAAQ,CAw9Yf,C;UAAe,OAAO,W;QACtB,sBAakB,UAAK,CAAL,CAAIB,C;QACoC,kBAAvB,eAAiB,gB AAjB,C;QAA+B,sBAAI,aAAJ,C;QAA5C,aiBtnoBO,W;QjBunoBP,iBAAc,CAAd,UAAAsB,gBAAtB,U;UACI,gBA Ac,UAAU,KAHV,EAAiB,aAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAA O,M;O;KAnBX,C;gHAsBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAYI,IA+YO,qBAAQ,CAs+Yf,C;UAAe,OA AO,W;QACtB,sBAakB,UAAK,CAAL,CAAIB,C;QACkC,kBAArB,eAAe,gBAaf,C;QAA6B,sBAAI,aAAJ,C;QA A1C,aiB5ooBO,W;QjB6ooBP,iBAAc,CAAd,UAAAsB,gBAAtB,U;UACI,gBAAc,UAAU,KAHV,EAAiB,aAAjB,EA A8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAnBX,C;gHAsBA,yB;MAA A,qD;MAAA,gE;MAAA,uC;QAYI,IAp/YO,qBAAQ,CAo/Yf,C;UAAe,OAAO,W;QACtB,sBAakB,UAAK,CAAL, CAAIB,C;QACmC,kBAAtB,eAAgB,gBAAhB,C;QAA8B,sBAAI,aAAJ,C;QAA3C,aiBlqoBO,W;QjBmqoBP,iBAA c,CAAd,UAAAsB,gBAAtB,U;UACI,gBAAc,UAAU,KAHV,EAAiB,aAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UAC d,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAnBX,C;gHAsBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAYI,I AlgZO,qBAAQ,CAkgZf,C;UAAe,OAAO,W;QACtB,sBAakB,UAAK,CAAL,CAAIB,C;QACoC,kBAAvB,eAAiB, gBAAjB,C;QAA+B,sBAAI,aAAJ,C;QAA5C,aiBxroBO,W;QjByroBP,iBAAc,CAAd,UAAAsB,gBAAtB,U;UACI,gB AAc,UAAU,KAHV,EAAiB,aAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OA AO,M;O;KAnBX,C;gHAsBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAYI,IAhhZO,qBAAQ,CAghZf,C;UAAe,O

AAO,W;QACtB,sBAAkB,UAAK,CAAL,CAAIB,C;QACqC,kBAAXB,eAAkB,gBAAIB,C;QAAgC,sBAAI,aAAJ,C;QAA7C,aiB9soBO,W;QjB+soBP,iBAAC,CAAd,UAAAsB,gBAAtB,U;UACI,gBAAC,UAAU,KAAV,EAAiB,aAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAnBX,C;gHAsBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAYI,IA9hZO,qBAAQ,CA8hZf,C;UAAe,OAAO,W;QACtB,sBAAkB,UAAK,C AAL,CAAIB,C;QACsC,kBAAZB,eAAmB,gBAAnB,C;QAAiC,sBAAI,aAAJ,C;QAA9C,aiBpuoBO,W;QjBquoBP,iBAAC,CAAd,UAAAsB,gBAAtB,U;UACI,gBAAC,UAAU,KAAV,EAAiB,aAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAnBX,C;gHAsBA,yB;MAAA,qD;MAAA,gE;MAAA,oC;MAAA,gC;MAAA,uC;QAYI,IA5iZO,qBAAQ,CA4iZf,C;UAAe,OAAO,W;QACtB,sBAAkB,UAAK,CAAL,CAAIB,C;QACmC,kBAAtB,eAAgB,gBAAhB,C;QAA8B,sBAAI,0BAAJ,C;QAA3C,aiB1voBO,W;QjB2voBP,iBAAC,CAAd,UAAAsB,gBAAtB,U;UACI,gBAAC,oBAAU,KAAV,EAAiB,0BAAjB,EAA8B,sBAAK,KAAL,EAA9B,E;UACd,MAAO,WAAI,0BAAJ,C;;QAEX,OAAO,M;O;KAnBX,C;8EAsBA,yB;MA/zBA,gD;MAAA,gE;MA+zBA,gD;QAcW,sB;;UA7zBS,Q;UAHhB,IAp0XO,qBAAQ,CAo0Xf,C;YAAe,qBAAO,OAgoBH,OAhoBG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBA+zBzB,OA/zByB,C;UAA5C,aiBj9mBO,W;UjBk9mBP,kBA8zBmB,O;UA7zBnB,iD;YAAgB,cAAhB,e;YACI,cA4zBwB,SA5zBV,CAAU,WAAV,EAAuB,OAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QAYzBP,yB;O;KADJ,C;gFAiBA,yB;MAzzBA,gD;MAAA,gE;MAyzBA,gD;QAEW,sB;;UA7zBS,Q;UAHhB,IAp1XO,qBAAQ,CAo1Xf,C;YAAe,qBAAO,OAo0zBH,OA1zBG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBAyzBzB,OAzzByB,C;UAA5C,aiBz+mBO,W;UjB0+mBP,kBAwzBmB,O;UA7zBnB,iD;YAAgB,cAAhB,e;YACI,cAszBwB,SAtzBV,CAAU,WAAV,EAAuB,OAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QAmzBP,yB;O;KAFJ,C;gFAkBA,yB;MANzBA,gD;MAAA,gE;MAmzBA,gD;QAEW,sB;;UA7zBS,Q;UAHhB,IAp2XO,qBAAQ,CAo2Xf,C;YAAe,qBAAO,OAozBH,OApozBG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBAmzBzB,OAanzByB,C;UAA5C,aiBjgnBO,W;UjBkgnBP,kBAkzBmB,O;UA7zBnB,iD;YAAgB,cAAhB,e;YACI,cAgzBwB,SAhzBV,CAAU,WAAV,EAAuB,OAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QA6yBP,yB;O;KAFJ,C;gFAkBA,yB;MA7yBA,gD;MAAA,gE;MA6yBA,gD;QAEW,sB;;UA3yBS,Q;UAHhB,IAp3XO,qBAAQ,CAo3Xf,C;YAAe,qBAAO,OA8yBH,OA9yBG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBA6yBzB,OA7yByB,C;UAA5C,aiBzhnBO,W;UjB0hnBP,kBA4yBmB,O;UA3yBnB,iD;YAAgB,cAAhB,e;YACI,cA0yBwB,SA1yBV,CAAU,WAAV,EAAuB,OAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QAuyBP,yB;O;KAFJ,C;gFAkBA,yB;MAvyBA,gD;MAAA,gE;MAuyBA,gD;QAEW,sB;;UAryBS,Q;UAHhB,IAp4XO,qBAAQ,CAo4Xf,C;YAAe,qBAAO,OAwyBH,OAxyBG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBAuyBzB,OAvyByB,C;UAA5C,aiBjnnBO,W;UjBknnBP,kBA5yBmB,O;UAryBnB,iD;YAAgB,cAAhB,e;YACI,cAoyBwB,SApyBV,CAAU,WAAV,EAAuB,OAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QAiyBP,yB;O;KAFJ,C;gFAkBA,yB;MAjyBA,gD;MAAA,gE;MAiyBA,gD;QAEW,sB;;UA/xBS,Q;UAHhB,IAp5XO,qBAAQ,CAo5Xf,C;YAAe,qBAAO,OAkyBH,OAlyBG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBAiyBzB,OAjyByB,C;UAA5C,aiBzknBO,W;UjB0knBP,kBAgyBmB,O;UA/xBnB,iD;YAAgB,cAAhB,e;YACI,cA8xBwB,SA9xBV,CAAU,WAAV,EAAuB,OAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QA2xBP,yB;O;KAFJ,C;gFAkBA,yB;MA3xBA,gD;MAAA,gE;MA2xBA,gD;QAEW,sB;;UAzxBS,Q;UAHhB,IAp6XO,qBAAQ,CAo6Xf,C;YAAe,qBAAO,OA4xBH,OA5xBG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBA2xBzB,OA3xByB,C;UAA5C,aiBjmnBO,W;UjBkmnBP,kBA0xBmB,O;UAzxBnB,iD;YAAgB,cAAhB,e;YACI,cAwxBwB,SAxxBV,CAAU,WAAV,EAAuB,OAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QAqxBP,yB;O;KAFJ,C;gFAkBA,yB;MARxBA,gD;MAAA,gE;MAqxBA,gD;QAEW,sB;;UANxBS,Q;UAHhB,IAp7XO,qBAAQ,CAo7Xf,C;YAAe,qBAAO,OA5xBH,OA6xBG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBAqxzBzB,OA6xByB,C;UAA5C,aiBznnBO,W;UjB0nnBP,kBAoxBmB,O;UANxBnB,iD;YAAgB,cAAhB,e;YACI,cAkxBwB,SA1xBV,CAAU,WAAV,EAAuB,OAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QA+wBP,yB;O;KAFJ,C;gFAkBA,yB;MA/wBA,gD;MAAA,gE;MAAA,oC;MAAA,gC;MAA+wBA,gD;QAEW,sB;;UA7wBS,Q;UAHhB,IAp8XO,qBAAQ,CAo8Xf,C;YAAe,qBAAO,OA6xBH,OA7xBG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBA+wBzB,OA/wByB,C;UAA5C,aiBjpnBO,W;UjBkpnBP,kBA8wBmB,O;UA7wBnB,iD;YAAgB,cAAhB,0B;YACI,cA4wBwB,SA5wBV,CAAU,WAAV,EAAuB,oBAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QAYwBP,yB;O;KAFJ,C;4FAkBA,yB;MA

UAAgB,cAAA,SAAhB,M;UACI,cAAO,SAAS,OAAT,CAAP,C;;QAEJ,OAAO,G;O;KAbX,C;mFagBA,yB;MAA
A,SASoB,gB;MATpB,sC;QUoB,Q;QADhB,Y;QACA,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,c
AAO,SAAS,OAAT,CAAP,C;;QAEJ,OAAO,G;O;KAbX,C;mFagBA,yB;MAAA,SASoB,gB;MATpB,sC;QUoB,
Q;QADhB,Y;QACA,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,cAAO,SAAS,OAAT,CAAP,C;;QAEJ
,OAAO,G;O;KAbX,C;mFagBA,yB;MAAA,SASoB,gB;MATpB,sC;QUoB,Q;QADhB,Y;QACA,wBAAgB,SA
AhB,gB;UAAgB,cAAA,SAAhB,M;UACI,cAAO,SAAS,OAAT,CAAP,C;;QAEJ,OAAO,G;O;KAbX,C;mFagBA,yB;
MAAA,SASoB,gB;MATpB,oC;MAAA,gC;MAAA,sC;QUoB,Q;QADhB,Y;QACA,wBAAgB,SAAhB,gB;UAAg
B,cAAhB,UAAgB,SAAhB,O;UACI,cAAO,SAAS,oBAAT,CAAP,C;;QAEJ,OAAO,G;O;KAbX,C;mFagBA,yB;M
GI7pBA,6B;MHk7pBA,sC;QAWoB,Q;QADhB,UGI7pBmC,cHk7pBnB,CGI7pBmB,C;QHm7pBnB,wBAAgB,SA
AhB,gB;UAAgB,cAAA,SAAhB,M;UACI,MGtvqBiD,cHsvqBjD,GGtvqB2D,KAAK,GHsvqBzD,SAAS,OAAT,CG
tvqBoE,KAAK,IAAf,C;;QHwvqBrD,OAAO,G;O;KAdX,C;mFAiBA,yB;MGn8pBA,6B;MHm8pBA,sC;QAWoB,Q
;QADhB,UGn8pBmC,cHm8pBnB,CGn8pBmB,C;QHo8pBnB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;U
ACI,MGvwqBiD,cHuwqBjD,GGvwqB2D,KAAK,GHuwqBzD,SAAS,OAAT,CGvwqBoE,KAAK,IAAf,C;;QH
ywqBrD,OAAO,G;O;KAdX,C;mFAiBA,yB;MGp9pBA,6B;MHo9pBA,sC;QAWoB,Q;QADhB,UGp9pBmC,cHo9pBn
B,CGp9pBmB,C;QHq9pBnB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,MGxxqBiD,cHwxqBjD,GGx
xqB2D,KAAK,GHwxqBzD,SAAS,OAAT,CGxxqBoE,KAAK,IAAf,C;;QH0xqBrD,OAAO,G;O;KAdX,C;mFAiBA
,yB;MGr+pBA,6B;MHq+pBA,sC;QAWoB,Q;QADhB,UGr+pBmC,cHq+pBnB,CGr+pBmB,C;QHs+pBnB,wBAAg
B,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,MGzyqBiD,cHyyqBjD,GGzyqB2D,KAAK,GHyyqBzD,SAAS,OA
AT,CGzyqBoE,KAAK,IAAf,C;;QH2yqBrD,OAAO,G;O;KAdX,C;mFAiBA,yB;MGt/pBA,6B;MHs/pBA,sC;QAW
oB,Q;QADhB,UGt/pBmC,cHs/pBnB,CGt/pBmB,C;QHs/pBnB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;U
ACI,MG1zqBiD,cH0zqBjD,GG1zqB2D,KAAK,GH0zqBzD,SAAS,OAAT,CG1zqBoE,KAAK,IAAf,C;;QH4zqBrD
,OAAO,G;O;KAdX,C;mFAiBA,yB;MGvgqBA,6B;MHugqBA,sC;QAWoB,Q;QADhB,UGvgqBmC,cHugqBnB,CG
vgqBmB,C;QHwgqBnB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,MG30qBiD,cH20qBjD,GG30qB2
D,KAAK,GH20qBzD,SAAS,OAAT,CG30qBoE,KAAK,IAAf,C;;QH60qBrD,OAAO,G;O;KAdX,C;mFAiBA,yB;
MGxhqBA,6B;MHwhqBA,sC;QAWoB,Q;QADhB,UGxhqBmC,cHwhqBnB,CGxhqBmB,C;QHhyqBnB,wBAAgB,
SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,MG51qBiD,cH41qBjD,GG51qB2D,KAAK,GH41qBzD,SAAS,OAA
T,CG51qBoE,KAAK,IAAf,C;;QH81qBrD,OAAO,G;O;KAdX,C;mFAiBA,yB;MGziqBA,6B;MHyiqBA,sC;QAWo
B,Q;QADhB,UGziqBmC,cHyiqBnB,CGziqBmB,C;QH0iqBnB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;U
ACI,MG72qBiD,cH62qBjD,GG72qB2D,KAAK,GH62qBzD,SAAS,OAAT,CG72qBoE,KAAK,IAAf,C;;QH+2qBr
D,OAAO,G;O;KAdX,C;mFAiBA,yB;MAAA,oC;MAAA,gC;MG1jqBA,6B;MH0jqBA,sC;QAWoB,Q;QADhB,UG
1jqBmC,cH0jqBnB,CG1jqBmB,C;QH2jqBnB,wBAAgB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UACI,M
G93qBiD,cH83qBjD,GG93qB2D,KAAK,GH83qBzD,SAAS,oBAAT,CG93qBoE,KAAK,IAAf,C;;QHg4qBrD,OA
AO,G;O;KAdX,C;mFAiBA,yB;MmBxkqBA,+B;MnBwkqBA,sC;QAWoB,Q;QADhB,UmBvkqBqC,eAAW,oBnBu
kqB/B,CmBvkqB+B,CAAX,C;QnBwkqBrC,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,MmB54qBm
D,enB44qBnD,GmB54qB8D,KAAK,KnB44qB5D,SAAS,OAAT,CmB54qBuE,KAAK,CAAhB,C;;QnB84qBvD,O
AAO,G;O;KAdX,C;mFAiBA,yB;MmBzLqBA,+B;MnBylqBA,sC;QAWoB,Q;QADhB,UmBxlqBqC,eAAW,oBnBw
lqB/B,CmBxlqB+B,CAAX,C;QnBylqBrC,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,MmB75qBmD,
enB65qBnD,GmB75qB8D,KAAK,KnB65qB5D,SAAS,OAAT,CmB75qBuE,KAAK,CAAhB,C;;QnB+5qBvD,OA
AO,G;O;KAdX,C;mFAiBA,yB;MmB1mqBA,+B;MnB0mqBA,sC;QAWoB,Q;QADhB,UmBzmqBqC,eAAW,oBn
BmqB/B,CmBzmqB+B,CAAX,C;QnB0mqBrC,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,MmB96q
BmD,enB86qBnD,GmB96qB8D,KAAK,KnB86qB5D,SAAS,OAAT,CmB96qBuE,KAAK,CAAhB,C;;QnBg7qBvD,
OAAO,G;O;KAdX,C;kFAiBA,yB;MmB3nqBA,+B;MnB2nqBA,sC;QAWoB,Q;QADhB,UmB1nqBqC,eAAW,oBn
B0nqB/B,CmB1nqB+B,CAAX,C;QnB2nqBrC,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,MmB/7qB
mD,enB+7qBnD,GmB/7qB8D,KAAK,KnB+7qB5D,SAAS,OAAT,CmB/7qBuE,KAAK,CAAhB,C;;QnBi8qBvD,O
AAO,G;O;KAdX,C;mFAiBA,yB;MmB5oqBA,+B;MnB4oqBA,sC;QAWoB,Q;QADhB,UmB3oqBqC,eAAW,oBnB
2oqB/B,CmB3oqB+B,CAAX,C;QnB4oqBrC,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,MmBh9qBm
D,enBg9qBnD,GmBh9qB8D,KAAK,KnBg9qB5D,SAAS,OAAT,CmBh9qBuE,KAAK,CAAhB,C;;QnBk9qBvD,O
AAO,G;O;KAdX,C;mFAiBA,yB;MmB7pqBA,+B;MnB6pqBA,sC;QAWoB,Q;QADhB,UmB5pqBqC,eAAW,oBnB

4pqB/B,CmB5pqB+B,CAAX,C;QnB6pqBrC,wBAAGB,SAAhB,gB;UAGB,cAAA,SAAhB,M;UACI,MmBj+qBm
D,enBi+qBnD,GmBj+qB8D,KAAK,KnBi+qB5D,SAAS,OAAT,CmBj+qBuE,KAAK,CAAhB,C;;QnBm+qBvD,OA
AO,G;O;KAdX,C;mFAiBA,yB;MmB9qqBA,+B;MnB8qqBA,sC;QAWoB,Q;QADhB,UmB7qqBqC,eAAW,oBnB6
qqB/B,CmB7qqB+B,CAAX,C;QnB8qqBrC,wBAAGB,SAAhB,gB;UAGB,cAAA,SAAhB,M;UACI,MmBl/qBmD,
enBk/qBnD,GmBl/qB8D,KAAK,KnBk/qB5D,SAAS,OAAT,CmBl/qBuE,KAAK,CAAhB,C;;QnBo/qBvD,OAAO,
G;O;KAdX,C;kFAiBA,yB;MmB/rqBA,+B;MnB+rqBA,sC;QAWoB,Q;QADhB,UmB9rqBqC,eAAW,oBnB8rqB/B,
CmB9rqB+B,CAAX,C;QnB+rqBrC,wBAAGB,SAAhB,gB;UAGB,cAAA,SAAhB,M;UACI,MmBngrBmD,enBmg
rBnD,GmBngrB8D,KAAK,KnBmgrB5D,SAAS,OAAT,CmBngrBuE,KAAK,CAAhB,C;;QnBqgrBvD,OAAO,G;O;
KAdX,C;mFAiBA,yB;MAAA,oC;MAAA,gC;MmBhtqBA,+B;MnBgtqBA,sC;QAWoB,Q;QADhB,UmB/sqBqC,e
AAW,oBnB+sqB/B,CmB/sqB+B,CAAX,C;QnBgtqBrC,wBAAGB,SAAhB,gB;UAGB,cAAhB,UAGB,SAAhB,O
;UACI,MmBphrBmD,enBohrBnD,GmBphrB8D,KAAK,KnBohrB5D,SAAS,oBAAT,CmBphrBuE,KAAK,CAAhB,
C;;QnBshrBvD,OAAO,G;O;KAdX,C;IAiBA,mC;MAIoB,UAMT,M;MANP,wBAAGB,SAAhB,gB;QAAGB,cAAA,
SAAhB,M;QACI,IAAI,eAAJ,C;UACI,MAAM,gCAAYB,2BAAwB,SAAxB,MAAZB,C;;;MAId,OAAO,0D;K;wFA
GX,yB;MAAA,+D;MAAA,6B;MAAA,uC;QAUoB,Q;QAFhB,YAAY,gB;QACZ,aAAa,gB;QACb,wBAAGB,SAAh
B,gB;UAGB,cAAA,SAAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,KAAM,WAAI,OAAJ,C;;YAEN,MAA
O,WAAI,OAAJ,C;;;QAGf,OAAO,cAAK,KAAL,EAAY,MAAZ,C;O;KAjBX,C;0FAoBA,yB;MAAA,+D;MAAA,6
B;MAAA,uC;QAUoB,Q;QAFhB,YAAY,gB;QACZ,aAAa,gB;QACb,wBAAGB,SAAhB,gB;UAGB,cAAA,SAAh
B,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,KAAM,WAAI,OAAJ,C;;YAEN,MAAO,WAAI,OAAJ,C;;;QAGf,
OAAO,cAAK,KAAL,EAAY,MAAZ,C;O;KAjBX,C;0FAoBA,yB;MAAA,+D;MAAA,6B;MAAA,uC;QAUoB,Q;
QAFhB,YAAY,gB;QACZ,aAAa,gB;QACb,wBAAGB,SAAhB,gB;UAGB,cAAA,SAAhB,M;UACI,IAAI,UAAU,O
AAV,CAAJ,C;YACI,KAAM,WAAI,OAAJ,C;;YAEN,MAAO,WAAI,OAAJ,C;;;QAGf,OAAO,cAAK,KAAL,EA
AY,MAAZ,C;O;KAjBX,C;0FAoBA,yB;MAAA,+D;MAAA,6B;MAAA,uC;QAUoB,Q;QAFhB,YAAY,gB;QACZ,
aAAa,gB;QACb,wBAAGB,SAAhB,gB;UAGB,cAAA,SAAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,KA
AM,WAAI,OAAJ,C;;YAEN,MAAO,WAAI,OAAJ,C;;;QAGf,OAAO,cAAK,KAAL,EAAY,MAAZ,C;O;KAjBX,C;
0FAoBA,yB;MAAA,+D;MAAA,6B;MAAA,uC;QAUoB,Q;QAFhB,YAAY,gB;QACZ,aAAa,gB;QACb,wBAAGB,
SAAhB,gB;UAGB,cAAA,SAAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,KAAM,WAAI,OAAJ,C;;YAEN,
MAAO,WAAI,OAAJ,C;;;QAGf,OAAO,cAAK,KAAL,EAAY,MAAZ,C;O;KAjBX,C;0FAoBA,yB;MAAA,+D;MA
AA,6B;MAAA,uC;QAUoB,Q;QAFhB,YAAY,gB;QACZ,aAAa,gB;QACb,wBAAGB,SAAhB,gB;UAGB,cAAA,S
AAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,KAAM,WAAI,OAAJ,C;;YAEN,MAAO,WAAI,OAAJ,C;;;Q
AGf,OAAO,cAAK,KAAL,EAAY,MAAZ,C;O;KAjBX,C;0FAoBA,yB;MAAA,+D;MAAA,6B;MAAA,uC;QAUoB
,Q;QAFhB,YAAY,gB;QACZ,aAAa,gB;QACb,wBAAGB,SAAhB,gB;UAGB,cAAA,SAAhB,M;UACI,IAAI,UAA
U,OAAV,CAAJ,C;YACI,KAAM,WAAI,OAAJ,C;;YAEN,MAAO,WAAI,OAAJ,C;;;QAGf,OAAO,cAAK,KAAL,E
AAY,MAAZ,C;O;KAjBX,C;0FAoBA,yB;MAAA,+D;MAAA,6B;MAAA,uC;QAUoB,Q;QAFhB,YAAY,gB;QAC
Z,aAAa,gB;QACb,wBAAGB,SAAhB,gB;UAGB,cAAA,SAAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,K
AAM,WAAI,OAAJ,C;;YAEN,MAAO,WAAI,OAAJ,C;;;QAGf,OAAO,cAAK,KAAL,EAAY,MAAZ,C;O;KAjBX,
C;0FAoBA,yB;MAAA,+D;MAAA,oC;MAAA,gC;MAAA,6B;MAAA,uC;QAUoB,Q;QAFhB,YAAY,gB;QACZ,a
AAa,gB;QACb,wBAAGB,SAAhB,gB;UAGB,cAAhB,UAGB,SAAhB,O;UACI,IAAI,UAAU,oBAAV,CAAJ,C;Y
ACI,KAAM,WAAI,oBAAJ,C;;YAEN,MAAO,WAAI,oBAAJ,C;;;QAGf,OAAO,cAAK,KAAL,EAAY,MAAZ,C;O;
KAjBX,C;IAoBA,+B;MAkGI,WkB3orBO,MAAO,KlB2orBG,gBkB3orBH,ElBjrbH,KA2FkB,OkB3orBf,C;MIB4
orBd,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WA9FqB,GA8FP,UAAK,C
AAL,CA9FO,EAAnB,KA8FqB,CAAM,CAAN,CA9FF,CA8FrB,C;;MA9FT,OAGGO,I;K;IA7FX,iC;MAwGI,WkB
3prBO,MAAO,KlB2prBG,gBkB3prBH,ElB0jrBH,KAIgkB,OkB3prBf,C;MIB4prBd,WAAW,iBAAa,IAAb,C;MA
CX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WApGqB,GAoGP,UAAK,CAAL,CAPGO,EAAnB,KAoGqB,C
AAM,CAAN,CAPGF,CAoGrB,C;;MAPGT,OAsGO,I;K;IANGX,iC;MA8GI,WkB3qrBO,MAAO,KlB2qrBG,gBkB3
qrBH,ElBokrBH,KAuGkB,OkB3qrBf,C;MIB4qrBd,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAI
B,M;QACI,IAAK,WA1GqB,GA0GP,UAAK,CAAL,CA1GO,EAAnB,KA0GqB,CAAM,CAAN,CA1GF,CA0GrB,
C;;MA1GT,OA4GO,I;K;IAzGX,iC;MAoHI,WkB3rrBO,MAAO,KlB2rrBG,gBkB3rrBH,ElB8krBH,KA6GkB,OkB
3rrBf,C;MIB4rrBd,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WAhHqB,GAG

HP,UAAK,CAAL,CAhHO,EAAAnB,KAgHqB,CAAM,CAAN,CAhHF,CAGhrB,C;;MAhHT,OAKHO,I;K;IA/GX,iC
;MA0HI,WkB3srBO,MAAO,KIB2srBG,gBk3srBH,ElBwlrBH,KAmHkB,OkB3srBf,C;MIB4srBd,WAAW,iBAAa
,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WAtHqB,GAsHP,UAAK,CAAL,CAtHO,EAAAnB,
KAsHqB,CAAM,CAAN,CAtHF,CAsHrB,C;;MatHT,OAWhO,I;K;IARHX,iC;MAGII,WkB3trBO,MAAO,KIB2trB
G,gBk3trBH,ElBkMrBH,KAyHkB,OkB3trBf,C;MIB4trBd,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAk
B,IAAIB,M;QACI,IAAK,WA5HqB,GA4HP,UAAK,CAAL,CA5HO,EAAAnB,KA4HqB,CAAM,CAAN,CA5HF,CA
4HrB,C;;MA5HT,OA8HO,I;K;IA3HX,iC;MA5II,WkB3urBO,MAAO,KIB2urBG,gBk3urBH,ElB4mrBH,KA+Hk
B,OkB3urBf,C;MIB4urBd,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WAlIq
B,GakIP,UAAK,CAAL,CAIIO,EAAAnB,KakIqB,CAAM,CAAN,CAIIF,CakIrB,C;;MAIIT,OAoIO,I;K;IAjIX,iC;M
A4II,WkB3vrBO,MAAO,KIB2vrBG,gBk3vrBH,ElBsnrBH,KaQIkB,OkB3vrBf,C;MIB4vrBd,WAAW,iBAAa,IA
Ab,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WAxIqB,GAwIP,UAAK,CAAL,CaxIO,EAAAnB,KA
wIqB,CAAM,CAAN,CaxIF,CawIrB,C;;MAXIT,OA0IO,I;K;IAvIX,iC;MAkJI,WkB3wrBO,MAAO,KIB2wrBG,gB
k3wrBH,ElBgorBH,KA2IkB,OkB3wrBf,C;MIB4wrBd,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,I
AAIB,M;QACI,IAAK,WA9IqB,GA8IP,sBAAK,CAAL,EA9IO,EAAAnB,KA8IqB,CAAM,CAAN,CA9IF,CA8IrB,C;
;MA9IT,OAGJO,I;K;8EA7IX,yB;MAAA,gE;MkBzorBA,iB;MIByorBA,8C;QAQI,WkB3orBO,MAAO,KIB2orBG,
gBk3orBH,ElB2orBS,KAAM,OkB3orBf,C;QIB4orBd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IA
AIB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,EAAMB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAA
O,I;O;KAbX,C;8EAgBA,yB;MAAA,gE;MkBzprBA,iB;MIByprBA,8C;QAQI,WkB3prBO,MAAO,KIB2prBG,gBk
B3prBH,ElB2prBS,KAAM,OkB3prBf,C;QIB4prBd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,
M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,EAAMB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;
O;KAbX,C;+EAgBA,yB;MAAA,gE;MkBzqrBA,iB;MIByqrBA,8C;QAQI,WkB3qrBO,MAAO,KIB2qrBG,gBk3q
rBH,ElB2qrBS,KAAM,OkB3qrBf,C;QIB4qrBd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;U
ACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,EAAMB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;K
AbX,C;8EAgBA,yB;MAAA,gE;MkBzrrBA,iB;MIByrrBA,8C;QAQI,WkB3rrBO,MAAO,KIB2rrBG,gBk3rrBH,E
lB2rrBS,KAAM,OkB3rrBf,C;QIB4rrBd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IA
AK,WAAI,UAAU,UAAK,CAAL,CAAV,EAAMB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;
+EAgBA,yB;MAAA,gE;MkBzsrBA,iB;MIBysrBA,8C;QAQI,WkB3srBO,MAAO,KIB2srBG,gBk3srBH,ElB2srB
S,KAAM,OkB3srBf,C;QIB4srBd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,W
AAI,UAAU,UAAK,CAAL,CAAV,EAAMB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgB
A,yB;MAAA,gE;MkBztrBA,iB;MIBytrBA,8C;QAQI,WkB3trBO,MAAO,KIB2trBG,gBk3trBH,ElB2trBS,KAAM
,OkB3trBf,C;QIB4trBd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAA
U,UAAK,CAAL,CAAV,EAAMB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MA
AA,gE;MkBzurBA,iB;MIByurBA,8C;QAQI,WkB3urBO,MAAO,KIB2urBG,gBk3urBH,ElB2urBS,KAAM,OkB3
urBf,C;QIB4urBd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,U
AAK,CAAL,CAAV,EAAMB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,
gE;MkBzvrBA,iB;MIByvrBA,8C;QAQI,WkB3vrBO,MAAO,KIB2vrBG,gBk3vrBH,ElB2vrBS,KAAM,OkB3vrB
f,C;QIB4vrBd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,UAA
K,CAAL,CAAV,EAAMB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;
MAAA,oC;MkBzwrBA,iB;MIBywrBA,8C;QAQI,WkB3wrBO,MAAO,KIB2wrBG,gBk3wrBH,ElB2wrBS,KAA
M,OkB3wrBf,C;QIB4wrBd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,
UAAU,sBAAK,CAAL,EA AV,EAAMB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;IAGBA,kC;
MAqGoB,gB;MAHhB,gBAAGB,gB;MACHB,WAAW,iBkbt3rBJ,MAAO,KIBs3rBsB,wBA5FzB,KA4FyB,EAawB
,EAaxB,Ckbt3rBtB,ElBs3rBmD,Skbt3rBnD,CIBs3rBH,C;MACX,QAAQ,C;MACQ,OA9FL,KA8FK,W;MAAhB,
OAAgB,cAAhB,C;QAAGB,yB;QACZ,IAAI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK,WahGqB,GAGP,UAAK
,UAAL,EA AK,kBAAL,SahGO,EAGGI,OA hGJ,CAGGrB,C;;MAhGT,OAKGO,I;K;IA/FX,kC;MA6GoB,gB;MAHh
B,gBAAGB,gB;MACHB,WAAW,iBkx4rBJ,MAAO,KIBw4rBsB,wBApGzB,KAoGyB,EAawB,EAaxB,Ckx4rB
tB,ElBw4rBmD,SkBx4rBnD,CIBw4rBH,C;MACX,QAAQ,C;MACQ,OA tGL,KAsGK,W;MAAhB,OAAgB,cAAhB
,C;QAAGB,yB;QACZ,IAAI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK,WAxGqB,GAwGP,UAAK,UAAL,EA AK,

kBAAL,SAxGO,EAwGI,OAxGJ,CAwGrB,C;;MAxGT,OA0GO,I;K;IAvGX,kC;MAqHoB,gB;MAHhB,gBAAgB,gB;MACHB,WAAW,iBkB15rBJ,MAAO,KIB05rBsB,wBA5GzB,KA4GyB,EAawB,EAAXB,CkB15rBtB,ElB05rBmD,SkB15rBnD,CIB05rBH,C;MACX,QAAQ,C;MACQ,OA9GL,KA8GK,W;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK,WAhHqB,GAgHP,UAAK,UAAAL,EAAK,kBAAL,SAhHO,EAghI,OAhhJ,CAgHrB,C;;MAhHT,OAKHO,I;K;IA/GX,kC;MA6HoB,gB;MAHhB,gBAAgB,gB;MACHB,WAAW,iBkB56rBJ,MAAO,KIB46rBsB,wBApHzB,KAoHyB,EAawB,EAAXB,CkB56rBtB,ElB46rBmD,SkB56rBnD,CIB46rBH,C;MACX,QAAQ,C;MACQ,OAthL,KAsHK,W;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK,WAxHqB,GAwHP,UAAK,UAAAL,EAAK,kBAAL,SAxHO,EAwHI,OAxHJ,CAwHrB,C;;MAxHT,OA0HO,I;K;IAvHX,kC;MAqIoB,gB;MAHhB,gBAAgB,gB;MACHB,WAAW,iBkB97rBJ,MAAO,KIB87rBsB,wBA5HzB,KA4HyB,EAawB,EAAXB,CkB97rBtB,ElB87rBmD,SkB97rBnD,CIB87rBH,C;MACX,QAAQ,C;MACQ,OA9HL,KA8HK,W;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK,WAhIqB,GAgIP,UAAK,UAAAL,EAAK,kBAAL,SAhIO,EAghI,OAhhJ,CAgIrB,C;;MAhIT,OAKIO,I;K;IA/HX,kC;MA6IoB,gB;MAHhB,gBAAgB,gB;MACHB,WAAW,iBkBh9rBJ,MAAO,KIBg9rBsB,wBApIzB,KAoIyB,EAawB,EAAXB,CkBh9rBtB,ElBg9rBmD,SkBh9rBnD,CIBg9rBH,C;MACX,QAAQ,C;MACQ,OAiL,KAsIK,W;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK,WAxIqB,GAwIP,UAAK,UAAAL,EAAK,kBAAL,SAxIO,EAwII,OAxIJ,CAwIrB,C;;MAxIT,OA0IO,I;K;IAvIX,kC;MAqJoB,gB;MAHhB,gBAAgB,gB;MACHB,WAAW,iBkBI+rBJ,MAAO,KIBk+rBsB,wBA5IzB,KA4IyB,EAawB,EAAXB,CkBI+rBtB,ElBk+rBmD,SkBI+rBnD,CIBk+rBH,C;MACX,QAAQ,C;MACQ,OA9IL,KA8IK,W;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK,WAhJqB,GAgJP,UAAK,UAAAL,EAAK,kBAAL,SAhJO,EAghI,OAhhJ,CAgIrB,C;;MAhJT,OAKJO,I;K;IA/IX,kC;MA6JoB,gB;MAHhB,gBAAgB,gB;MACHB,WAAW,iBkBp/rBJ,MAAO,KIBo/rBsB,wBApJzB,KAoJyB,EAawB,EAAXB,CkBp/rBtB,ElBo/rBmD,SkBp/rBnD,CIBo/rBH,C;MACX,QAAQ,C;MACQ,OAiJL,KAsJK,W;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK,WAxJqB,GAwJP,UAAK,UAAAL,EAAK,kBAAL,SAxJO,EAwJI,OAxJJ,CAwJrB,C;;MAxJT,OA0JO,I;K;IAvJX,kC;MAqKoB,gB;MAHhB,gBAAgB,gB;MACHB,WAAW,iBkBtgsBJ,MAAO,KIBsgsBsB,wBA5JzB,KA4JyB,EAawB,EAAXB,CkBtgsBtB,ElBsgsBmD,SkBtgsBnD,CIBsgsBH,C;MACX,QAAQ,C;MACQ,OA9JL,KA8JK,W;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK,WAhKqB,GAgKP,sBAAK,UAAAL,EAAK,kBAAL,UAhKO,EAghI,OAhhJ,CAgKrB,C;;MAhKT,OAKKO,I;K;+EA/JX,yB;MAAA,kF;MAAA,gE;Mkbn3rBA,iB;MIBm3rBA,8C;QAWoB,UAEY,M;QAL5B,gBAAgB,gB;QACHB,WAAW,ekBt3rBJ,MAAO,KIBs3rBsB,wBAAN,KAAM,EAawB,EAAXB,CkBt3rBtB,ElBs3rBmD,SkBt3rBnD,CIBs3rBH,C;QACX,QAAQ,C;QACQ,uB;QAaHb,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAAI,KAAK,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,UAAK,UAAAL,EAAK,kBAAL,SAAV,EAAqB,OAARb,CAAJ,C;;QAET,OAAO,I;O;KafX,C;+EakBA,yB;MAAA,kF;MAAA,gE;MkBr4rBA,iB;MIBq4rBA,8C;QAWoB,UAEY,M;QAL5B,gBAAgB,gB;QACHB,WAAW,ekBx4rBJ,MAAO,KIBw4rBsB,wBAAN,KAAM,EAawB,EAAXB,CkBx4rBtB,ElBw4rBmD,SkBx4rBnD,CIBw4rBH,C;QACX,QAAQ,C;QACQ,uB;QAaHb,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAAI,KAAK,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,UAAK,UAAAL,EAAK,kBAAL,SAAV,EAAqB,OAARb,CAAJ,C;;QAET,OAAO,I;O;KafX,C;+EakBA,yB;MAAA,kF;MAAA,gE;MkBv5rBA,iB;MIBu5rBA,8C;QAWoB,UAEY,M;QAL5B,gBAAgB,gB;QACHB,WAAW,ekB15rBJ,MAAO,KIB05rBsB,wBAAN,KAAM,EAawB,EAAXB,CkB15rBtB,ElB05rBmD,SkB15rBnD,CIB05rBH,C;QACX,QAAQ,C;QACQ,uB;QAaHb,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAAI,KAAK,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,UAAK,UAAAL,EAAK,kBAAL,SAAV,EAAqB,OAARb,CAAJ,C;;QAET,OAAO,I;O;KafX,C;+EakBA,yB;MAAA,kF;MAAA,gE;MkBz6rBA,iB;MIBy6rBA,8C;QAWoB,UAEY,M;QAL5B,gBAAgB,gB;QACHB,WAAW,ekB56rBJ,MAAO,KIB46rBsB,wBAAN,KAAM,EAawB,EAAXB,CkB56rBtB,ElB46rBmD,SkB56rBnD,CIB46rBH,C;QACX,QAAQ,C;QACQ,uB;QAaHb,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAAI,KAAK,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,UAAK,UAAAL,EAAK,kBAAL,SAAV,EAAqB,OAARb,CAAJ,C;;QAET,OAAO,I;O;KafX,C;+EakBA,yB;MAAA,kF;MAAA,gE;MkB37rBA,iB;MIB27rBA,8C;QAWoB,UAEY,M;QAL5B,gBAAgB,gB;QACHB,WAAW,ekB97rBJ,MAAO,KIB87rBsB,wBAAN,KAAM,EAawB,EAAXB,CkB97rBtB,ElB87rBmD,SkB97rBnD,CIB87rBH,C;QACX,QAAQ,C;QACQ,uB;QAaHb,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAAI,KAAK,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,UAAK,UAAAL,EAAK,kBAAL,SAAV,EAAqB,OAARb,CAAJ,C;;QAET,

OAAO,I;O;KafX,C;+EakBA,yB;MAAA,kF;MAAA,gE;MkB78rBA,iB;MIB68rBA,8C;QAWoB,UAEY,M;QAL5B,gBAAgB,gB;QACbB,WAAW,ekBh9rBJ,MAAO,KIBg9rBsB,wBAAN,KAAM,EAAwB,EAAxB,CkBh9rBtB,ElBg9rBmD,SkBh9rBnD,CIBg9rBH,C;QACX,QAAQ,C;QACQ,uB;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAAI,KAAK,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,UAAK,UAAAL,EAAK,kBAAL,SAAV,EAAqB,OAArB,CAAJ,C;;QAET,OAAO,I;O;KafX,C;+EakBA,yB;MAAA,kF;MAAA,gE;MkB/9rBA,iB;MIB+9rBA,8C;QAWoB,UAEY,M;QAL5B,gBAAgB,gB;QACbB,WAAW,ekBl+rBJ,MAAO,KIBk+rBsB,wBAAN,KAAM,EAAwB,EAAxB,CkBl+rBtB,ElBk+rBmD,SkBl+rBnD,CIBk+rBH,C;QACX,QAAQ,C;QACQ,uB;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAAI,KAAK,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,UAAK,UAAAL,EAAK,kBAAL,SAAV,EAAqB,OAArB,CAAJ,C;;QAET,OAAO,I;O;KafX,C;+EakBA,yB;MAAA,kF;MAAA,gE;MkBj/rBA,iB;MIBi/rBA,8C;QAWoB,UAEY,M;QAL5B,gBAAgB,gB;QACbB,WAAW,ekBp/rBJ,MAAO,KIBo/rBsB,wBAAN,KAAM,EAAwB,EAAxB,CkBp/rBtB,ElBo/rBmD,SkBp/rBnD,CIBo/rBH,C;QACX,QAAQ,C;QACQ,uB;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAAI,KAAK,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,UAAK,UAAAL,EAAK,kBAAL,SAAV,EAAqB,OAArB,CAAJ,C;;QAET,OAAO,I;O;KafX,C;+EakBA,yB;MAAA,kF;MAAA,gE;MAAA,oC;MkBngsBA,iB;MlBmgsBA,8C;QAWoB,UAEY,M;QAL5B,gBAAgB,gB;QACbB,WAAW,ekBtgsBJ,MAAO,KIBsgsBsB,wBAAN,KAAM,EAAwB,EAAxB,CkBtgsBtB,ElBsgsBmD,SkBtgsBnD,CIBsgsBH,C;QACX,QAAQ,C;QACQ,uB;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAAI,KAAK,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,sBAAK,UAAAL,EAAK,kBAAL,UAAV,EAAqB,OAArB,CAAJ,C;;QAET,OAAO,I;O;KafX,C;IAkB A,kC;MAwFI,WkBvmsBO,MAAO,KIBumsBG,gBkBvmsBH,ElBshsBH,KAiFkB,OkBvmsBf,C;MlBwmsBd,WAAW,iBAaA,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WApFqB,GAoFP,UAAK,CAAL,CAPFO,EAAnB,KAoFqB,CAAM,CAAN,CAPFF,CAoFrB,C;;MApFT,OAsFO,I;K;IANFX,kC;MA8FI,WkBvnsBO,MAAO,KIBunsBG,gBkBvnsBH,ElBgisBH,KAuFkB,OkBvnsBf,C;MlBwnsBd,WAAW,iBAaA,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WA1FqB,GA0FP,UAAK,CAAL,CA1FO,EAAnB,KA0FqB,CAAM,CAAN,CA1FF,CA0FrB,C;;MA1FT,OA4FO,I;K;IAzFX,kC;MAoGI,WkBvosBO,MAAO,KIBuosBG,gBkBvosBH,ElB0isBH,KA6FkB,OkBvosBf,C;MlBwosBd,WAAW,iBAaA,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WAhGqB,GAgGP,UAAK,CAAL,CAhGO,EAAnB,KAgGqB,CAAM,CAAN,CAhGF,CAGrB,C;;MAhGT,OAkGO,I;K;IAFX,kC;MA0GI,WkBvpsBO,MAAO,KIBupsBG,gBkBvpsBH,ElBojsBH,KAmGkB,OkBvpsBf,C;MlBwpsBd,WAAW,iBAaA,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WAtGqB,GAsGP,UAAK,CAAL,CAtGO,EAAnB,KAsGqB,CAAM,CAAN,CAtGF,CAsGrB,C;;MAtGT,OAwGO,I;K;IARGX,kC;MAGHI,WkBvqsBO,MAAO,KIBuqsBG,gBkBvqsBH,ElB8jsBH,KAyGkB,OkBvqsBf,C;MlBwqsBd,WAAW,iBAaA,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WA5GqB,GA4GP,UAAK,CAAL,CA5GO,EAAnB,KA4GqB,CAAM,CAAN,CA5GF,CA4GrB,C;;MA5GT,OA8GO,I;K;IA3GX,kC;MASHI,WkBvrsBO,MAAO,KIBursBG,gBkBvrsBH,ElBwksBH,KA+GkB,OkBvrsBf,C;MlBwrsBd,WAAW,iBAaA,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WAlHqB,GakHP,UAAK,CAAL,CAIHO,EAAnB,KakHqB,CAAM,CAAN,CAIHF,CAkHrB,C;;MAIHT,OAoHO,I;K;IAjHX,kC;MA4HI,WkBvssBO,MAAO,KIBussBG,gBkBvssBH,ElBklsBH,KAqHkB,OkBvssBf,C;MlBwssBd,WAAW,iBAaA,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WAxHqB,GAwHP,UAAK,CAAL,CAXHO,EAAnB,KAwHqB,CAAM,CAAN,CAXHF,CAwHrB,C;;MAXHT,OA0HO,I;K;IAvHX,kC;MAkII,WkBvtsBO,MAAO,KIButsBG,gBkBvtsBH,ElB4lsBH,KA2HkB,OkBvtsBf,C;MlBwtsBd,WAAW,iBAaA,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WA9HqB,GA8HP,sBAAK,CAAL,EA9HO,EA8HE,YA9HrB,KA8HqB,CAAM,CAAN,EA9HF,CA8HrB,C;;MA9HT,OAgIO,I;K;+EA7HX,yB;MAAA,gE;MkBrmsBA,iB;MlBqmsBA,8C;QAQI,WkBvmsBO,MAAO,KIBumsBG,gBkBvmsBH,ElBumsBS,KAAM,OkBvmsBf,C;QlBwmsBd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,EAAMB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MkBrnsBA,iB;MlBqnsBA,8C;QAQI,WkBvnsBO,MAAO,KIBunsBG,gBkBvnsBH,ElBunsBS,KAAM,OkBvnsBf,C;QlBwnsBd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,EAAMB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MkBrmsBA,iB;MlBqmsBA,8C;QAQI,WkBvosBO,MAAO,KIBuosBG,gBkBvosBH,ElBuosBS,KAAM,OkBvosBf,C;QlBwosBd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,EAAMB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MkBrps

BA,iB;MIBqpsBA,8C;QAQI,WkBvpsBO,MAAO,KIBupsBG,gBkBvpsBH,ElBupsBS,KAAM,OkBvpsBf,C;QIBwp
sBd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,C
AAV,EAAMb,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MkBrqsBA
,iB;MlBqqsBA,8C;QAQI,WkBvqsBO,MAAO,KlBuqsBG,gBkBvqsBH,ElBuqsBS,KAAM,OkBvqsBf,C;QlBwqsBd
,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAA
V,EAAMb,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MkBrssBA,iB;
MlBqrsBA,8C;QAQI,WkBvrsBO,MAAO,KlBursBG,gBkBvrsBH,ElBursBS,KAAM,OkBvrsBf,C;QlBwrsBd,WAA
W,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,EA
AmB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MkBrssBA,iB;MlBq
ssBA,8C;QAQI,WkBvssBO,MAAO,KlBussBG,gBkBvssBH,ElBussBS,KAAM,OkBvssBf,C;QlBwssBd,WAAW,e
AAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,EAAMb
,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MAAA,oC;MkBrtsBA,iB;
MlBqtsBA,8C;QAQI,WkBvtsBO,MAAO,KlButsBG,gBkBvtsBH,ElButsBS,KAAM,OkBvtsBf,C;QlBwtsBd,WAA
W,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,sBAAK,CAAL,EA
AmB,kBAAM,CAAN,EAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;IAgBA,4F;MAQ8D,yB;QAAA,YAA0B,I;M
AAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K
;MAAO,yB;QAAA,YAAoC,I;MAGvN,Q;MAFhB,MAAO,gBAAO,MAAP,C;MACP,YAAY,C;MACZ,wBAAgB,
SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,IAAI,iCAAU,CAAd,C;UAAiB,MAAO,gBAAO,SAAP,C;QACxB,IA
AI,QAAQ,CAAR,IAAa,SAAS,KAA1B,C;UACW,gBAAP,MAAO,EAAC,OAAd,EAuB,SAAvB,C;;UACJ,K;;MA
EX,IAAI,SAAS,CAAT,IAAc,QAAQ,KAA1B,C;QAAiC,MAAO,gBAAO,SAAP,C;MACxC,MAAO,gBAAO,OAA
P,C;MACP,OAAO,M;K;IAGX,8F;MAQwD,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,
UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAuC,I;MAGpN,Q;MA
FhB,MAAO,gBAAO,MAAP,C;MACP,YAAY,C;MACZ,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,I
AAI,iCAAU,CAAd,C;UAAiB,MAAO,gBAAO,SAAP,C;QACxB,IAAI,QAAQ,CAAR,IAAa,SAAS,KAA1B,C;UA
CI,IAAI,iBAAJ,C;YACI,MAAO,gBAAO,UAAU,OAAV,CAAP,C;;YAEP,MAAO,gBAAO,OAAQ,WAAf,C;;UAC
R,K;;MAEX,IAAI,SAAS,CAAT,IAAc,QAAQ,KAA1B,C;QAAiC,MAAO,gBAAO,SAAP,C;MACxC,MAAO,gBA
AO,OAAO,M;K;IAGX,8F;MAQyD,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,u
B;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAwC,I;MAG
tN,Q;MAFhB,MAAO,gBAAO,MAAP,C;MACP,YAAY,C;MACZ,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,
M;QACI,IAAI,iCAAU,CAAd,C;UAAiB,MAAO,gBAAO,SAAP,C;QACxB,IAAI,QAAQ,CAAR,IAAa,SAAS,KA
A1B,C;UACI,IAAI,iBAAJ,C;YACI,MAAO,gBAAO,UAAU,OAAV,CAAP,C;;YAEP,MAAO,gBAAO,OAAQ,WA
Af,C;;UACR,K;;MAEX,IAAI,SAAS,CAAT,IAAc,QAAQ,KAA1B,C;QAAiC,MAAO,gBAAO,SAAP,C;MACxC,
MAAO,gBAAO,OAAO,M;K;IAGX,8F;MAQuD,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAu
B,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YA
AsC,I;MAGIN,Q;MAFhB,MAAO,gBAAO,MAAP,C;MACP,YAAY,C;MACZ,wBAAgB,SAAhB,gB;QAAGB,cAA
A,SAAhB,M;QACI,IAAI,iCAAU,CAAd,C;UAAiB,MAAO,gBAAO,SAAP,C;QACxB,IAAI,QAAQ,CAAR,IAAa,
SAAS,KAA1B,C;UACI,IAAI,iBAAJ,C;YACI,MAAO,gBAAO,UAAU,OAAV,CAAP,C;;YAEP,MAAO,gBAAO,
OAAQ,WAAf,C;;UACR,K;;MAEX,IAAI,SAAS,CAAT,IAAc,QAAQ,KAA1B,C;QAAiC,MAAO,gBAAO,SAAP,
C;MACxC,MAAO,gBAAO,OAAO,M;K;IAGX,8F;MAQwD,yB;QAAA,YAA0B,I;MAAM,sB;Q
AAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB
;QAAA,YAAuC,I;MAGpN,Q;MAFhB,MAAO,gBAAO,MAAP,C;MACP,YAAY,C;MACZ,wBAAgB,SAAhB,gB;
QAAGB,cAAA,SAAhB,M;QACI,IAAI,iCAAU,CAAd,C;UAAiB,MAAO,gBAAO,SAAP,C;QACxB,IAAI,QAAQ,
CAAR,IAAa,SAAS,KAA1B,C;UACI,IAAI,iBAAJ,C;YACI,MAAO,gBAAO,UAAU,OAAV,CAAP,C;;YAEP,MA
AO,gBAAO,OAAQ,WAAf,C;;UACR,K;;MAEX,IAAI,SAAS,CAAT,IAAc,QAAQ,KAA1B,C;QAAiC,MAAO,gB
AAO,SAAP,C;MACxC,MAAO,gBAAO,OAAO,M;K;IAGX,8F;MAQyD,yB;QAAA,YAA0B,I;M
AAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K
;MAAO,yB;QAAA,YAAwC,I;MAGtN,Q;MAFhB,MAAO,gBAAO,MAAP,C;MACP,YAAY,C;MACZ,wBAAgB,
SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,IAAI,iCAAU,CAAd,C;UAAiB,MAAO,gBAAO,SAAP,C;QACxB,IA

AI,QAAQ,CAAR,IAAa,SAAS,KAA1B,C;UACI,IAAI,iBAAJ,C;YACI,MAAO,gBAAO,UAAU,OAAV,CAAP,C;;
YAEP,MAAO,gBAAO,OAAQ,WAAf,C;;UACR,K;;MAEX,IAAI,SAAS,CAAT,IAAc,QAAQ,KAA1B,C;QAAiC,
MAAO,gBAAO,SAAP,C;MACxC,MAAO,gBAAO,OAAP,C;MACP,OAAO,M;K;IAGX,8F;MAQ0D,yB;QAAA,Y
AA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,
YAA0B,K;MAAO,yB;QAAA,YAAyC,I;MAGxN,Q;MAFhB,MAAO,gBAAO,MAAP,C;MACP,YAAy,C;MACZ,
wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,IAAI,iCAAU,CAAd,C;UAAiB,MAAO,gBAAO,SAAP,C;
QACxB,IAAI,QAAQ,CAAR,IAAa,SAAS,KAA1B,C;UACI,IAAI,iBAAJ,C;YACI,MAAO,gBAAO,UAAU,OAAV,
CAAP,C;;YAEP,MAAO,gBAAO,OAAQ,WAAf,C;;UACR,K;;MAEX,IAAI,SAAS,CAAT,IAAc,QAAQ,KAA1B,C
;QAAiC,MAAO,gBAAO,SAAP,C;MACxC,MAAO,gBAAO,OAAP,C;MACP,OAAO,M;K;IAGX,8F;MAQ2D,yB;
QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,y
B;QAAA,YAA0B,K;MAAO,yB;QAAA,YAA0C,I;MAG1N,Q;MAFhB,MAAO,gBAAO,MAAP,C;MACP,YAAy,
C;MACZ,wBAAgB,SAAhB,gB;QAAGB,cAAA,SAAhB,M;QACI,IAAI,iCAAU,CAAd,C;UAAiB,MAAO,gBAAO,
SAAP,C;QACxB,IAAI,QAAQ,CAAR,IAAa,SAAS,KAA1B,C;UACI,IAAI,iBAAJ,C;YACI,MAAO,gBAAO,UAA
U,OAAV,CAAP,C;;YAEP,MAAO,gBAAO,OAAQ,WAAf,C;;UACR,K;;MAEX,IAAI,SAAS,CAAT,IAAc,QAAQ,
KAA1B,C;QAAiC,MAAO,gBAAO,SAAP,C;MACxC,MAAO,gBAAO,OAAP,C;MACP,OAAO,M;K;IAGX,8F;M
AQwD,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,
E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAuC,I;MAGpN,Q;MAFhB,MAAO,gBAAO,MAAP,C;MAC
P,YAAy,C;MACZ,wBAAgB,SAAhB,gB;QAAGB,cAAhB,UAAgB,SAAhB,O;QACI,IAAI,iCAAU,CAAd,C;UAAi
B,MAAO,gBAAO,SAAP,C;QACxB,IAAI,QAAQ,CAAR,IAAa,SAAS,KAA1B,C;UACI,IAAI,iBAAJ,C;YACI,MA
AO,gBAAO,UAAU,oBAAV,CAAP,C;;YAEP,MAAO,gBAAO,OAAP,C;;UACR,K;;MAEX,IAAI,SAAS,CAAT,IA
Ac,QAAQ,KAA1B,C;QAAiC,MAAO,gBAAO,SAAP,C;MACxC,MAAO,gBAAO,OAAP,C;MACP,OAAO,M;K;I
AGX,0F;MAQyC,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QA
AA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAoC,I;MACIN,OAAO,kBAAO,sBAAP,EAAwB
,SAAxB,EAAmC,MAAnC,EAA2C,OAA3C,EAAoD,KAApD,EAA2D,SAA3D,EAAeE,SAAtE,CAAI,F,W;K;IAG5
F,4F;MAQkC,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,
QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAuC,I;MAC9M,OAAO,oBAAO,sBAAP,EAAwB,S
AAxB,EAAmC,MAAnC,EAA2C,OAA3C,EAAoD,KAApD,EAA2D,SAA3D,EAAeE,SAAtE,CAAI,F,W;K;IAG5F,
4F;MAQmC,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,
QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAwC,I;MACHN,OAAO,oBAAO,sBAAP,EAAwB,S
AAxB,EAAmC,MAAnC,EAA2C,OAA3C,EAAoD,KAApD,EAA2D,SAA3D,EAAeE,SAAtE,CAAI,F,W;K;IAG5F,
4F;MAQiC,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,Q
AAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAuC,I;MAC5M,OAAO,oBAAO,sBAAP,EAAwB,SAA
xB,EAAmC,MAAnC,EAA2C,OAA3C,EAAoD,KAApD,EAA2D,SAA3D,EAAeE,SAAtE,CAAI,F,W;K;IAG5F,4F;
MAQkC,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAA
a,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAuC,I;MAC9M,OAAO,oBAAO,sBAAP,EAAwB,SAAxB
,EAAmC,MAAnC,EAA2C,OAA3C,EAAoD,KAApD,EAA2D,SAA3D,EAAeE,SAAtE,CAAI,F,W;K;IAG5F,4F;M
AQmC,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,
E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAwC,I;MACHN,OAAO,oBAAO,sBAAP,EAAwB,SAAxB,E
AAmC,MAAnC,EAA2C,OAA3C,EAAoD,KAApD,EAA2D,SAA3D,EAAeE,SAAtE,CAAI,F,W;K;IAG5F,4F;MAQ
oC,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;M
AAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAyC,I;MACIN,OAAO,oBAAO,sBAAP,EAAwB,SAAxB,EAAm
C,MAAnC,EAA2C,OAA3C,EAAoD,KAApD,EAA2D,SAA3D,EAAeE,SAAtE,CAAI,F,W;K;IAG5F,4F;MAQqC,y
B;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI
,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAA0C,I;MACpN,OAAO,oBAAO,sBAAP,EAAwB,SAAxB,EAAmC,
MAAnC,EAA2C,OAA3C,EAAoD,KAApD,EAA2D,SAA3D,EAAeE,SAAtE,CAAI,F,W;K;IAG5F,4F;MAQkC,yB;
QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,y
B;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAuC,I;MAC9M,OAAO,oBAAO,sBAAP,EAAwB,SAAxB,EAAmC,M
AAAnC,EAA2C,OAA3C,EAAoD,KAApD,EAA2D,SAA3D,EAAeE,SAAtE,CAAI,F,W;K;IAQxE,4C;MAAA,mB;Q

vB,GAAGC,MAAM,K;K;IAGjD,wB;MAMoB,Q;MADhB,UAAe,C;MACf,wBAAGB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,YAAO,O;;MAEX,OAAO,G;K;IAGX,0B;MAMoB,Q;MADhB,UAAe,C;MACf,wBAAGB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,YAAO,O;;MAEX,OAAO,G;K;IAGX,0B;MAMoB,Q;MADhB,UAAe,C;MACf,wBAAGB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,YAAO,OAAP,I;;MAEJ,OAAO,G;K;IAGX,0B;MAMoB,Q;MADhB,Y;MACA,wBAAGB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,cAAO,OAAP,C;;MAEJ,OAAO,G;K;IAGX,0B;MAMoB,Q;MADhB,UAAiB,G;MACjB,wBAAGB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OA AO,O;;MAEX,OAAO,G;K;IAGX,0B;MAMoB,Q;MADhB,UAAkB,G;MACiB,wBAAGB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,O;;MAEX,OAAO,G;K;IAGX,0B;MAKoB,Q;MADhB,UAAe,C;MACf,wBAAGB,SA AhB,gB;QAAgB,cAAA,SAAhB,M;QACI,YAAO,O;;MAEX,OAAO,G;K;IAGX,0B;MAKoB,Q;MADhB,UAAe,C; MACf,wBAAGB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,YAAO,O;;MAEX,OAAO,G;K;IAGX,0B;MAKoB, Q;MADhB,UAAe,C;MACf,wBAAGB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,YAAO,OAAP,I;;MAEJ,OAA O,G;K;IAGX,0B;MAKoB,Q;MADhB,Y;MACA,wBAAGB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,cAAO,O AAP,C;;MAEJ,OAAO,G;K;IAGX,0B;MAKoB,Q;MADhB,UAAiB,G;MACjB,wBAAGB,SAAhB,gB;QAAgB,cAA A,SAAhB,M;QACI,OAAO,O;;MAEX,OAAO,G;K;IAGX,2B;MAKoB,Q;MADhB,UAAkB,G;MACiB,wBAAGB,S AAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,O;;MAEX,OAAO,G;K;Ia5uuBX,oD;MAQuF,wC;K;IARvF,8C ASI,Y;MAAuC,8B;K;IAT3C,gF;4FOOA,qB;MAOI,OAAO,sBAAL,CAAJ,C;K;4FAGX,qB;MAOI,OAAO,sBAAL, CAAJ,C;K;4FAGX,qB;MAOI,OAAO,sBAAL,CAAJ,C;K;4FAGX,qB;MAOI,OAAO,sBAAL,CAAJ,C;K;4FAGX,q B;MAOI,OAAO,sBAAL,CAAJ,C;K;IAGX,wC;MAII,IAAI,oCAAJ,C;QACI,OAAO,yBAAS,OAAT,C;MACX,OA AO,qBAAQ,OAAR,KAAoB,C;K;IAWG,yC;MAAA,qB;QAAE,MAAM,8BAA0B,iDAA8C,aAA9C,MAA1B,C;O; K;IAR1C,qC;MAMI,IAAI,8BAAJ,C;QACI,OAAO,sBAAL,KAAJ,C;MACX,OAAO,6BAAgB,KAAhB,EAAuB,uB AA vB,C;K;0FAGX,4B;MAOI,OAAO,sBAAL,KAAJ,C;K;IAGX,2D;MACqB,Q;MARjB,IAAI,8BAAJ,C;QACI,OA AsB,KA4Lf,IAAS,CAAT,IA5Le,KA4LD,IAAS,iBA5LvB,SA4LuB,CAA3B,GA5LI,SA4LkC,aA5LnB,KA4LmB,C AAtC,GA5L0B,YA4L4B,CA5LnC,KA4LmC,C;;MA3L7D,IAAI,QAAQ,CAAZ,C;QACI,OAAO,aAAa,KAAb,C;M ACX,eAAe,oB;MACf,YAAAY,C;MACZ,OAAO,QAAS,UAAhB,C;QACI,cAAc,QAAS,O;QACvB,IAAI,WAAS,Y AAT,EAAS,oBAAT,OA AJ,C;UACI,OAAO,O;;MAEf,OAAO,aAAa,KAAb,C;K;sGAGX,yB;MAAA,8D;MAAA,i D;QAOI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAsC,sBAAL,KAAJ,CAATC,GAAsD,aAAa,KAAb,C;O;K APjE,C;IAUA,6C;MACqB,Q;MARjB,IAAI,8BAAJ,C;QACI,OAAY,YAAL,SAAK,EAAU,KAAV,C;MACHB,IAA I,QAAQ,CAAZ,C;QACI,OAAO,I;MACX,eAAe,oB;MACf,YAAAY,C;MACZ,OAAO,QAAS,UAAhB,C;QACI,cAA c,QAAS,O;QACvB,IAAI,WAAS,YAAT,EAAS,oBAAT,OA AJ,C;UACI,OAAO,O;;MAEf,OAAO,I;K;sGAGX,yB; MAAA,sD;MAAA,mC;QAOI,OAAY,UAAAL,SAAK,EAAU,KAAV,C;O;KAPhB,C;gFAUA,gC;MAOW,sB;;QAU HS,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAGB,yB;UAAM,IAvHH,SAuHO,CAAU,OA AV,CAAJ,C;YAAw B,qBAAO,O;YAAP,uB;;QAC9C,qBAAO,I;;MAxHP,yB;K;wFAGJ,gC;MA2VoB,Q;MADhB,WAAe,I;MACC,2B ;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,IARVc,SAqVV,CAAU,OA AV,CAAJ,C;UACI,OAAO,O;;MATVf, OAYVO,I;K;wFATVX,gC;MAOW,qB;;QAwVP,eAAoB,+BAAa,cAAb,C;QACpB,OAAO,QAAS,cAAhB,C;UACI, cAAc,QAAS,W;UACvB,IA3Vc,SA2VV,CAAU,OA AV,CAAJ,C;YAAwB,oBAAO,O;YAAP,sB;;QAE5B,oBAAO ,I;;MA7VP,wB;K;IAGJ,6B;MAMQ,kBADE,SACF,Q;QAAW,OAAY,SAAL,SAAK,C;;QAE nB,eAAe,oB;QACf,IA AI,CAAC,QAAS,UAA d,C;UACI,MAAM,2BAAuB,sBAAvB,C;QACV,OAAO,QAAS,O;;K;IAK5B,6B;MAKI,IA AI,mBAAJ,C;QACI,MAAM,2BAAuB,gBAAvB,C;MACV,OAAO,sBAAK,CAAL,C;K;mFAGX,yB;MAAA,iE; MAAA,uC;QAKoB,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAGB,yB;UAAM,IAAI,UAAU,OA AV,CAAJ,C;Y AAwB,OAAO,O;;QACrD,MAAM,gCAAuB,wDAAvB,C;O;KANV,C;oGASA,yB;MAAA,iE;MAAA,uC;QASW, Q;QAAA,+B;;UAYS,U;UAAA,6B;UAAhB,OAAGB,gBAAhB,C;YAAgB,2B;YACZ,aAbwB,SAaX,CAAU,OA AV ,C;YACb,IAAI,cAAJ,C;cACI,8BAAO,M;cAAP,gC;;UAGR,8BAAO,I;;QAlBA,kC;QAAA,iB;UAAmC,MAAM,g CAAuB,mEAAvB,C;;QAAhD,OAAO,I;O;KATX,C;gHAYA,gC;MASoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB, C;QAAGB,yB;QACZ,aAAa,UAAU,OA AV,C;QACb,IAAI,cAAJ,C;UACI,OAAO,M;;MAGf,OAAO,I;K;IAGX,m C;MAKQ,kBADE,SACF,Q;QACI,IAAI,mBAAJ,C;UACI,OAAO,I;UAEP,OAAO,sBAAK,CAAL,C;;QAGX,eAA e,oB;QACf,IAAI,CAAC,QAAS,UAA d,C;UACI,OAAO,I;QACX,OAAO,QAAS,O;;K;IAK5B,mC;MAII,OAAW,m BAAJ,GAAe,IAAf,GAAyB,sBAAK,CAAL,C;K;+FAGpC,gC;MAIoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;Q AAGB,yB;QAAM,IAAI,UAAU,OA AV,CAAJ,C;UAAwB,OAAO,O;;MACrD,OAAO,I;K;0FAGX,yB;MAAA,8D;

MAAA,iD;QAKI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAsC,sBAAl,KAAJ,CAATc,GAAsD,aAAa,KA
Ab,C;O;KALjE,C;IAQA,uC;MAMI,OAAW,SAAS,CAAT,IAAc,SAAS,2BAA3B,GAAsC,sBAAl,KAAJ,CAATc,G
AAsD,I;K;IAGjE,uC;MAMiB,Q;MAFb,IAAI,8BAAJ,C;QAAkB,OAAO,SAAK,eAAQ,OAAAR,C;MAC9B,YAAAY,
C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,mBAAmB,KAAAnB,C;QACA,IAAI,gBAAW,IAAX,CAAJ,
C;UACI,OAAO,K;QACX,qB;;MAEJ,OAAO,E;K;IAGX,uC;MAKI,OAAO,wBAAQ,OAAAR,C;K;gGAGX,yB;MA
AA,wE;MAAA,uC;QAKiB,Q;QADb,YAAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,mBAAmB,KA
AnB,C;UACA,IAAI,UAAU,IAAV,CAAJ,C;YACI,OAAO,K;UACX,qB;;QAEJ,OAAO,E;O;KAXX,C;gGAcA,gC;
MAKiB,Q;MADb,YAAAY,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,IAAI,UAAU,IAAV,CAAJ,C;UA
CI,OAAO,K;QACX,qB;;MAEJ,OAAO,E;K;8FAGX,yB;MAAA,wE;MAAA,uC;QAMiB,Q;QAFb,gBAAgB,E;QA
ChB,YAAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,mBAAmB,KAAAnB,C;UACA,IAAI,UAAU,IAA
V,CAAJ,C;YACI,YAAAY,K;UACHb,qB;;QAEJ,OAAO,S;O;KAZX,C;8FAeA,gC;MAII,eAAe,SAAK,sBAAa,cAAb
,C;MACpB,OAAO,QAAS,cAAhB,C;QACI,IAAI,UAAU,QAAS,WAAAnB,CAAJ,C;UACI,OAAO,QAAS,Y;;MAG
xB,OAAO,E;K;IAGX,4B;MASQ,kBADE,SACF,Q;QAAW,OAAAY,QAAL,SAAK,C;;QAEEnB,eAAe,oB;QACf,IAA
I,CAAC,QAAS,UAAAd,C;UACI,MAAM,2BAAuB,sBAAvB,C;QACV,WAAW,QAAS,O;QACpB,OAAO,QAAS,U
AAhB,C;UACI,OAAO,QAAS,O;QACpB,OAAO,I;;K;IAKnB,4B;MAQI,IAAI,mBAAJ,C;QACI,MAAM,2BAAuB,
gBAAvB,C;MACV,OAAO,sBAAK,2BAAL,C;K;iFAGX,yB;MAAA,iE;MAAA,gB;MAAA,8B;MAAA,uC;QUO
B,UAQT,M;QAVP,WAAe,I;QACf,YAAAY,K;QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAAI,UAA
U,OAAV,CAAJ,C;YACI,OAAO,O;YACP,QAAQ,I;;QAGhB,IAAI,CAAC,KAAL,C;UAAAY,MAAM,gCAAuB,w
DAAvB,C;QAEIB,OAAO,2E;O;KAIBX,C;iFAqBA,yB;MAAA,iE;MAAA,uC;QAQI,eAAe,SAAK,sBAAa,cAAb,C
;QACpB,OAAO,QAAS,cAAhB,C;UACI,cAAc,QAAS,W;UACvB,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,
O;;QAEEnC,MAAM,gCAAuB,kDAAvB,C;O;KAbV,C;IAGBA,2C;MAOiB,Q;MAHb,IAAI,8BAAJ,C;QAAkB,OAA
O,SAAK,mBAAAY,OAAZ,C;MAC9B,gBAAgB,E;MACHb,YAAAY,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;
QACT,mBAAmB,KAAAnB,C;QACA,IAAI,gBAAW,IAAX,CAAJ,C;UACI,YAAAY,K;QACHb,qB;;MAEJ,OAAO,S;
K;IAGX,2C;MAKI,OAAO,4BAAAY,OAAZ,C;K;IAGX,kC;MAOQ,kBADE,SACF,Q;QAAW,OAAW,mBAAJ,GA
Ae,IAAf,GAAyB,sBAAK,iBAAO,CAAP,IAAL,C;;QAEvC,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UACI,
OAAO,I;QACX,WAAW,QAAS,O;QACpB,OAAO,QAAS,UAAhB,C;UACI,OAAO,QAAS,O;QACpB,OAAO,I;;K
;IAKnB,kC;MAMI,OAAW,mBAAJ,GAAe,IAAf,GAAyB,sBAAK,iBAAO,CAAP,IAAL,C;K;6FAGpC,gC;MAOo
B,Q;MADhB,WAAe,I;MACC,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI,UAAU,OAAV,CAAJ,C;U
ACI,OAAO,O;;MAGf,OAAO,I;K;6FAGX,gC;MAMI,eAAe,SAAK,sBAAa,cAAb,C;MACpB,OAAO,QAAS,cAA
hB,C;QACI,cAAc,QAAS,W;QACvB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,O;;MAEnC,OAAO,I;K;qFA
GX,yB;MAAA,mC;MAAA,gD;MAAA,4B;QAQI,OAAO,kBAAO,cAAP,C;O;KARX,C;IAWA,sC;MAOI,IAAI,m
BAAJ,C;QACI,MAAM,2BAAuB,sBAAvB,C;MACV,OAAO,qBAAU,MAAO,iBAAQ,cAAR,CAAjB,C;K;iGAGX
,yB;MAAA,mC;MAAA,4D;MAAA,4B;QAOI,OAAO,wBAAa,cAAb,C;O;KAPX,C;IAUA,4C;MAMI,IAAI,mBAA
J,C;QACI,OAAO,I;MACX,OAAO,qBAAU,MAAO,iBAAQ,cAAR,CAAjB,C;K;IAGX,8B;MAKQ,kBADE,SACF,
Q;QAAW,OAAAY,UAAL,SAAK,C;;QAEEnB,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UACI,MAAM,2BAAu
B,sBAAvB,C;QACV,aAAa,QAAS,O;QACtB,IAAI,QAAS,UAAb,C;UACI,MAAM,gCAAyB,uCAAzB,C;QACV,
OAAO,M;;K;IAKnB,8B;MAIiB,IAAN,I;MAAA,QAAM,cAAN,C;aACH,C;UAAK,MAAM,2BAAuB,gBAAvB,C;
aACX,C;UAAK,6BAAK,CAAL,C;UAAL,K;;UACQ,MAAM,gCAAyB,iCAAzB,C;;MAHIB,W;K;qFAOJ,yB;MA
AA,kF;MAAA,iE;MAAA,gB;MAAA,8B;MAAA,uC;QAMoB,UAST,M;QAXP,aAAiB,I;QACjB,YAAAY,K;QACI,
2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAAI,UAAU,OAAV,CAAJ,C;YACI,IAAI,KAAJ,C;cAAW,MA
AM,8BAAyB,qDAAzB,C;YACjB,SAAS,O;YACT,QAAQ,I;;QAGhB,IAAI,CAAC,KAAL,C;UAAAY,MAAM,gCA
AuB,wDAAvB,C;QAEIB,OAAO,6E;O;KAFX,C;IAkBA,oC;MAKQ,kBADE,SACF,Q;QAAW,OAAW,mBAAQ,C
AAZ,GAAe,sBAAK,CAAL,CAAf,GAA4B,I;;QAEIc,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UACI,OAAO
,I;QACX,aAAa,QAAS,O;QACtB,IAAI,QAAS,UAAb,C;UACI,OAAO,I;QACX,OAAO,M;;K;IAKnB,oC;MAII,OA
AW,mBAAQ,CAAZ,GAAe,sBAAK,CAAL,CAAf,GAA4B,I;K;iGAGvC,gC;MAMoB,Q;MAFhB,aAAiB,I;MACjB
,YAAAY,K;MACI,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI,UAAU,OAAV,CAAJ,C;UACI,IAAI,K
AAJ,C;YAAW,OAAO,I;UACIB,SAAS,O;UACT,QAAQ,I;;MAGhB,IAAI,CAAC,KAAL,C;QAAAY,OAAO,I;MA
CnB,OAAO,M;K;IAGX,8B;MAoBsC,UAGT,MAHS,EAarB,M;MN7pBb,IAAI,EMooBI,KAAK,CNpoBT,CAAJ,C

;QACI,cMmoBc,sD;QNloBd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MMmoBV,IAAI,MAAK,CAAT,C;QAAY,OA
AO,mB;MACnB,Q;MACA,IAAI,oCAAJ,C;QACI,iBAAiB,iBAAO,CAAP,I;QACjB,IAAI,cAAc,CAAIB,C;UACI,
OAAO,W;QACX,IAAI,eAAc,CAAIB,C;UACI,OAAO,OAAO,kBAAP,C;QACX,OAAO,iBAAa,UAAb,C;QACP,I
AAI,8BAAJ,C;UACI,IAAI,sCAAJ,C;YAC0B,qB;YAAtB,iBAAc,CAAd,wB;cACI,IAAK,WAAI,sBAAK,KAAL,C
AAJ,C;;YAEI,wCAAA,CAAb,C;YAAb,OAAa,gBAAb,C;cAAa,wB;cACT,IAAK,WAAI,IAAJ,C;;;UAEb,OAAO,I;;
;QAIX,OAAO,gB;;MAEX,YAAAY,C;MACC,6B;MAAb,OAAa,gBAAb,C;QAAa,0B;QACT,IAAI,SAAS,CAAb,C;
UAAgB,IAAK,WAAI,MAAJ,C;;UAAe,qB;;MAExC,OAAAY,qBAAL,IAAK,C;K;IAGhB,kC;MNnqBI,IAAI,EM2q
BI,KAAK,CN3qBT,CAAJ,C;QACI,cM0qBc,sD;QNzqBd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MM0qBV,OAAO,
kBAAGB,gBAAV,iBAAO,CAAP,IAAU,EAAC,CAAd,CAAhB,C;K;kGAGX,yB;MAAA,4C;MAAA,qD;MAAA,u
C;QAMI,IAAI,CAAC,mBAAL,C;UACI,eAAe,+BAAa,cAAb,C;UACf,OAAO,QAAS,cAAhB,C;YACI,IAAI,CAA
C,UAAU,QAAS,WAAAnB,CAAL,C;cACI,OAAO,gBAAK,QAAS,YAAT,GAAuB,CAAvB,IAAL,C;;;QAIInB,OAA
O,W;O;KAdX,C;0FAiBA,yB;MAAA,+D;MAAA,uC;QAQiB,Q;QAFb,eAAe,K;QACf,WAAW,gB;QACE,2B;QA
Ab,OAAa,cAAb,C;UAAa,sB;UACT,IAAI,QA AJ,C;YACI,IAAK,WAAI,IAAJ,C;eACJ,IAAI,CAAC,UAAU,IAAV,
CAAL,C;YACD,IAAK,WAAI,IAAJ,C;YACL,WAAW,I;;QAE nB,OAAO,I;O;KafX,C;0FAkBA,yB;MAAA,+D;
MAAA,uC;QAMW,kBAAS,gB;QA2FA,Q;QAAA,2B;QA AhB,OAAGB,cAAhB,C;UAAgB,yB;UAAM,IA3FU,SA
2FN,CAAU,OAAV,CAAJ,C;YAAwB,WAA Y,WAAI,OAAJ,C;;QA3F1D,OA4FO,W;O;KAIGX,C;kGASA,yB;MA
AA,+D;MA6jCA,wE;MA7jCA,uC;QAQW,kBAAGB,gB;QA4jCV,gB;QADb,YAAAY,C;QACC,2B;QAAb,OAAa,c
AAb,C;UAAa,sB;UAhjCT,IAZmC,SAY/B,CAgjCkB,oBAAmB,cAAnB,EAAMb,sBAAnB,UAhjCIB,EAjC+C,IA
hjC/C,CAAJ,C;YAA2C,sBAgjCQ,IAhjCR,C;;QAZ/C,OAcO,W;O;KATBX,C;sGAWA,yB;MAkjCA,wE;MAIjCA,o
D;QAyjCiB,gB;QADb,YAAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UAhjCT,IAAI,UAgjCkB,oBAAmB,c
AAnB,EAAMb,sBAAnB,UAhjCIB,EAjC+C,IAhjC/C,CAAJ,C;YAA2C,sBAgjCQ,IAhjCR,C;;QAE/C,OAAO,W;
O;KAXX,C;wGAcA,yB;MAAA,+D;MAAA,sC;QAMW,kBAAmB,gB;QASV,Q;QAAA,2B;QA AhB,OAAGB,cAA
hB,C;UAAgB,yB;UAAM,IAAI,YAAJ,C;YAAkB,WAA Y,WAAI,OAAJ,C;;QATpD,OAuO,W;O;KAhBX,C;4GAS
A,4C;MAMoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAM,IAAI,YAAJ,C;UAAkB,WAA Y,WA
AI,OAAJ,C;;MACpD,OAAO,W;K;0FAGX,yB;MAAA,+D;MAAA,uC;QAMW,kBAAY,gB;QA4BH,Q;QAAA,2B;
QA AhB,OAAGB,cAAhB,C;UAAgB,yB;UAAM,IAAI,CA5BS,SA4BR,CAAU,OAAV,CAAL,C;YAAyB,WAA Y,
WAAI,OAAJ,C;;QA5B3D,OA6BO,W;O;KANCX,C;IASA,oC;MAMI,OAAO,6BAAGB,gBAAhB,C;K;IAGX,mD;
MAMoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAM,IAAI,eAAJ,C;UAAqB,WAA Y,WAAI,OA
AJ,C;;MACvD,OAAO,W;K;8FAGX,6C;MAMoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAM,I
AAI,CAAC,UAAU,OAAV,CAAL,C;UAAyB,WAA Y,WAAI,OAAJ,C;;MAC3D,OAAO,W;K;wFAGX,6C;MAMo
B,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,WAA Y,
WAAI,OAAJ,C;;MAC1D,OAAO,W;K;IAGX,sC;MAIL,IAAI,OAAQ,UAAZ,C;QAAuB,Od3wBe,W;;Mc4wBtC,O
AA6D,SAAtD,SAAK,iBAAQ,OAAQ,MAAhB,EAAuB,OAAQ,aAAR,GAAuB,CAAvB,IAAvB,CAAiD,C;K;IAGj
E,sC;MAOkB,Q;MAHd,WAAmB,wBAAR,OAAQ,EA AwB,EAAXB,C;MACnB,IAAI,SAAQ,CAAZ,C;QAAe,OA
AO,W;MACTB,WAAW,iBAAa,IAAb,C;MACG,yB;MAAd,OAAC,cAAd,C;QAAC,uB;QACV,IAAK,WAAI,sBAAI
,KAAJ,CAAJ,C;;MAET,OAAO,I;K;IAGX,8B;MAGBiB,Q;MN51Bb,IAAI,EMo1BI,KAAK,CNp1BT,CAAJ,C;QA
CI,cMm1Bc,sD;QN11Bd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MMm1BV,IAAI,MAAK,CAAT,C;QAAY,OAAO,
W;MACnB,IAAI,oCAAJ,C;QACI,IAAI,KAAK,cAAT,C;UAAe,OAAO,mB;QACTB,IAAI,MAAK,CAAT,C;UAA
Y,OAAO,OAAO,mBAAP,C;;MAEvB,YAAAY,C;MACZ,WAAW,iBAAa,CAAb,C;MACE,2B;MAAb,OAAa,cAAb,
C;QAAa,sB;QACT,IAAK,WAAI,IAAJ,C;QACL,IAAI,mCAAW,CAAf,C;UACI,K;;MAER,OAAAY,qBAAL,IAAK,
C;K;IAGhB,kC;MAeqC,IAGhB,I;MNt3BjB,IAAI,EM42BI,KAAK,CN52BT,CAAJ,C;QACI,cM22Bc,sD;QN12Bd,
MAAM,gCAAYB,OAAQ,WAAjC,C;;MM22BV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,c;MA
CX,IAAI,KAAK,IAAT,C;QAAC,OAAO,mB;MACTB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,kBAAP,C;M
ACnB,WAAW,iBAAa,CAAb,C;MACX,IAAI,sCAAJ,C;QACI,iBAAc,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;UA
CI,IAAK,WAAI,sBAAK,KAAL,CAAJ,C;;QAEI,sCAAA,OAAO,CAAP,IAAb,C;QAAb,OAAa,cAAb,C;UAAa,sB;
UACT,IAAK,WAAI,IAAJ,C;;MAEb,OAAO,I;K;kGAGX,yB;MAAA,qD;MAAA,gE;MAAA,gD;MAAA,uC;QA
MI,IAAI,mBAAJ,C;UACI,OAAO,W;QACX,eAAe,+BAAa,cAAb,C;QACf,OAAO,QAAS,cAAhB,C;UACI,IAAI,C
AAC,UAAU,QAAS,WAAAnB,CAAL,C;YACI,QAAS,O;YACT,mBAAmB,iBAAO,QAAS,YAAhB,I;YACnB,IAAI

,iBAAGB,CAAPB,C;CAAU,OOAO,W;YACI,kBAA3B,eAAa,YAAb,C;YACH,OOAGB,kBAAhB,C;cACI,sBAAa,eAAb,C;YAFR,OH1IBD,W;;;QGg2BP,OOAO,iB;O;KAPBX,C;0FAuBA,yB;MAAA,+D;MAAA,uC;QAOiB,Q;QADb,WAAW,gB;QACE,2B;QAAb,OOAa,cAAb,C;UAAa,sB;UACT,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACI,K;UACJ,IAAK,WAAI,IAAJ,C;;QAET,OOAO,I;O;KAZX,C;IAoBA,+B;MAII,IAAI,wCAAsB,kBAAQ,CAAIC,C;QAAqC,OOAO,mB;MAC5C,WAAW,0B;MACN,WAAL,IAAK,C;MACL,OOAO,I;K;IAGX,uC;MAOI,aAAU,2BAAV,OOA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAAR,C;QACf,sBAAK,CAAL,EAAU,SAAK,aAAI,CAAJ,EAAO,sBAAK,CAAL,CAAP,CAAF,C;;K;oFAIR,yB;MAAA,oD;MJn4BA,sC;MAAA,oC;MAAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MI43Bf,sC;QAMI,IAAI,iBAAO,CAAX,C;UAAc,oBJl4Bd,eAAW,iBIk4BsB,QJl4BtB,CAAX,CIk4Bc,C;;O;KANIB,C;wGASA,yB;MAAA,oD;MJz3BA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN,CAA3B,C;W;S;OA+EI,C;MIk3Bf,sC;QAMI,IAAI,iBAAO,CAAX,C;UAAc,oBJx3Bd,eAAW,2BIw3BgC,QJx3BhC,CAAX,CIw3Bc,C;;O;KANIB,C;IASA,sC;MAMI,sBAAS,cAAT,C;K;IAGJ,6B;MASgB,Q;MAHZ,IAAI,oCAAJ,C;QACI,IAAI,kBAAQ,CAAZ,C;UAAe,OAAY,SAAL,SAAK,C;QAEwB,kBAA3C,sBC5+Bsd,sBD4+BtD,uB;QAAmD,mB;QAA3D,OOAOe,OHl7BjE,WGk7BiE,C;;MAEjD,kBAAhB,0B;MAAwB,oB;MAA/B,OHp7BO,W;K;wFGu7BX,yB;MAAA,wD;MJ56BA,sC;MAAA,oC;MAAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MIq6Bf,sC;QAQI,OOAO,sBJ76BP,eAAW,iBI66BiB,QJ76BjB,CAAX,CI66BO,C;O;KARX,C;4GAWA,yB;MAAA,wD;MJp6BA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN,CAA3B,C;W;S;OA+EI,C;MI65Bf,sC;QAMI,OOAO,sBJn6BP,eAAW,2BIIm6B2B,QJn6B3B,CAAX,CIIm6BO,C;O;KANX,C;IASA,uC;MAMI,OOAO,wBAAW,cAAX,C;K;IAGX,6C;MASe,Q;MAHX,IAAI,oCAAJ,C;QACG,IAAI,kBAAQ,CAAZ,C;UAAe,OAAY,SAAL,SAAK,C;QAEe,kBAAIC,sBCvhCuD,sBDuhCvD,uB;QAA0C,iC;QAAID,OOAyE,OH79BrE,WG69BqE,C;;MAErD,kBAAhB,0B;MAAwB,mC;MAA/B,OH/9BO,W;K;IGk+BX,qC;MAMoB,UACL,M;MAHX,aAAa,oBAAa,cAAb,C;MACb,YAAY,C;MACI,2B;MAAhB,OOAGB,cAAhB,C;QAAGB,yB;QACZ,OOAO,cAAP,EAAO,sBAAP,YAAkB,O;;MACtB,OOAO,M;K;IAGX,kC;MAMoB,UACL,M;MAHX,aAAa,cAAU,cAAV,C;MACb,YAAY,C;MACI,2B;MAAhB,OOAGB,cAAhB,C;QAAGB,yB;QACZ,OOAO,cAAP,EAAO,sBAAP,YAAkB,O;;MACtB,OOAO,M;K;IAGX,kC;MAMoB,UACL,M;MAHX,aAAa,iBAAU,cAAV,C;MACb,YAAY,C;MACI,2B;MAAhB,OOAGB,cAAhB,C;QAAGB,oC;QACZ,OOAO,cAAP,EAAO,sBAAP,YAAkB,O;;MACtB,OOAO,M;K;IAGX,oC;MAMoB,UACL,M;MAHX,aAAa,iBAAU,cAAZ,C;MACb,YAAY,C;MACI,2B;MAAhB,OOAGB,cAAhB,C;QAAGB,yB;QACZ,OOAO,cAAP,EAAO,sBAAP,YAAkB,O;;MACtB,OOAO,M;K;IAGX,mC;MAMoB,UACL,M;MAHX,aAAa,iBAAW,cAAX,C;MACb,YAAY,C;MACI,2B;MAAhB,OOAGB,cAAhB,C;QAAGB,yB;QACZ,OOAO,cAAP,EAAO,sBAAP,YAAkB,O;;MACtB,OOAO,M;K;IAGX,iC;MAMoB,UACL,M;MAHX,aAAa,eAAS,cAAT,C;MACb,YAAY,C;MACI,2B;MAAhB,OOAGB,cAAhB,C;QAAGB,yB;QACZ,OOAO,cAAP,EAAO,sBAAP,YAAkB,O;;MACtB,OOAO,M;K;IAGX,kC;MAMoB,UACL,M;MAHX,aAAa,iBAAU,cAAV,C;MACb,YAAY,C;MACI,2B;MAAhB,OOAGB,cAAhB,C;QAAGB,yB;QACZ,OOAO,cAAP,EAAO,sBAAP,YAAkB,O;;MACtB,OOAO,M;K;IAGX,mC;MAMoB,UACL,M;MAHX,aAAa,eAAW,cAAX,C;MACb,YAAY,C;MACI,2B;MAAhB,OOAGB,cAAhB,C;QAAGB,yB;QACZ,OOAO,cAAP,EAAO,sBAAP,YAAkB,O;;MACtB,OOAO,M;K;0FAGX,yB;MAAA,kF;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,uC;QAWI,eAAwD,cAAzC,YAAY,mCAAwB,EAAXB,CAAZ,CAAYC,EAAC,EAAd,C;QACjD,kBAAY,mBAAoB,QAAPB,C;QAYEH,Q;QAAA,2B;QAAhB,OOAGB,cAAhB,C;UAAgB,yB;UACZ,WA1E8C,SA0E/B,CAAU,OOAV,C;UbpkBnB,wBAALI,IAAK,MAAT,EAAGB,IAAK,OOArB,C;;QaofA,OA4EO,W;O;KAxFX,C;+FAeA,yB;MAAA,kF;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yC;QAWI,eAAwD,cAAzC,YAAY,mCAAwB,EAAXB,CAAZ,CAAYC,EAAC,EAAd,C;QACjD,kBAAc,mBAAoB,QAAPB,C;QA2BL,Q;QAAA,2B;QAAhB,OOAGB,cAAhB,C;UAAgB,yB;UACZ,WAAy,aA5BoC,WA4BhC,CAAY,OOAZ,CAAJ,EAA0B,OOA1B,C;;QA5BhB,OA8BO,W;O;KA1CX,C;+FAeA,yB;MAAA,kF;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yD;QAUl,eAAwD,cAAzC,YAAY,mCAAwB,EAAXB,CAAZ,CAAYC,EAAC,EAAd,C;QACjD,kBAAc,mBAAoB,QAAPB,C;QA6BL,Q;QAAA,2B;QAAhB,OOAGB,cAAhB,C;UAAgB,yB;UACZ,WAAy,aA9BoC,WA8BhC,CAAY,OOAZ,CAAJ,EA9BiD,cA8BvB,CAAe,OOAf,CAA1B,C;;QA9BhB,OA9CO,W;O;KA3CX,C;mGAcA,+C;MAUoB,Q;MAAA,2B;MAAhB

,OAAgB,cAAhB,C;QAAgB,yB;QACZ,WAAY,aAAI,YAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;MAEhB,OAAO,W;K;mGAGX,+D;MAUoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAgB,yB;QACZ,WAAY,aAAI,YAAY,OAAZ,CAAJ,EAA0B,eAAe,OAAf,CAA1B,C;;MAEhB,OAAO,W;K;8FAGX,6C;MASoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAgB,yB;QACZ,WAAY,UAAU,OAAV,C;QbpbBnB,wBAAI,IAAK,MAAT,EAAGB,IAAK,OAARb,C;;MaskBA,OAAO,W;K;kGAGX,yB;MAAA,kF;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,2C;QAYI,aAAa,mBAA6D,cAAzC,YAAY,mCAAwB,EAAXB,CAAZ,CAAyC,EAAC,EAAd,CAA7D,C;QAcG,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UAbO,MAcP,aAAI,OAAJ,EAd,e,aAcF,CAAc,OAAAd,CAAb,C;;QAdhB,OAAuB,M;O;KAb3B,C;sGAgBA,iD;MAUoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAgB,yB;QACZ,WAAY,aAAI,OAAJ,EAAa,cAAc,OAAAd,CAAb,C;;MAEhB,OAAO,W;K;IAGX,gD;MAIiB,Q;MAAA,2B;MAAb,OAAa,cAAb,C;QAAA,sB;QACT,WAAY,WAAI,IAAJ,C;;MAEhB,OAAO,W;K;IAGX,gC;MAII,OAAO,0BAAa,eAAW,YAAY,mCAAwB,EAAXB,CAAZ,CAAX,CAAb,C;K;IAGX,6B;MAKqB,IAAN,I;MADX,IAAI,oCAAJ,C;QACW,QAAM,cAAN,C;eACH,C;YAAK,kB;YAAL,K;eACA,C;YAAK,cAAW,8BAAJ,GAaKB,sBAAI,CAAJ,CAAIb,GAA8B,oBAAW,OAAhD,C;YAAL,K;;YACa,uBAAL,SAAK,C;YAHV,K;;QAAP,W;;MAMJ,OAA4B,qBAAhB,gBAAL,SAAK,CAAGB,C;K;IAGhC,oC;MAII,IAAI,oCAAJ,C;QACI,OAAy,gBAAL,SAAK,C;MACHb,OAAO,0BAAa,gBAAb,C;K;IAGX,oC;MAII,OAAO,iBAAU,SAAV,C;K;IAGX,4B;MAOqB,IAAN,I;MADX,IAAI,oCAAJ,C;QACW,QAAM,cAAN,C;eACH,C;YAAK,iB;YAAL,K;eACA,C;YAAK,aAAU,8BAAJ,GAaKB,sBAAK,CAAL,CAAIb,GA A+B,oBAAW,OAAhD,C;YAAL,K;;YACQ,iCAAa,qBAAiB,YAAY,cAAZ,CAAjB,CAAb,C;YAHL,K;;QAAP,W;;MAMJ,OAAwC,oBAAjC,0BAAa,sBAAb,CAAIc,C;K;sFAG5C,yB;MAAA,+D;MAwFA,gD;MAxFA,uC;QAMW,kBAAU,gB;QAsFD,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,WAvF6B,SAuFIB,CAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAxFhB,OA0FO,W;O;KAhGX,C;uFASA,yB;MAAA,+D;MA0FA,gD;MA1FA,uC;QAUW,kBAAU,gB;QAwFD,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,WAzF6B,SAyFIB,CAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QA1FhB,OA4FO,W;O;KA1GX,C;oGAaA,yB;MAAA,+D;MA8BA,wE;MAAA,gD;MA9BA,uC;QAYW,kBAAiB,gB;QA6BR,gB;QADhB,YAAY,C;QACI,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,WA9BoC,SA8BzB,CAAU,oBAAmB,cAAnB,EAAMb,sBAAnB,UAAV,EAAuC,OAAvC,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QA/BhB,OAIcO,W;O;KA7CX,C;oGAeA,yB;MAAA,+D;MAiCA,wE;MAAA,gD;MAjCA,uC;QAYW,kBAAiB,gB;QAgCR,gB;QADhB,YAAY,C;QACI,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,WajCoC,SAiCzB,CAAU,oBAAmB,cAAnB,EAAMb,sBAAnB,UAAV,EAAuC,OAAvC,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QA1ChB,OAoCO,W;O;KAhDX,C;wGAeA,yB;MAAA,wE;MAAA,gD;MAAA,oD;QAWoB,UAC4B,M;QAF5C,YAAY,C;QACI,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,WAAW,UAAU,oBAAmB,cAAnB,EAAMb,sBAAnB,UAAV,EAAuC,OAAvC,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KafX,C;yGakBA,yB;MAAA,wE;MAAA,gD;MAAA,oD;QAWoB,UAC4B,M;QAF5C,YAAY,C;QACI,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,WAAW,UAAU,oBAAmB,cAAnB,EAAMb,sBAAnB,UAAV,EAAuC,OAAvC,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KafX,C;0FAkBA,yB;MAAA,gD;MAAA,oD;QAIoB,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KARX,C;2FAWA,yB;MAAA,gD;MAAA,oD;QAQoB,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KAZX,C;uFAeA,yB;MAAA,wE;MAyBA,+D;MAzBA,yC;QASW,kBAAU,oB;QAYBD,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,UA1BiD,WA0BvC,C;AAY,OAAZ,C;UbnCP,U;UADP,YaynCe,WbznCH,WaynCwB,GbznCxB,C;UACL,IAAI,aAAJ,C;YACH,aaunCuC,gB;YAA5B,WbznCX,aasnCgC,GbznChC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UamnCA,iB;UACA,IAAK,WAAI,OAAJ,C;;QA5BT,OA8BO,W;O;KAvCX,C;uFAYa,yB;MAAA,wE;MA8BA,+D;MA9BA,yD;QAUW,kBAAU,oB;QA8BD,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,UA/BiD,WA+BvC,CAAY,OAAZ,C;UbnCP,U;UADP,Ya2oCe,Wb3oCH,WA2oCwB,Gb3oCxB,C;UACL,IAAI,aAAJ,C;YACH,aaayCuC,gB;YAA5B,WbxoCX,aaawoCgC,GbxoChC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UaqoCA,iB;UACA,IAAK,WajCyD,cAiCrD,CAAE,OAAf,CAAJ,C;;QajCT,OAmCO,W;O;KA7CX,C;0FAaA,yB;MAAA,+D;MAAA,sD;QASoB,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,UAAU,YAAY,OAAZ,C;UbnCP,U;UADP,YaynCe,WbznCH,WaynCwB,GbznCxB,C;UACL,IAAI,aAAJ,C;YACH,aaunCuC,gB;YAA5B,WbznCX,aasnCgC,GbznChC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UamnCA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KADx,C;2FAiBA,yB;MAAA,+D;

MAAA,sE;QAUoB,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,UAAU,YAAY,OAAZ,C;UbzoC
P,U;UADP,Ya2oCe,Wb3oCH,Wa2oCwB,Gb3oCxB,C;UACL,IAAI,aAAJ,C;YACH,aayoCuC,gB;YAA5B,WbxoC
X,aawoCgC,GbxoChC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UaqoCA,iB;UACA,IAAK,WAAI,eAAe,OAAf,CAAJ,
C;;QAET,OAAO,W;O;KafX,C;4FAkBA,yB;MAAA,kC;MAAA,4C;MAAA,wE;QAQW,sC;QAAA,8C;O;MARX,
oDASQ,Y;QAA6C,OAAA,oBAAgB,W;O;MATrE,iDAUQ,mB;QAAoC,gCAAY,OAAZ,C;O;MAV5C,gF;MAAA,
yC;QAQI,2D;O;KARJ,C;8EAca,yB;MAAA,kF;MAAA,gE;MAAA,uC;QAOW,kBAAM,eAAa,mCAAwB,EAAXB
,CAAb,C;QAUeA,Q;QAAA,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,WAAy,WaxEwC,SAwEpC,CAAU,IAA
V,CAAJ,C;;QAxehB,OAyEO,W;O;KAhFX,C;4FAUA,yB;MAAA,kF;MAAA,gE;MA+BA,wE;MA/BA,uC;QAO
W,kBAAa,eAAa,mCAAwB,EAAXB,CAAb,C;QAgCP,gB;QADb,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAA
a,sB;UACT,WAAy,WajC+C,SaiC3C,CAAU,oBAAmB,cAAnB,EAAMb,sBAAnB,UAAV,EAuC,IAAvC,CAAJ
,C;;QajChB,OAkCO,W;O;KAZCX,C;0GAUA,yB;MAAA,+D;MAoSA,wE;MapSA,uC;QAOW,kBAAoB,gB;QAO
Sd,gB;QADb,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UA1RsB,U;UAAA,cAVQ,SAUR,CA0RT,oB
AAmB,cAAnB,EAAMb,sBAAnB,UA1RS,EA0RoB,IA1RpB,W;YAA6C,6B;;;QAVhF,OAwo,W;O;KAIBX,C;8G
AUA,yB;MA0RA,wE;MA1RA,oD;QaiSiB,gB;QADb,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UA1
RsB,U;UAAA,wBA0RT,oBAAmB,cAAnB,EAAMb,sBAAnB,UA1RS,EA0RoB,IA1RpB,W;YAA6C,6B;;;QACHF,
OAAO,W;O;KARX,C;+FAWA,yB;MAAA,wE;MAAA,oD;QAQiB,UACoC,M;QAFjD,YAAY,C;QACC,2B;QAA
b,OAAa,cAAb,C;UAAa,sB;UACT,WAAy,WAAI,UAAU,oBAAmB,cAAnB,EAAMb,sBAAnB,UAAV,EAuC,IA
AvC,CAAJ,C;;QACHB,OAAO,W;O;KAVX,C;4FAaA,yB;MAAA,+D;MAAA,uC;QAOW,kBAAa,gB;QAwPJ,Q;Q
AAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UAhPK,U;UAAA,cARe,SAQf,CAGPQ,OAHPR,W;YAA5C,6B;;;
QAR3D,OASO,W;O;KAhBX,C;gGAUA,yB;MAAA,oD;QAqPoB,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAg
B,yB;UAhPK,U;UAAA,wBAGPQ,OAHPR,W;YAA5C,6B;;;QAC3D,OAAO,W;O;KANX,C;kFASA,6C;MAKiB,Q;
MAAA,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,WAAy,WAAI,UAAU,IAAV,CAAJ,C;;MACHB,OAAO,W;K;
IAQiB,4C;MAAA,mB;QAAE,gC;O;K;IAL9B,gC;MAKI,OAAO,qBAAiB,6BAAjB,C;K;IAGX,+B;MASI,OAA2B,
SAAf,eAAL,SAAK,CAAE,C;K;4FAG/B,yB;MAAA,2D;MAAA,+D;MAAA,sC;QAYc,Q;QAFV,UAAU,c;QACV,
WAAW,gB;QACD,2B;QAAV,OAAU,cAAV,C;UAAU,mB;UACN,UAAU,SAAS,CAAT,C;UACV,IAAI,GAAI,W
AAI,GAJ,CAAR,C;YACI,IAAK,WAAI,CAAJ,C;;QAEb,OAAO,I;O;KAjBX,C;IAoBA,uC;MAQI,UAAe,eAAL,S
AAK,C;MACX,YAAJ,GAAI,EAU,KAUV,C;MACJ,OAAO,G;K;IAGX,sC;MAMI,UAAe,eAAL,SAAK,C;MAC
X,YAAJ,GAAI,EAU,KAUV,C;MACJ,OAAO,G;K;IAGX,mC;MAMI,IAAN,I;MACH,kBADs,SACT,c;QAAoB
,4BAAc,SAAd,C;;QACZ,iCAAa,sBAAb,C;MAFZ,W;K;IAMJ,mC;MAUI,UAAe,eAAL,SAAK,C;MACX,OAAJ,G
AAI,EAao,KAAP,C;MACJ,OAAO,G;K;8EAGX,yB;MAAA,gD;MAAA,uC;QAooB,Q;QADhB,IAAI,wCAAsB,
mBAA1B,C;UAAqC,OAAO,I;QAC5B,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UAAM,IAAI,CAAC,UAAU,O
AAV,CAAL,C;YAAyB,OAAO,K;;QACtD,OAAO,I;O;KARX,C;IAWA,2B;MAMI,IAAI,oCAAJ,C;QAAwB,OAA
O,CAAC,mB;MACHC,OAAO,oBAAW,U;K;+EAGtB,yB;MAAA,gD;MAAA,uC;QAooB,Q;QADhB,IAAI,wCAA
sB,mBAA1B,C;UAAqC,OAAO,K;QAC5B,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UAAM,IAAI,UAAU,OAA
V,CAAJ,C;YAAwB,OAAO,I;;QACrD,OAAO,K;O;KARX,C;IAWA,6B;MAMoB,Q;MAFhB,IAAI,oCAAJ,C;QAA
wB,OAAO,c;MAC/B,YAAY,C;MACI,2B;MAAhB,OAAGB,cAAhB,C;QAAgB,yB;QAAM,oBAAMb,qBAAnB,E
AAmB,KAAAnB,E;;MACTB,OAAO,K;K;mFAGX,qB;MAKI,OAAO,c;K;mFAGX,yB;MAAA,gD;MAAA,wE;MAA
A,uC;QAMoB,Q;QAFhB,IAAI,wCAAsB,mBAA1B,C;UAAqC,OAAO,C;QAC5C,YAAY,C;QACI,2B;QAAhB,OA
AgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,oBAAMb,qBAAnB,EAAMb,KAAAnB,
E;;QAC9C,OAAO,K;O;KAPX,C;gFAUA,yC;MAUoB,Q;MADhB,kBAakB,O;MACF,2B;MAAhB,OAAGB,cAAh
B,C;QAAgB,yB;QAAM,cAAc,UAAU,WAAV,EAuB,OAAvB,C;;MACpC,OAAO,W;K;8FAGX,yB;MAAA,wE;
MAAA,gD;QAYoB,UAAiD,M;QAFjE,YAAY,C;QACZ,kBAakB,O;QACF,2B;QAAhB,OAAGB,cAAhB,C;UAAg
B,yB;UAAM,cAAc,UAAU,oBAAMb,cAAnB,EAAMb,sBAAnB,UAAV,EAuC,WAAvC,EAoD,OAApD,C;;Q
ACpC,OAAO,W;O;KAbX,C;0FAgBA,yC;MASI,kBAakB,O;MACIB,IAAI,CAAC,mBAAL,C;QACI,eAAe,+BAA
a,cAAb,C;QACf,OAAO,QAAS,cAAhB,C;UACI,cAAc,UAAU,QAAS,WAAAnB,EA+B,WAA/B,C;;;MAGtB,OA
AO,W;K;wGAGX,yC;MAUI,kBAakB,O;MACIB,IAAI,CAAC,mBAAL,C;QACI,eAAe,+BAAa,cAAb,C;QACf,O
AAO,QAAS,cAAhB,C;UACI,YAAY,QAAS,gB;UACrB,cAAc,UAAU,KAUV,EAiB,QAAS,WAA1B,EAAsC,W
AAtC,C;;MAGtB,OAAO,W;K;sFAGX,6B;MAKoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAgB,yB;QAA

M,OAAO,OAAP,C;;K;oGAG1B,yB;MAAA,wE;MAAA,oC;QAOiB,UAAgC,M;QAD7C,YAAY,C;QACC,2B;QA
Ab,OAAa,cAAb,C;UAAa,sB;UAAM,OAAO,oBAAmB,cAAnB,EAAmB,sBAAnB,UAAP,EAAoC,IAAP,C,C;;O;K
APvB,C;IAUA,0B;MAII,OAAO,sB;K;IAGX,2B;MAII,OAAO,uB;K;IAGX,2B;MAGI,OAAO,uB;K;kFAGX,+B;M
AGW,sB;;QAUP,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,qBAAO,I;UAAP,uB;;QACzB,cAAc,QA
AS,O;QACvB,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,qBAAO,O;UAAP,uB;;QACzB,eAdmB,QAcJ,CAAS,OAAT,
C;;UAEX,QAAQ,QAAS,O;UACjB,QAjBe,QAiBP,CAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAJ,C;YACI,
UAAU,C;YACV,WAAW,C;;QAED,QAAT,QAAS,W;QACIB,qBAAO,O;;;MAvBP,yB;K;8FAGJ,+B;MAOI,eAA
e,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHc,cAAc,QAAS,O;MACvB,IAAI,CAAC,QAAS
,UAAAd,C;QAAyB,OAAO,O;MACHc,eAAe,SAAS,OAAT,C;;QAEX,QAAQ,QAAS,O;QACjB,QAAQ,SAAS,CAA
T,C;QACR,IAAI,2BAAW,CAAX,KAJ,C;UACI,UAAU,C;UACV,WAAW,C;;MAED,QAAT,QAAS,W;MACIB,
OAAO,O;K;mFAGX,yB;MAAA,sE;MF/yDA,iB;ME+yDA,sC;QAaI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C
;UAAyB,MAAM,6B;QAC/B,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,
QAAS,OAAIB,C;UACR,WFzzDG,MAAO,KEyzDO,QFzzDP,EEyzDiB,CFzzDjB,C;;QE2zDd,OAAO,Q;O;KApB
X,C;mFAuBA,yB;MAAA,sE;MFj1DA,iB;MEi1DA,sC;QAaI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAy
B,MAAM,6B;QAC/B,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,
OAAIB,C;UACR,WF31DG,MAAO,KE21DO,QF31DP,EE21DiB,CF31DjB,C;;QE61Dd,OAAO,Q;O;KApBX,C;m
FAuBA,yB;MAAA,sE;MAAA,sC;QAWI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC
/B,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,I
AAI,2BAAW,CAAX,KAJ,C;YACI,WAAW,C;;QAGnB,OAAO,Q;O;KApBX,C;+FAuBA,yB;MFp3DA,iB;MEo
3DA,sC;QAWI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,I;QACHc,eAAe,SAAS,QAAS,OA
AIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,WF53DG,MAAO,KE43DO,
QF53DP,EE43DiB,CF53DjB,C;;QE83Dd,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MFp5DA,iB;MEo5DA,sC;QAWI,e
AAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,I;QACHc,eAAe,SAAS,QAAS,OAAIB,C;QACf,OA
AO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,WF55DG,MAAO,KE45DO,QF55DP,EE45Di
B,CF55DjB,C;;QE85Dd,OAAO,Q;O;KAlBX,C;+FAqBA,+B;MASI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C
;QAAyB,OAAO,I;MACHc,eAAe,SAAS,QAAS,OAAIB,C;MACf,OAAO,QAAS,UAAhB,C;QACI,QAAQ,SAAS,
QAAS,OAAIB,C;QACR,IAAI,2BAAW,CAAX,KAJ,C;UACI,WAAW,C;;MAGnB,OAAO,Q;K;0FAGX,yB;MA
AA,sE;MAAA,kD;QAWI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eAAe,SAAS
,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,IAAI,UAAW,S
AAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAT,C;YACI,WAAW,C;;QAGnB,OAAO,Q;O;KApBX,C;sGAu
BA,2C;MASI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHc,eAAe,SAAS,QAAS,OAAI
B,C;MACf,OAAO,QAAS,UAAhB,C;QACI,QAAQ,SAAS,QAAS,OAAIB,C;QACR,IAAI,UAAW,SAAQ,QAAR,E
AAkB,CAAIB,CAAX,GAakC,CAAT,C;UACI,WAAW,C;;MAGnB,OAAO,Q;K;IAGX,gC;MAOI,eAAe,oB;MA
Cf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHc,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;Q
ACI,QAAQ,QAAS,O;QACjB,MFn+DG,MAAO,KEm+DE,GFn+DF,EEem+DO,CFn+DP,C;;MEq+Dd,OAAO,G;K;
IAGX,iC;MAOI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHc,UAAU,QAAS,O;MAC
nB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,MF//DG,MAAO,KE+/DE,GF//DF,EE+/DO,CF//DP,
C;;MEigEd,OAAO,G;K;IAGX,iC;MAKI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHc
,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,sBAAM,CAAN,KAJ,
C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,0C;MAGI,OAAO,2BAAc,UAAAd,C;K;IAGX,gD;MAKI,eAAe,oB
;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHc,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB
,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,UAAW,SAAQ,GAAR,EAaA,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,
MAAM,C;;MAE9C,OAAO,G;K;IAGX,0B;MAII,OAAO,sB;K;IAGX,2B;MAII,OAAO,uB;K;IAGX,2B;MAGI,OA
AO,uB;K;kFAGX,+B;MAGW,sB;;QAUP,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,qBAAO,I;UAAP
,uB;;QACzB,cAAc,QAAS,O;QACvB,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,qBAAO,O;UAAP,uB;;QACzB,eAdm
B,QAcJ,CAAS,OAAT,C;;UAEX,QAAQ,QAAS,O;UACjB,QAjBe,QAiBP,CAAS,CAAT,C;UACR,IAAI,2BAAW,
CAAX,KAJ,C;YACI,UAAU,C;YACV,WAAW,C;;QAED,QAAT,QAAS,W;QACIB,qBAAO,O;;;MAvBP,yB;K;
8FAGJ,+B;MAOI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHc,cAAc,QAAS,O;MAC

vB,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,O;MACHC,eAAe,SAAS,OAAT,C;;QAEX,QAAQ,QAAS,O;QACjB,QAAQ,SAAS,CAAT,C;QACR,IAAI,2BAAW,CAAX,KAAJ,C;UACI,UAAU,C;UACV,WAAW,C;;;MAED,Q AAT,QAAS,W;MACIB,OAAO,O;K;mFAGX,yB;MAAA,sE;MFI4DA,iB;MEk4DA,sC;QAaI,eAAe,oB;QACf,IAA I,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAh B,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,WF54DG,MAAO,KE44DO,QF54DP,EE44DiB,CF54DjB,C;;Q E84Dd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MFp6DA,iB;MEo6DA,sC;QAaI,eAAe,oB;QACf,IAAI,C AAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C ;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,WF96DG,MAAO,KE86DO,QF96DP,EE86DiB,CF96DjB,C;;QEg7 Dd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAAA,sC;QAWI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd ,C;UAAyB,MAAM,6B;QAC/B,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SA AS,QAAS,OAAIB,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;+ FAuBA,yB;MFv8DA,iB;MEu8DA,sC;QAWI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,I;QA ChC,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR ,WF/8DG,MAAO,KE++8DO,QF/8DP,EE++8DiB,CF/8DjB,C;;QEi9Dd,OAAO,Q;O;KAIBX,C;+FAqBA,yB;MFv+D A,iB;MEu+DA,sC;QAWI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,I;QAChC,eAAe,SAAS,Q AAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,WF/+DG,MAAO, KE++DO,QF/+DP,EE++DiB,CF/+DjB,C;;QEi/Dd,OAAO,Q;O;KAIBX,C;+FAqBA,+B;MASI,eAAe,oB;MACf,IA AI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHC,eAAe,SAAS,QAAS,OAAIB,C;MACf,OAAO,QAAS,UAAh B,C;QACI,QAAQ,SAAS,QAAS,OAAIB,C;QACR,IAAI,2BAAW,CAAX,KAAJ,C;UACI,WAAW,C;;;MAGnB,OA AO,Q;K;0FAGX,yB;MAAA,sE;MAAA,kD;QAWI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM ,6B;QAC/B,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C ;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAA O,Q;O;KApBX,C;sGAuBA,2C;MASI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHC,e AAe,SAAS,QAAS,OAAIB,C;MACf,OAAO,QAAS,UAAhB,C;QACI,QAAQ,SAAS,QAAS,OAAIB,C;QACR,IAAI ,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;UACI,WAAW,C;;;MAGnB,OAAO,Q;K;IAGX, gC;MAOI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHC,UAAU,QAAS,O;MACnB,OA AO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,MFtjEG,MAAO,KEsjEE,GFtjEF,EEsjEO,CFtjEP,C;;MEwj Ed,OAAO,G;K;IAGX,iC;MAOI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHC,UAAU, QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,MFIIEG,MAAO,KEkIEE,GFIIIEF,EE kIEO,CFIIIEP,C;;MEoIEd,OAAO,G;K;IAGX,iC;MAKI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OA AO,I;MACHC,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,sBAAM, CAAN,KAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,0C;MAGI,OAAO,2BAAc,UAAAd,C;K;IAGX,gD;M AKI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHC,UAAU,QAAS,O;MACnB,OAAO, QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAA jC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,4B;MAMI,IAAI,oCAAJ,C;QAAwB,OAAO,mB;MAC/B,OA AO,CAAC,oBAAW,U;K;iFAGvB,yB;MAAA,gD;MAAA,uC;QAOoB,Q;QADhB,IAAI,wCAAsB,mBAA1B,C;UA AqC,OAAO,I;QAC5B,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,UAAU,OAAV,CAAJ,C;YAAwB, OAAO,K;;QACrD,OAAO,I;O;KARX,C;oFAWA,6B;MAKmC,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB, yB;QAAM,OAAO,OAAP,C;;MAArC,gB;K;kGAGJ,yB;MAAA,6B;MAAA,sC;MArNB,wE;MAqnBA,2BAQiB,y B;QA7nBjB,wE;eA6nBiB,0B;UAAA,4B;YAAE,aAAe,c;YAtNBjB,gB;YADb,YAAY,C;YACC,2B;YAAb,OAAa,c AAb,C;cAAa,sB;cAAM,OAAO,oBAAmB,cAAnB,EAAMb,sBAAnB,UAAP,EAAoC,IAApC,C;;YAsnBmB,W;W; S;OAAzB,C;MARjB,oC;QA9mBiB,gB;QADb,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UAAM,OAA O,oBAAmB,cAAnB,EAAMb,sBAAnB,UAAP,EAAoC,IAApC,C;;QAsnBnB,gB;O;KARJ,C;oFAWA,yB;MAAA,4 F;MAAA,uC;QAaI,eAAe,SAAK,W;QACpB,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,mCAA8B,oCAA9B,C ;QAC/B,kBAAqB,QAAS,O;QAC9B,OAAO,QAAS,UAAhB,C;UACI,cAAc,UAAU,WAAV,EAAB,QAAS,OAAh C,C;;QAEIB,OAAO,W;O;KANBX,C;kGAsBA,yB;MAAA,4F;MAAA,wE;MAAA,uC;QAKBmD,Q;QAL/C,eAAe,S AAK,W;QACpB,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,mCAA8B,oCAA9B,C;QAC/B,YAAY,C;QACZ,k BAAqB,QAAS,O;QAC9B,OAAO,QAAS,UAAhB,C;UACI,cAAc,UAAU,oBAAmB,YAAnB,EAAMb,oBAAnB,Q

AAV,EAAuC,WAAvC,EAAoD,QAAS,OAA7D,C;;QAEIB,OAAO,W;O;KApBX,C;8GAuBA,yB;MAAA,wE;MAAA,uC;QAKBmD,Q;QAL/C,eAAe,SAAK,W;QACpB,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,I;QAChC,YAAY,C;QACZ,kBAAqB,QAAS,O;QAC9B,OAAO,QAAS,UAAhB,C;UACI,cAAc,UAAU,oBAAmB,YAAAnB,EAAmB,oBAAAnB,QAAV,EAAuC,WAAvC,EAAoD,QAAS,OAA7D,C;;QAEIB,OAAO,W;O;KApBX,C;gGAuBA,gC;MAcI,eAAe,SAAK,W;MACpB,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MAChC,kBAAqB,QAAS,O;MAC9B,OAAO,QAAS,UAAhB,C;QACI,cAAc,UAAU,WAAV,EAAuB,QAAS,OAAhC,C;;MAEIB,OAAO,W;K;8FAGX,yB;MAAA,4F;MAAA,uC;QAaI,eAAe,+BAAa,cAAb,C;QACf,IAAI,CAAC,QAAS,cAAAd,C;UACI,MAAM,mCAA8B,8BAA9B,C;QACV,kBAAqB,QAAS,W;QAC9B,OAAO,QAAS,cAAhB,C;UACI,cAAc,UAAU,QAAS,WAAAnB,EAA+B,WAA/B,C;;QAEIB,OAAO,W;O;KApBX,C;4GAuBA,yB;MAAA,4F;MAAA,uC;QAaI,eAAe,+BAAa,cAAb,C;QACf,IAAI,CAAC,QAAS,cAAAd,C;UACI,MAAM,mCAA8B,8BAA9B,C;QACV,kBAAqB,QAAS,W;QAC9B,OAAO,QAAS,cAAhB,C;UACI,YAAY,QAAS,gB;UACrB,cAAc,UAAU,KAAV,EAAiB,QAAS,WAA1B,EAA5C,WAA1C,C;;QAEIB,OAAO,W;O;KArBX,C;wHAwBA,gC;MAaI,eAAe,+BAAa,cAAb,C;MACf,IAAI,CAAC,QAAS,cAAAd,C;QACI,OAAO,I;MACX,kBAAqB,QAAS,W;MAC9B,OAAO,QAAS,cAAhB,C;QACI,YAAY,QAAS,gB;QACrB,cAAc,UAAU,KAAV,EAAiB,QAAS,WAA1B,EAA5C,WAA1C,C;;MAEIB,OAAO,W;K;0GAGX,gC;MAcI,eAAe,+BAAa,cAAb,C;MACf,IAAI,CAAC,QAAS,cAAAd,C;QACI,OAAO,I;MACX,kBAAqB,QAAS,W;MAC9B,OAAO,QAAS,cAAhB,C;QACI,cAAc,UAAU,QAAS,WAAAnB,EAA+B,WAA/B,C;;MAEIB,OAAO,W;K;8FAGX,yB;MAAA,kF;MAAA,gD;MAAA,gE;MAAA,gD;QAiBoB,Q;QAJhB,oBAAoB,mCAAwB,CAAxB,C;QACpB,IAAI,kBAAiB,CAArB,C;UAAwB,OAAO,OAAO,OAAP,C;QACc,kBAAhC,eAAa,gBAAgB,CAAhB,IAAb,C;QAAwC,8B;QAARd,aHjjFO,W;QGkjFP,kBAAkB,O;QACF,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KArBX,C;4GAwBA,yB;MAAA,kF;MAAA,gD;MAAA,gE;MAAA,gD;QAmBoB,UACY,M;QAN5B,oBAAoB,mCAAwB,CAAxB,C;QACpB,IAAI,kBAAiB,CAArB,C;UAAwB,OAAO,OAAO,OAAP,C;QACc,kBAAhC,eAAa,gBAAgB,CAAhB,IAAb,C;QAAwC,8B;QAARd,aH1kFO,W;QG2kFP,YAAY,C;QACZ,kBAAkB,O;QACF,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,cAAc,WAAU,cAAV,EAAU,sBAAV,WAAmB,WAAAnB,EAAgC,OAAhC,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAvBX,C;kGA0BA,yB;MAAA,qD;MAAA,kF;MAAA,gE;MAAA,uC;QAcI,eAAe,SAAK,W;QACpB,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,W;QAChC,sBAAqB,QAAS,OAA9B,C;QACuD,kBAA1C,eAAa,mCAAwB,EAAxB,CAAb,C;QAAkD,sBAAI,aAAJ,C;QAA/D,aHrmFO,W;QGsmFP,OAAO,QAAS,UAAhB,C;UACI,gBAAc,UAAU,aAAV,EAAuB,QAAS,OAAhC,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KATBX,C;gHAyBA,yB;MAAA,qD;MAAA,kF;MAAA,gE;MAAA,uC;QAoBgC,Q;QAN5B,eAAe,SAAK,W;QACpB,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,W;QAChC,sBAAqB,QAAS,OAA9B,C;QACuD,kBAA1C,eAAa,mCAAwB,EAAxB,CAAb,C;QAAkD,sBAAI,aAAJ,C;QAA/D,aH9nFO,W;QG+nFP,YAAY,C;QACZ,OAAO,QAAS,UAAhB,C;UACI,gBAAc,WAAU,YAAV,EAAU,oBAAV,SAAmB,aAAAnB,EAAgC,QAAS,OAAzC,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAvBX,C;gFA0BA,yB;MArGA,kF;MAAA,gD;MAAA,gE;MAqGA,gD;QAcW,sB;;UAIGS,Q;UAJhB,oBAAoB,mCAAwB,CAAxB,C;UACpB,IAAI,kBAAiB,CAArB,C;YAAwB,qBAAO,OAqGZ,OArGY,C;YAAP,uB;;UACqB,kBAAhC,eAAa,gBAAgB,CAAhB,IAAb,C;UAAwC,sBAoGIC,OApGkC,C;UAArD,aHjjFO,W;UGkjFP,kBAAmGmB,O;UAIGH,2B;UAAhB,OAAgB,cAAhB,C;YAAgB,yB;YACZ,cAiGwB,SAjGV,CAAU,WAAV,EAAuB,OAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QA8FP,yB;O;KAdJ,C;8FAiBA,yB;MA9FA,kF;MAAA,gD;MAAA,gE;MA8FA,gD;QAeW,6B;;UA1FS,gB;UALhB,oBAAoB,mCAAwB,CAAxB,C;UACpB,IAAI,kBAAiB,CAArB,C;YAAwB,4BAAO,OA8FL,OA9FK,C;YAAP,8B;;UACqB,kBAAhC,eAAa,gBAAgB,CAAhB,IAAb,C;UAAwC,sBA6F3B,OA7F2B,C;UAArD,aH1kFO,W;UG2kFP,YAAY,C;UACZ,kBA2F0B,O;UA1FV,2B;UAAhB,OAAgB,cAAhB,C;YAAgB,yB;YACZ,cAyF+B,SAzFjB,EAAU,cAAV,EAAU,sBAAV,WAAmB,WAAAnB,EAAgC,OAAhC,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;;QAsFP,gC;O;KafJ,C;kFAkBA,+B;MAOoB,Q;MADhB,UAAe,C;MACC,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QACZ,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;8FAGX,+B;MAOoB,Q;MADhB,UAAkB,G;MACF,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QACZ,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;mFAGX,+B;MAUoB,Q;MADhB,UAAoB,C;MACJ,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QACZ,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;mFAGX,+B;MAUoB,Q;MADhB,UAAe,C;MACC,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QACZ,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;mFAGX,yB;MAAA,SASoB,gB;MATpB,sC;QAUoB,Q;QADh

B, Y; QACgB, 2B; QAAhB, OAAgB, cAAhB, C; UAAgB, yB; UACZ, cAAO, SAAS, OAAT, CAAP, C;; QAEJ, OAAO, G; O; KAbX, C; mFAGBA, yB; MjB/7EA, 6B; MiB+7EA, sC; QAWoB, Q; QADhB, UjB/7EmC, ciB+7EnB, CjB/7EmB, C; QiBg 8EnB, 2B; QAAhB, OAAgB, cAAhB, C; UAAgB, yB; UACZ, MjBnwFiD, ciBmwFjD, GjBnwF2D, KAAK, GiBmwFzD, S AAS, OAAT, CjBnwFoE, KAAX, IAaf, C;; QiBqwFrD, OAAO, G; O; KAdX, C; mFAiBA, yB; MD78EA, +B; MC68EA, s C; QAWoB, Q; QADhB, UD58EqC, eAAW, oBC48E/B, CD58E+B, CAAX, C; QC68ErB, 2B; QAAhB, OAAgB, cAAhB, C; UAAgB, yB; UACZ, MDjxFmD, eCixFnD, GDjxF8D, KAAK, KCixF5D, SAAS, OAAT, CDjxFuE, KAAX, CAAhB, C; ; QCmxFvD, OAAO, G; O; KAdX, C; IAiBA, qC; MAIoB, UAMT, M; MANS, 2B; MAAhB, OAAgB, cAAhB, C; QAAgB, y B; QACZ, IAAI, eAAJ, C; UACI, MAAM, gCAAYB, 2BAAwB, SAAxB, MAAzB, C;; MAId, OAAO, mE; K; IAGX, qC; M AIoB, UAMT, M; MANS, 2B; MAAhB, OAAgB, cAAhB, C; QAAgB, yB; QACZ, IAAI, eAAJ, C; UACI, MAAM, gCAAYB , 2BAAwB, SAAxB, MAAzB, C;; MAId, OAAO, +D; K; IAGX, kC; MAWI, OAAO, oBAAS, IAAT, EAAe, IAaf, EAAsC, IAATc, C; K; IAGX, +C; MAgBI, OAAO, sBAAS, IAAT, EAAe, IAaf, EAAsC, IAATc, EAAwD, SAAxD, C; K; IAGX, mC; MAII, aAAa, iBAaA, mCAAwB, EAAxB, CAAb, C; MACb, kBAAc, KAAd, C; MAnIeG, Q; MAAA, OAoIET, SAplES, W; MAAhB, OAAgB, cAAhB, C; QAAgB, 2B; QAAU, oB; QAoIEK, IAAI, CAAC, SAAD, IAAY, OAplEX, SAoIEW, UA AhB, C; UAAiC, YAAU, I; UAA3C, mBAAiD, K;; UAAjD, mBAA8D, I;; QAplEvE, qB; UAoIED, MAplEqC, WAAI, SAA J, C;; MAoIEID, OAAqB, M; K; IAGzB, sC; MAQI, IAAI, QpB0yJG, YAAQ, CoB1yJf, C; QAAwB, OAAy, SAAL, SAAK, C; MACpC, YAAqB, 8BAAT, QAAS, C; MAtoEd, kBAAY, gB; MA4BH, Q; MAAA, OA2mET, SA3mES, W; MAAhB, O AAgB, cAAhB, C; QAAgB, yB; QAAM, IAAI, CA2mEF, qBA3mEa, OA2mEb, CA3mEF, C; UAAyB, WAAY, WAAI, O AAJ, C;; MA2mE3D, OA1mEO, W; K; IA6mEX, sC; MAQI, YAAqB, gCAAT, QAAS, EAAGC, SAAhC, C; MACrB, IAAI, KAAM, UAAV, C; QACI, OAAy, SAAL, SAAK, C; MAppET, kBAAY, gB; MA4BH, Q; MAAA, OAynET, SAznES, W; MAAhB, OAAgB, cAAhB, C; QAAgB, yB; QAAM, IAAI, CAynEF, qBAznEa, OAynEb, CAznEF, C; UAAyB, WAAY, W AAI, OAAJ, C;; MAynE3D, OAxnEO, W; K; IA2nEX, sC; MAQI, YAAqB, 8BAAT, QAAS, C; MACrB, IAAI, KAAM, UA AV, C; QACI, OAAy, SAAL, SAAK, C; MAIqET, kBAAY, gB; MA4BH, Q; MAAA, OAuoET, SAvoES, W; MAAhB, OA AgB, cAAhB, C; QAAgB, yB; QAAM, IAAI, CAuoEF, qBAvoEa, OAuoEb, CAvoEF, C; UAAyB, WAAY, WAAI, OAAJ, C;; MAuoE3D, OAtoEO, W; K; 8FAyoEX, yB; MAAA, 8C; MAAA, qC; QAKI, OAAO, iBAAM, OAAAN, C; O; KALX, C; 0F AQA, yB; MAAA, +D; MAAA, 6B; MAAA, uC; QAUoB, Q; QAFhB, YAAy, gB; QACZ, aAAa, gB; QACG, 2B; QAAhB, O AAgB, cAAhB, C; UAAgB, yB; UACZ, IAAI, UAAU, OAAV, CAAJ, C; YACI, KAAM, WAAI, OAAJ, C;; YAEN, MAOO, WAAI, OAAJ, C;; QAGf, OAAO, cAAK, KAAL, EAAY, MAAZ, C; O; KAjBX, C; IAoBA, kC; MAII, IAAI, oCAAJ, C; QA AwB, OAAy, OAAL, SAAK, EAAK, OAAL, C; MACpC, aAAa, gB; MACN, OAAP, MAAO, EAAO, SAAP, C; MACP, M AAO, WAAI, OAAJ, C; MACP, OAAO, M; K; IAGX, oC; MAII, aAAa, iBAaA, iBAAO, CAAP, IAAb, C; MACb, MAAO, g BAAO, SAAP, C; MACP, MAAO, WAAI, OAAJ, C; MACP, OAAO, M; K; IAGX, qC; MAII, IAAI, oCAAJ, C; QAAwB, O AAY, OAAL, SAAK, EAAK, QAAL, C; MACpC, aAAa, gB; MACN, OAAP, MAAO, EAAO, SAAP, C; MACA, SAAP, M AAO, EAAO, QAAP, C; MACP, OAAO, M; K; IAGX, qC; MAII, aAAa, iBAaA, SAAK, KAAL, GAAY, QAAS, OAArB, IA Ab, C; MACb, MAAO, gBAAO, SAAP, C; MACA, SAAP, MAAO, EAAO, QAAP, C; MACP, OAAO, M; K; IAGX, qC; MA II, IAAI, oCAAJ, C; QAAwB, OAAy, OAAL, SAAK, EAAK, QAAL, C; MACpC, aAAa, gB; MACN, OAAP, MAAO, EAA O, SAAP, C; MACA, OAAP, MAAO, EAAO, QAAP, C; MACP, OAAO, M; K; IAGX, qC; MAII, IAAI, mCAAJ, C; QACI, a AAa, iBAaA, SAAK, KAAL, GAAY, QAAS, KAArB, IAAb, C; QACb, MAAO, gBAAO, SAAP, C; QACP, MAAO, gBAA O, QAAP, C; QACP, OAAO, M;; QAEP, eAAa, iBAaA, SAAb, C; QACN, OAAP, QAAO, EAAO, QAAP, C; QACP, OAAO , Q;; K; IAIf, qC; MAII, aAAa, gB; MACN, OAAP, MAAO, EAAO, SAAP, C; MACA, SAAP, MAAO, EAAO, QAAP, C; MA CP, OAAO, M; K; IAGX, qC; MAII, aAAa, iBAaA, SAAK, KAAL, GAAY, EAAZ, IAAb, C; MACb, MAAO, gBAAO, SAA P, C; MACA, SAAP, MAAO, EAAO, QAAP, C; MACP, OAAO, M; K; 4FAGX, yB; MAAA, 4C; MAAA, qC; QAKI, OAAO, gBAAK, OAAL, C; O; KALX, C; 8FAQA, yB; MAAA, 4C; MAAA, qC; QAKI, OAAO, gBAAK, OAAL, C; O; KALX, C; IA QA, yD; MAgB+C, oB; QAAA, OAAy, C; MAAG, 8B; QAAA, iBAA0B, K; MAOzE, Q; MANX, oBAAoB, IAAPB, EAAO B, IAA1B, C; MACA, IAAI, 0CAAwB, 8BAA5B, C; QACI, eAAe, SAAK, K; QACpB, qBAAqB, YAAW, IAAX, SAAsB, WAAW, IAAX, KAAM, CAAvB, GAA0B, CAA1B, GAAiC, CAAnD, K; QACrB, aAAa, iBAAM, cAAAN, C; QACb, g BAAY, CAAZ, C; QACA, Y; UAAO, c; UAAP, MAAGB, CAAT, mBAAiB, QAAxB, E; YAAA, K; UACI, iBAAsB, eAAL, IAAK, EAAa, WAAW, OAAX, IAAb, C; UACtB, IAAI, aAAa, IAAb, IAAqB, CAAC, cAA1B, C; YAA0C, K; Ud59FID, W AAW, iBc69Fa, Ud79Fb, C; UWCX, mBAAc, CAAd, YG49FwB, UH59FxB, Y; YXA6B, ec49FS, sBH39F3B, OG29FgC, GAAK, OAAL, IAAL, Cd59FT, C;; Uc49FrB, MAAO, Wd39FR, Ic29FQ, C; UACP, oBAAS, IAAT, I;; QAEJ, OAAO, M;;

MAEX,eAAa,gB;MACiE,kBAA9E,iBAAiB,oBAAjB,EAA6B,IAA7B,EAAmC,IAAnC,EAAyC,cAAzC,EAAuE,KAAvE,C;ME5lGA,OAAgB,qBAAhB,C;QAAGB,gC;QF6lGL,mBE7lGqB,OF6lGrB,C;;MAEX,OAAO,Q;K;IAGX,sE;MAkBkD,oB;QAAA,OAAy,C;MAAG,8B;QAAA,iBAA0B,K;MACvF,oBAAoB,IAApB,EAA0B,IAA1B,C;MACA,IAAI,0CAAwB,8BAA5B,C;QACI,eAAe,SAAK,K;QACpB,qBAAqB,YAAW,IAAX,SAAsB,WAAW,IAAX,KAAmB,CAAvB,GAA0B,CAA1B,GAAiC,CAAnD,K;QACrB,aAAa,iBAAa,cAAb,C;QACb,eAAa,kBAAc,SAAd,C;QACb,YAAy,C;QACZ,OAAgB,CAAT,qBAAiB,QAAxB,C;UACI,iBAAAsB,eAAL,IAAK,EAAa,WAAW,KAAX,IAAb,C;UACtB,IAAI,CAAC,cAAD,IAAmB,aAAa,IAApC,C;YAA0C,K;UAC1C,QAAO,cAAK,KAAL,EAAy,QA AQ,UAAAR,IAAZ,C;UACP,MAAO,WAAI,UAAU,QAAV,CAAJ,C;UACP,gBAAS,IAAT,I;;QAEJ,OAAO,M;;MAEX,eAAa,gB;MACgE,kBAA7E,iBAAiB,oBAAjB,EAA6B,IAA7B,EAAmC,IAAnC,EAAyC,cAAzC,EAAuE,IAAvE,C;MEtoGA,OAAgB,qBAAhB,C;QAAGB,gC;QFuoGL,mBAAI,UEvoGiB,OFuoGjB,CAAJ,C;;MAEX,OAAO,Q;K;IAGX,kC;MAqBoB,gB;MAHhB,gBAXW,KAWW,O;MACtB,WAAW,iBF17FJ,MAAO,KE07FgB,mCAAwB,EAAxB,CF17FhB,EE07F6C,SF17F7C,CE07FH,C;MACX,QAAQ,C;MACQ,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QACZ,IAAI,KAAC,SAAT,C;UAAoB,K;QACpB,IAAK,WAhBqB,GAgBP,OAhBO,EAAhB,KAAGBqB,CAAM,UAAAN,EAAM,kBAAN,SAhBF,CAGrB,C;;MAhBT,OAKBO,I;K;+EafX,yB;MAAA,kF;MAAA,gE;MFv7FA,iB;MEu7FA,8C;QAWoB,UAEsB,M;QALtC,gBAAgB,KAAM,O;QACtB,WAAW,eF17FJ,MAAO,KE07FgB,mCAAwB,EAAxB,CF17FhB,EE07F6C,SF17F7C,CE07FH,C;QACX,QAAQ,C;QACQ,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAAI,KAAC,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,OAAV,EAAmB,MAAM,UAAAN,EAAM,kBAAN,SAANB,CAAJ,C;;QAET,OAAO,I;O;KafX,C;IAkBA,kC;MAkBI,YAAy,oB;MACZ,aAZW,KAYQ,W;MACnB,WAAW,iBFv9FJ,MAAO,KEu9FgB,mCAAwB,EAAxB,CFv9FhB,EEu9FmD,wBAbtD,KAAsD,EA AwB,EAAxB,CFv9FnD,CEu9FH,C;MACX,OAAO,KAAM,UAAAN,IAAmB,MAAO,UAAjC,C;QACI,IAAK,WafqB,GAeP,KAAM,OafC,EAeO,MAAO,Oafd,CAerB,C;;MAfT,OAIBo,I;K;+EAdX,yB;MAAA,kF;MAAA,gE;MFn9FA,iB;MEm9FA,8C;QAQI,YAAy,oB;QACZ,aAAa,KAAM,W;QACnB,WAAW,eFv9FJ,MAAO,KEu9FgB,mCAAwB,EAAxB,CFv9FhB,EEu9FmD,wBAAN,KAAM,EAAwB,EAAxB,CFv9FnD,CEu9FH,C;QACX,OAAO,KAAM,UAAAN,IAAmB,MAAO,UAAjC,C;UACI,IAAK,WAAI,UAAU,KAAM,OAAhB,EAAwB,MAAO,OAA/B,CAAJ,C;;QAET,OAAO,I;O;KAdX,C;IAiBA,gC;MASW,sB;;QAaP,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,qBAAO,W;UAAP,uB;;QACzB,ad/pGoD,gB;QcggGpD,cAAc,QAAS,O;QACvB,OAAO,QAAS,UAAhB,C;UACI,WAAW,QAAS,O;UACpB,MAAO,WAnBkB,GAmBJ,OAnBI,EAmBK,IAnBL,CAmBIB,C;UACP,UAAU,I;;QAE d,qBAAO,M;;MATBP,yB;K;8FAGJ,yB;MAAA,qD;MdZpGA,+D;McypGA,uC;QAUI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,W;QACChC,ad/pGoD,gB;QcggGpD,cAAc,QAAS,O;QACvB,OAAO,QAAS,UAAhB,C;UACI,WAAW,QAAS,O;UACpB,MAAO,WAAI,UAAU,OAAV,EAAmB,IAAnB,CAAJ,C;UACP,UAAU,I;;QAE d,OAAO,M;O;KAnBX,C;IASBA,8F;MAQ6D,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAoC,I;MAGtN,Q;MAFhB,MAAO,gBAAO,MAAP,C;MACP,YAAy,C;MACI,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QACZ,IAAI,iCAAU,CAAd,C;UAAiB,MAAO,gBAAO,SAAP,C;QACxB,IAAI,QAAQ,CAAR,IAAa,SAAS,KAA1B,C;UACW,gBAAP,MAAO,EAAC,OAAd,EAAuB,SAAvB,C;;UACJ,K;;MAEX,IAAI,SAAS,CAAT,IAAc,QAAQ,KAA1B,C;QAAiC,MAAO,gBAAO,SAAP,C;MACxC,MAAO,gBAAO,OAAP,C;MACP,OAAO,M;K;IAGX,4F;MAQwC,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAoC,I;MACjN,OAAO,oBAAO,sBAAP,EAAwB,SAAXB,EAAmC,MAAnC,EAA2C,OAA3C,EAAoD,KAApD,EAA2D,SAA3D,EAASe,SAAtE,CAAI f,W;K;4FAG5F,qB;MAKl,OA AO,S;K;IASS,8C;MAAA,mB;QAAE,OAAA,eAAK,W;O;K;IAN3B,iC;MAMI,oCAAgB,8BAAhB,C;K;IAGJ,+B;MAOoB,Q;MAFhB,UAAkB,G;MACIB,YAAiB,C;MACD,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QACZ,OA AO,O;QACP,oBAAmB,qBAAnB,EAAmB,KAAhB,E;;MAEJ,OAAW,UAAAS,CAAb,GAAGB,wCAAo,IAAvB,GA AgC,MAAM,K;K;IAGjD,+B;MAOoB,Q;MAFhB,UAAkB,G;MACIB,YAAiB,C;MACD,2B;MAAhB,OAAgB,cAA hB,C;QAAGB,yB;QACZ,OAAO,O;QACP,oBAAmB,qBAAnB,EAAmB,KAAhB,E;;MAEJ,OAAW,UAAAS,CAAb, GAAGB,wCAAo,IAAvB,GAAGC,MAAM,K;K;IAGjD,+B;MAOoB,Q;MAFhB,UAAkB,G;MACIB,YAAiB,C;MA CD,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QACZ,OAAO,O;QACP,oBAAmB,qBAAnB,EAAmB,KAAhB,E;; MAEJ,OAAW,UAAAS,CAAb,GAAGB,wCAAo,IAAvB,GAAGC,MAAM,K;K;IAGjD,+B;MAOoB,Q;MAFhB,UAA kB,G;MACIB,YAAiB,C;MACD,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QACZ,OAAO,O;QACP,oBAAmB,q

,OAAO,K;;MACtD,OAAO,I;K;IAGX,2B;MAMI,OAAO,CAAC,mB;K;+EAGZ,gC;MAOoB,Q;MADhB,IAAI,mB
AAJ,C;QAAe,OAAO,K;MACN,OAAA,SjB6LoE,QAAQ,W;MiB7L5F,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IA
AI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,I;;MACrD,OAAO,K;K;mFAGX,qB;MAKI,OAAO,c;K;mFAGX,gC;
MAMoB,Q;MAFhB,IAAI,mBAAJ,C;QAAe,OAAO,C;MACtB,YAAY,C;MACI,OAAA,SjB2KoE,QAAQ,W;MiB3
K5F,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,qB;;MAC9C,OAAO,K;K;sFA
GX,6B;MAKoB,Q;MAAA,OAAA,SjBkKoE,QAAQ,W;MiBIK5F,OAAgB,cAAhB,C;QAAgB,yB;QAAM,OAAO,
OAAp,C;;K;kFAG1B,+B;MAemB,kBAAR,iB;MAAQ,sB;;QJkoDf,eAAe,sB;QACf,IAAI,CAAC,QAAS,UAd,C;
UAAyB,qBAAO,I;UAAP,uB;;QACzB,cAAc,QAAS,O;QACvB,IAAI,CAAC,QAAS,UAd,C;UAAyB,qBAAO,O;
UAAP,uB;;QACzB,eIjpDmB,QJipDJ,CAAS,OAAT,C;;UAEX,QAAQ,QAAS,O;UACjB,QIppDe,QJopDP,CAAS,C
AAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;QAED,QAAT,QAAS,W;QAC
IB,qBAAO,O;;MI1pDP,yB;K;8FAGJ,+B;MAQmB,kBAAR,iB;MAAQ,sB;;QJkoDf,eAAe,sB;QACf,IAAI,CAAC,
QAAS,UAd,C;UAAyB,qBAAO,I;UAAP,uB;;QACzB,cAAc,QAAS,O;QACvB,IAAI,CAAC,QAAS,UAd,C;UA
AyB,qBAAO,O;UAAP,uB;;QACzB,eItoD2B,QJsoDZ,CAAS,OAAT,C;;UAEX,QAAQ,QAAS,O;UACjB,QIzoDuB
,QJyoDf,CAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;QAED,QAAT,
QAAS,W;QACIB,qBAAO,O;;MI/oDP,yB;K;mFAGJ,yB;MJ+oDA,sE;MF/yDA,iB;MMgKA,sC;QJ4pDI,eI/oDO,
iBJ+oDQ,W;QACf,IAAI,CAAC,QAAS,UAd,C;UAAyB,MAAM,6B;QAC/B,eIjpDqB,QJipDN,CAAS,QAAS,OA
AIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QInpDiB,QJmpDT,CAAS,QAAS,OAAIB,C;UACR,WFzzDG,MAA
O,KEyzDO,QFzzDP,EEyzDiB,CFzzDjB,C;;QMqKd,OJspDO,Q;O;KInqDX,C;mFagBA,yB;MJspDA,sE;MFj1DA,
iB;MM2LA,sC;QJmqDI,eItpDO,iBJspDQ,W;QACf,IAAI,CAAC,QAAS,UAd,C;UAAyB,MAAM,6B;QAC/B,eI
pDqB,QJwpDN,CAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QI1pDiB,QJ0pDT,CAAS,QAAS,
OAAIB,C;UACR,WF31DG,MAAO,KE21DO,QF31DP,EE21DiB,CF31DjB,C;;QMgMd,OJ6pDO,Q;O;KI1qDX,C;
mFagBA,yB;MJ6pDA,sE;MI7pDA,sC;QJwqDI,eI7pDO,iBJ6pDQ,W;QACf,IAAI,CAAC,QAAS,UAd,C;UAAyB
,MAAM,6B;QAC/B,eI/pDqB,QJ+pDN,CAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QIjqDiB,Q
JiqDT,CAAS,QAAS,OAAIB,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;QInqDnB,OJsqDO,Q;O;
KIjrDX,C;+FACa,yB;MN9MA,iB;MM8MA,sC;QAWmB,kBAAR,iB;QAAQ,sB;;UJsqDf,eAAe,sB;UACf,IAAI,C
AAC,QAAS,UAd,C;YAAyB,qBAAO,I;YAAP,uB;;UACzB,eIxd2B,QJwqDZ,CAAS,QAAS,OAAIB,C;UACf,O
AAO,QAAS,UAAhB,C;YACI,QI1qDuB,QJ0qDf,CAAS,QAAS,OAAIB,C;YACR,WF53DG,MAAO,KE43DO,QF5
3DP,EE43DiB,CF53DjB,C;;UE83Dd,qBAAO,Q;;QI7qDP,yB;O;KAXJ,C;+FACa,yB;MNvOA,iB;MMuOA,sC;QA
WmB,kBAAR,iB;QAAQ,sB;;UJ6qDf,eAAe,sB;UACf,IAAI,CAAC,QAAS,UAd,C;YAAyB,qBAAO,I;YAAP,uB;
;UACzB,eI/qD2B,QJ+qDZ,CAAS,QAAS,OAAIB,C;UACf,OAAO,QAAS,UAAhB,C;YACI,QIjrDuB,QJirDf,CAA
S,QAAS,OAAIB,C;YACR,WF55DG,MAAO,KE45DO,QF55DP,EE45DiB,CF55DjB,C;;UE85Dd,qBAAO,Q;;QIpr
DP,yB;O;KAXJ,C;+FACa,+B;MASmB,kBAAR,iB;MAAQ,sB;;QJorDf,eAAe,sB;QACf,IAAI,CAAC,QAAS,UAd,
C;UAAyB,qBAAO,I;UAAP,uB;;QACzB,eItrD2B,QJsrDZ,CAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAh
B,C;UACI,QIxrDuB,QJwrDf,CAAS,QAAS,OAAIB,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;Q
AGnB,qBAAO,Q;;MI7rDP,yB;K;0FAGJ,yB;MJ6rDA,sE;MI7rDA,kD;QJwsDI,eI7rDO,iBJ6rDQ,W;QACf,IAAI,C
AAC,QAAS,UAd,C;UAAyB,MAAM,6B;QAC/B,eI/rDqC,QJ+rDtB,CAAS,QAAS,OAAIB,C;QACf,OAAO,QA
S,UAAhB,C;UACI,QIjsDiC,QJisDzB,CAAS,QAAS,OAAIB,C;UACR,IIIsDqB,UJksDN,SAAQ,QAAR,EAakB,C
AAIB,CAAX,GAakC,CAAT,C;YACI,WAAW,C;;QInsDnB,OJssDO,Q;O;KIjtDX,C;sGAcA,2C;MASmB,kBAA
R,iB;MAAQ,0B;;QJssDf,eAAe,sB;QACf,IAAI,CAAC,QAAS,UAd,C;UAAyB,yBAAO,I;UAAP,2B;;QACzB,eI
sD2C,QJwsD5B,CAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QI1sDuC,QJ0sD/B,CAAS,QAAS
,OAAIB,C;UACR,II3sD2B,UJ2sDZ,SAAQ,QAAR,EAakB,CAIB,CAAX,GAakC,CAAT,C;YACI,WAAW,C;;
QAGnB,yBAAO,Q;;MI/sDP,6B;K;sFAGJ,yB;MAOA,8D;MAPA,wC;QAIL,OASe,cAAR,iBAAQ,EATM,UASN,C
;O;KAbnB,C;kGAOA,yB;MAAA,8D;MAAA,wC;QAMIL,OAAe,cAAR,iBAAQ,EAAC,UAd,C;O;KANnB,C;kFA
SA,+B;MAcmB,kBAAR,iB;MAAQ,sB;;QJwxDf,eAAe,sB;QACf,IAAI,CAAC,QAAS,UAd,C;UAAyB,qBAAO,I;
UAAP,uB;;QACzB,cAAc,QAAS,O;QACvB,IAAI,CAAC,QAAS,UAd,C;UAAyB,qBAAO,O;UAAP,uB;;QACzB
,eIvyDmB,QJuyDJ,CAAS,OAAT,C;;UAEX,QAAQ,QAAS,O;UACjB,QI1yDe,QJ0yDP,CAAS,CAAT,C;UACR,IA
AI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;QAED,QAAT,QAAS,W;QACIB,qBAAO,O;;M
IhzDP,yB;K;8FAGJ,+B;MAQmB,kBAAR,iB;MAAQ,sB;;QJwxDf,eAAe,sB;QACf,IAAI,CAAC,QAAS,UAd,C;U

AAyB,qBAAO,I;UAAP,uB;;QACzB,cAAc,QAAS,O;QACvB,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,qBAAO,O;U
AAP,uB;;QACzB,e15xD2B,QJ4xDZ,CAAS,OAAT,C;;UAEX,QAAQ,QAAS,O;UACjB,QI/xDuB,QJ+xDf,CAAS,C
AAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;QAED,QAAT,QAAS,W;QAC
IB,qBAAO,O;;MIryDP,yB;K;mFAGJ,yB;MJqyDA,sE;MF14DA,iB;MM6FA,sC;QJkzDI,eIryDO,iBJqyDQ,W;QAC
f,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eIvyDqB,QJuyDN,CAAS,QAAS,OAAIB,C;QACf,OA
AO,QAAS,UAAhB,C;UACI,QIzyDiB,QJyyDT,CAAS,QAAS,OAAIB,C;UACR,WF54DG,MAAO,KE44DO,QF54
DP,EE44DiB,CF54DjB,C;;QMkGd,OJ4yDO,Q;O;KIzzDX,C;mFAGBA,yB;MJ4yDA,sE;MFp6DA,iB;MMwHA,sC;
QJyzDI,eI5yDO,iBJ4yDQ,W;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eI9yDqB,QJ8yDN,
CAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QIhzDiB,QJgzDT,CAAS,QAAS,OAAIB,C;UACR
,WF96DG,MAAO,KE86DO,QF96DP,EE86DiB,CF96DjB,C;;QM6Hd,OJmzDO,Q;O;KIh0DX,C;mFAGBA,yB;MJ
mzDA,sE;MInzDA,sC;QJ8zDI,eInzDO,iBJmzDQ,W;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QA
C/B,eIrzDqB,QJqzDN,CAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QIvzDiB,QJuzDT,CAAS,Q
AAS,OAAIB,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;QIzzDnB,OJ4zDO,Q;O;KIv0DX,C;+FA
cA,yB;MN3IA,iB;MM2IA,sC;QAWmB,kBAAR,iB;QAAQ,sB;;UJ4zDf,eAAe,sB;UACf,IAAI,CAAC,QAAS,UAA
d,C;YAAyB,qBAAO,I;YAAP,uB;;UACzB,eI9zD2B,QJ8zDZ,CAAS,QAAS,OAAIB,C;UACf,OAAO,QAAS,UAA
hB,C;YACI,QIh0DuB,QJg0Df,CAAS,QAAS,OAAIB,C;YACR,WF/8DG,MAAO,KE+8DO,QF/8DP,EE+8DiB,CF/
8DjB,C;;UEi9Dd,qBAAO,Q;;QIn0DP,yB;O;KAXJ,C;+FAcA,yB;MNpKA,iB;MMoKA,sC;QAWmB,kBAAR,iB;Q
AAQ,sB;;UJm0Df,eAAe,sB;UACf,IAAI,CAAC,QAAS,UAAAd,C;YAAyB,qBAAO,I;YAAP,uB;;UACzB,eI9D2B,
QJq0DZ,CAAS,QAAS,OAAIB,C;UACf,OAAO,QAAS,UAAhB,C;YACI,QIv0DuB,QJu0Df,CAAS,QAAS,OAAIB,
C;YACR,WF/+DG,MAAO,KE++DO,QF/+DP,EE++DiB,CF/+DjB,C;;UEi/Dd,qBAAO,Q;;QI10DP,yB;O;KAXJ,C
;+FAcA,+B;MASmB,kBAAR,iB;MAAQ,sB;;QJ00Df,eAAe,sB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,qBA
AO,I;UAAP,uB;;QACzB,eI50D2B,QJ40DZ,CAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QI90
DuB,QJ80Df,CAAS,QAAS,OAAIB,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;QAGnB,qBAAO,
Q;;MIn1DP,yB;K;0FAGJ,yB;MJm1DA,sE;MIn1DA,kD;QJ81DI,eIn1DO,iBJm1DQ,W;QACf,IAAI,CAAC,QAAS,
UAAAd,C;UAAyB,MAAM,6B;QAC/B,eI1DqC,QJq1DtB,CAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C
;UACI,QIv1DiC,QJu1DzB,CAAS,QAAS,OAAIB,C;UACR,Iix1DqB,UJw1DN,SAAQ,QAAR,EAakB,CAAIB,CA
AX,GAakC,CAAtC,C;YACI,WAAW,C;;QIz1DnB,OJ41DO,Q;O;KIv2DX,C;sGAcA,2C;MASmB,kBAAR,iB;M
AAQ,0B;;QJ41Df,eAAe,sB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,yBAAO,I;UAAP,2B;;QACzB,eI91D2C,
QJ81D5B,CAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QIh2DuC,QJg2D/B,CAAS,QAAS,OAA
IB,C;UACR,Iij2D2B,UJi2DZ,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;QAGn
B,yBAAO,Q;;MIr2DP,6B;K;IAGJ,0C;MAGI,OASe,gBAAR,iBAAQ,EATM,UASN,C;K;kGANnB,yB;MAAA,8D;
MAAA,wC;QAMI,OAAe,cAAR,iBAAQ,EAAC,UAAAd,C;O;KANnB,C;IASA,4B;MAMI,OAAO,mB;K;iFAGX,gC;
MAOoB,Q;MADhB,IAAI,mBAAJ,C;QAae,OAAO,I;MACN,OAAA,SjBnJoE,QAAQ,W;MiBmJ5F,OAAgB,cAAh
B,C;QAAGB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,K;;MACrD,OAAO,I,K;oFAGX,6B;MAK
mC,Q;MAAA,OjB5JqD,iBAAQ,W;MiB4J7E,OAAgB,cAAhB,C;QAAGB,yB;QAAM,OAAO,OAAP,C;;MAArC,g
B;K;kGAGJ,yB;MAAA,6B;MAAA,sC;MJwyCA,wE;MIxyCA,2BAQiB,yB;QJgyCjB,wE;eIhyCiB,0B;UAAA,4B;
YAAU,kBAAR,iB;YAAQ,aAAe,c;YJuyCzB,gB;YADb,YAAY,C;YACC,6B;YAAb,OAAa,cAAb,C;cAAa,sB;cAA
M,OAAO,oBAAmB,cAAAnB,EAAMb,sBAAnB,UAAp,EAAoC,IAApC,C;;YIvyC2B,W;W;S;OAAjC,C;MARjB,oC
;QJ+yCiB,gB;QADb,YAAY,C;QACC,OIvyCE,iBJuyCF,W;QAAb,OAAa,cAAb,C;UAAa,sB;UAAM,OAAO,oBA
AmB,cAAAnB,EAAMb,sBAAnB,UAAp,EAAoC,IAApC,C;;QIvyCnB,gB;O;KARJ,C;4FAWA,qB;MAKI,OAAO,iB
;K;IAGX,iC;MAII,OAAe,aAAR,iBAAQ,C;K;IC9hBnB,kC;MAEI,gBCmE2D,8BAAY,c;MDIEvE,IAAI,SAAU,OA
AV,GAAMb,CAAvB,C;QACW,Q;QAAA,IAAI,cAAQ,GAAZ,C;UAAA,OAAsB,S;;uBAAe,qBAAU,CAAV,C;UA
AA,YAAe,SEiNc,WFjNM,CEiNN,Caff,c;UFIMnD,OG8MoD,2BAAL,GAakB,K;;QH9MxE,W;;MAEJ,OAAuB,o
BAAhB,wBAAgB,C;K;gFxBD3B,yB;MAAA,mC;MAAA,2C;MAAA,4B;QAQI,OAAO,kBAAO,cAAP,C;O;KAR
X,C;gFAWA,yB;MAAA,mC;MAAA,2C;MAAA,4B;QAQI,OAAO,kBAAO,cAAP,C;O;KARX,C;gFAWA,yB;MA
AA,mC;MAAA,2C;MAAA,4B;QAQI,OAAO,kBAAO,cAAP,C;O;KARX,C;IAWA,sC;;QAQQ,OAAc,QAAP,MA
AO,EAAQ,SAAR,C;;QACHB,+C;UACE,MAAM,2BAAuB,CAAE,QAazB,C;;UAHV,O;;K;IAOJ,sC;;QAQQ,OA
Ac,SAAP,MAAO,EAAS,SAAT,C;;QACHB,+C;UACE,MAAM,2BAAuB,CAAE,QAazB,C;;UAHV,O;;K;IAOJ,sC

:::QAQQ,OAAiD,OAA1C,MAAO,iBAAQ,e4BtCgB,I5BsCx B,EAAoB,CAAA,c4BtCI,I5BsCJ,IAAY,CAAZ,IAAp B,CAAmC,C;;QACnD,+C;UACE,MAAM,2BAAuB,CAA E,QAAzB,C;;UAHV,O;;K;4FAOJ,yB;MAAA,mC;MAA A,uD;MAAA,4B;QAOI,OAAO,wBAAa,cAAb,C;O;KAPX,C;4FAUA,yB;MAAA,mC;MAAA,uD;MAAA,4B;QAOI,OAAO,wBAAa,cAAb,C;O;KAPX,C;4FAUA,yB;MAAA,mC;MAAA,uD;MAAA,4B;QAOI,OAAO,wBAAa,cAAb,C;O;KAPX,C;IAUA,4C;MAMI,IAAI,mBAAJ,C;QACI,OAAO,I;MACX,OAAc,QAAP,MAAO,EAAQ,SAAR,C;K;IAGIB,4C;MAMI,IAAI,mBAAJ,C;QACI,OAAO,I;MACX,OAAc,SAAP,MAAO,EAAQ,SAAR,C;K;IAGIB,4C;MAMI,IAAI,mBAAJ,C;QACI,OAAO,I;MACX,OAAiD,OAA1C,MAAO,iBAAQ,e4BxGoB,I5BwG5B,EAAoB,CAAA,c4BxGQ,I5BwGR,IAAY,CAAZ,IAApB,CAAmC,C;K;mFAGrD,8B;MAQI,OAAO,mBAAmB,2BAAS,OAA T,C;K;oFAG9B,8B;MAQI,OAAO,mBAAmB,2BAAS,OAAT,C;K;oFAG9B,8B;MAQI,OAAO,mBAAmB,2BAAS,OAAT,C;K;IAG9B,uC;MAKI,OAAO,2BA Ae,KAAf,C;K;IAGX,uC;MAKI,OAAO,2BA Ae,oBAAN,KAAM,CAAF,C;K;IAGX,uC;MAKI,OAAO,2BA Ae,KAAf,C;K;IAGX,uC;MAOI,OAAO,2BA Ae,KAAf,C;K;IAGX,uC;MgBzHW,ShBgIM,mBAAN,KAAM,C;MAAb,OAA0C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAakC,K;K;IAG5E,uC;MgBnIW,ShB0IM,kBAAN,KAAM,C;MAAb,OAA2C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAakC,K;K;IAG7E,uC;MgB7IW,ShBoJM,oBAAN,KAAM,C;MAAb,OAA2C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAakC,K;K;IAG7E,uC;MgBvJW,ShB8JM,qBAAN,KAAM,C;MAAb,OAA4C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAakC,K;K;IAG9E,uC;MAKI,OAAO,2BA Ae,KAAf,C;K;IAGX,uC;MgBzKW,ShBgLM,mBAAN,KAAM,C;MAAb,OAA0C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAakC,K;K;IAG5E,uC;MgBnLW,ShB0LM,oBAAN,KAAM,C;MAAb,OAA2C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAakC,K;K;IAG7E,uC;MgB7LW,ShBoMM,oBAAN,KAAM,C;MAAb,OAA2C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAakC,K;K;IAG7E,uC;MgBvMW,ShB8MM,qBAAN,KAAM,C;MAAb,OAA4C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAakC,K;K;IAG9E,uC;MAKI,OAAO,2BA Ae,KAAf,C;K;IAGX,uC;MAKI,OAAO,2BA Ae,oBAAN,KAAM,CAAF,C;K;IAGX,uC;MgBjOW,ShBsOM,kBAAN,KAAM,C;MAAb,OAA2C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAakC,K;K;IAG7E,uC;MgBzOW,ShB8OM,mBAAN,KAAM,C;MAAb,OAA4C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAakC,K;K;IAG9E,uC;MAOI,OAAO,2BA Ae,KAAf,C;K;IAGX,uC;MAOI,OAAO,2BA Ae,KAAf,C;K;IAGX,uC;MgBrQW,ShB0QM,iBAAN,KAAM,C;MAAb,OAA0C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAakC,K;K;IAG5E,uC;MgB7QW,ShBkRM,oBAAN,KAAM,C;MAAb,OAA2C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAakC,K;K;IAG7E,uC;MgBrRW,ShB0RM,qBAAN,KAAM,C;MAAb,OAA4C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAakC,K;K;IAG9E,uC;MAOI,OAAO,2BAAS,KAAM,WAAf,C;K;IAGX,uC;MAOI,OAAO,2BAAS,KAAM,WAAf,C;K;IAGX,uC;MAKI,OAAO,2BA Ae,KAAf,C;K;IAGX,uC;MAKI,OAAO,2BA Ae,oBAAN,KAAM,CAAF,C;K;IAGX,uC;MgBjUW,ShBsUM,oBAAN,KAAM,C;MAAb,OAA2C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAakC,K;K;IAG7E,uC;MAOI,OAAO,2BA Ae,KAAf,C;K;IAGX,uC;MAOI,OAAO,2BA Ae,KAAf,C;K;IAGX,+B;MAOI,OAAO,sCAAe,yBAAGB,SAAhB,EAAyB,EAAzB,EAAkC,EAAIC,C;K;IAG1B,iC;MAOI,OAAO,uCAAGB,yBAAGB,SAAhB,EAAyB,oBAAH,EAAG,CAAzB,M;K;IAG3B,iC;MAOI,OAAO,sCAAe,yBAAqB,SAArB,EAAiC,EAAjC,EAA0C,EAA1C,C;K;IAG1B,iC;MAOI,OAAO,sCAAe,yBAAqB,SAArB,EAAiC,EAAjC,EAA0C,EAA1C,C;K;IAG1B,iC;MAOI,OAAO,uCAAGB,yBAAGB,SAAhB,EAAyB,EAAsB,EAAtB,EAA0B,EAA1B,C;K;IAG3B,iC;MAOI,OAAO,sCAAe,yBAAqB,SAArB,EAA8B,EAA9B,EAAkC,EAAIC,C;K;IAG1B,iC;MAOI,OAAO,sCAAe,yBAAqB,SAArB,EAA8B,EAA9B,EAAkC,EAAIC,C;K;IAG1B,iC;MAOI,OAAO,uCAAGB,yBAAqB,oBAAL,SAAK,CAArB,EAA+B,EAA/B,M;K;IAG3B,iC;MAOI,OAAO,uCAAGB,yBAAGB,SAAhB,EAAyB,EAAsB,EAAtB,M;K;IAG3B,kC;MAOI,OAAO,uCAAGB,yBAAqB,oBAAL,SAAK,CAArB,EAA+B,EAA/B,M;K;IAG3B,kC;MAOI,OAAO,sCAAe,yBAAGB,SAAhB,EAAyB,EAAzB,EAAkC,EAAIC,C;K;IAG1B,kC;MAOI,OAAO,uCAAGB,yBAAGB,SAAhB,EAAyB,oBAAH,EAAG,CAAzB,M;K;IAG3B,kC;MAOI,OAAO,sCAAe,yBAAGB,SAArB,EAAiC,EAAjC,EAA0C,EAA1C,C;K;IAG1B,kC;MAOI,OAAO,sCAAe,yBAAqB,SAArB,EAAiC,EAAjC,EAA0C,EAA1C,C;K;IAG1B,+B;MAII,OAAO,sCAAe,yBAAGB,cAAhB,EAAsB,EAAtB,EAA6B,CAAC,cAAD,IAA7B,C;K;IAG1B,gC;MAII,OAAO,uCAAGB,yBAAGB,cAAhB,EAAsB,EAAtB,EAA6B,CAAC,cAAD,IAA7B,C;K;IAG3B,gC;MAII,OAAO,uCAAGB,yBAAGB,cAAhB,EAAsB,EAAtB,EAA6B,CAAC,cAAD,IAA7B,C;K;IAG3B,+B;MAII,oBAAoB,OAAO,CAA3B,EAA8B,IAA9B,C;MACA,OAAO,sCAAe,yBAAGB,eAAhB,EAAuB,cAAvB,EA

AAyB,4BAAO,YAAP,KAA7B,C;UAAkD,OAAO,Y;:MAE7D,OAAO,S;K;IAGX,2D;MAQI,IAAI,eAAe,YAAAnB,
C;QAAiC,MAAM,gCAAyB,oDAAiD,YAAjD,8BAAoF,YAApF,MAAzB,C;MACvC,IAAI,YAAO,YAAX,C;QAA
yB,OAAO,Y;MACHc,IAAI,YAAO,YAAX,C;QAAyB,OAAO,Y;MACHc,OAAO,S;K;IAGX,2D;MAQI,IAAI,eAA
e,YAAAnB,C;QAAiC,MAAM,gCAAyB,oDAAiD,YAAjD,8BAAoF,YAApF,MAAzB,C;MACvC,IAAI,YAAO,YA
AX,C;QAAyB,OAAO,Y;MACHc,IAAI,YAAO,YAAX,C;QAAyB,OAAO,Y;MACHc,OAAO,S;K;IAGX,2D;MAQ
I,IAAI,eAAe,YAAAnB,C;QAAiC,MAAM,gCAAyB,oDAAiD,YAAjD,8BAAoF,YAApF,MAAzB,C;MACvC,IAAI,
YAAO,YAAX,C;QAAyB,OAAO,Y;MACHc,IAAI,YAAO,YAAX,C;QAAyB,OAAO,Y;MACHc,OAAO,S;K;IAG
X,2D;MAQI,IAAI,6BA Ae,YAAf,KAAJ,C;QAAiC,MAAM,gCAAyB,oDAAiD,YAAjD,yCAAO,F,YAApF,iBA AzB
,C;MACvC,IAAI,0BAAO,YAAP,KAAJ,C;QAAyB,OAAO,Y;MACHc,IAAI,0BAAO,YAAP,KAAJ,C;QAAyB,OA
AO,Y;MACHc,OAAO,S;K;IAGX,2D;MAQI,IAAI,eAAe,YAAAnB,C;QAAiC,MAAM,gCAAyB,oDAAiD,YAAjD,8
BAAoF,YAApF,MAAzB,C;MACvC,IAAI,YAAO,YAAX,C;QAAyB,OAAO,Y;MACHc,IAAI,YAAO,YAAX,C;Q
AAyB,OAAO,Y;MACHc,OAAO,S;K;IAGX,2D;MAQI,IAAI,eAAe,YAAAnB,C;QAAiC,MAAM,gCAAyB,oDAAi
D,YAAjD,8BAAoF,YAApF,MAAzB,C;MACvC,IAAI,YAAO,YAAX,C;QAAyB,OAAO,Y;MACHc,IAAI,YAAO,
YAAX,C;QAAyB,OAAO,Y;MACHc,OAAO,S;K;IAGX,sC;MAUW,Q;MADP,IAAI,KAAM,UAAV,C;QAAqB,M
AAM,gCAAyB,4CAAYC,KAAzC,MAAzB,C;MAGvB,IAAA,KAAM,0BAAiB,SAAjB,EAAuB,KAAM,MAA7B,
CAAN,IAA6C,CAAC,KAAM,0BAAiB,KAAM,MAAvB,EAA8B,SAA9B,CAApD,C;QAAiG,OAAN,KAAM,M;
WAEjG,IAAA,KAAM,0BAAiB,KAAM,aAAvB,EAAqC,SAArC,CAAN,IAAoD,CAAC,KAAM,0BAAiB,SAAjB,
EAAuB,KAAM,aAA7B,CAA3D,C;QAA+G,OAAN,KAAM,a;:QACvG,gB;MALZ,W;K;IASJ,sC;MAYW,Q;MAJP
,IAAI,8CAAJ,C;QACI,OAAY,WAAL,SAAK,EAAY,KAAZ,C;:MAEhB,IAAI,KAAM,UAAV,C;QAAqB,MAAM,
gCAAyB,4CAAYC,KAAzC,MAAzB,C;MAEvB,gCAAO,KAAM,MAAb,M;QAA4B,OAAN,KAAM,M;WAC5B,g
CAAO,KAAM,aAAb,M;QAAMC,OAAN,KAAM,a;:QAC3B,gB;MAHZ,W;K;IAOJ,sC;MAYW,Q;MAJP,IAAI,8C
AAJ,C;QACI,OAAY,WAAL,SAAK,EAAC,KAAZ,C;:MAEhB,IAAI,KAAM,UAAV,C;QAAqB,MAAM,gCAAyB,
4CAAYC,KAAzC,MAAzB,C;MAEvB,gBAAO,KAAM,MAAb,C;QAA4B,OAAN,KAAM,M;WAC5B,gBAAO,KA
AM,aAAb,C;QAAMC,OAAN,KAAM,a;:QAC3B,gB;MAHZ,W;K;IAOJ,sC;MAYW,Q;MAJP,IAAI,8CAAJ,C;QA
CI,OAAY,WAAL,SAAK,EA Ae,KAAf,C;:MAEhB,IAAI,KAAM,UAAV,C;QAAqB,MAAM,gCAAyB,4CAAYC,K
AAzC,MAAzB,C;MAEvB,8BAAO,KAAM,MAAb,M;QAA4B,OAAN,KAAM,M;WAC5B,8BAAO,KAAM,aAAb,
M;QAAMC,OAAN,KAAM,a;:QAC3B,gB;MAHZ,W;K;IW1rCJ,oD;MAMuF,wC;K;IANvF,8CAOI,Y;MAAuC,8B;
K;IAP3C,gF;IkbQA,yC;MAMI,OAAO,sBAAQ,OAAR,KAAoB,C;K;IAWG,2C;MAAA,qB;QAAE,MAAM,8BAA
0B,+CAA4C,aAA5C,MAA1B,C;O;K;IAR1C,uC;MAQI,OAAO,8BAAgB,KAAhB,EAAuB,yBAAvB,C;K;IAGX,4
D;MAcqB,Q;MANjB,IAAI,QAAQ,CAAZ,C;QACI,OAAO,aAAa,KAAb,C;MACX,eAAe,oB;MACf,YAAy,C;MA
CZ,OAAO,QAAS,UAAhB,C;QACI,cAAc,QAAS,O;QACvB,IAAI,WAAS,YAAT,EAAS,oBAAT,OAAJ,C;UACI,
OAAO,O;:MAEf,OAAO,aAAa,KAAb,C;K;IAGX,8C;MAcqB,Q;MANjB,IAAI,QAAQ,CAAZ,C;QACI,OAAO,I;M
ACX,eAAe,oB;MACf,YAAy,C;MACZ,OAAO,QAAS,UAAhB,C;QACI,cAAc,QAAS,O;QACvB,IAAI,WAAS,Y
AAT,EAAS,oBAAT,OAAJ,C;UACI,OAAO,O;:MAEf,OAAO,I;K;8EAGX,gC;MASW,sB;:QA2FS,Q;QAAA,2B;Q
AAhB,OAAGB,cAAhB,C;UAAgB,yB;UAAM,IA3FH,SA2FO,CAAU,OA AV,CAAJ,C;YAAwB,qBAAO,O;YAAP,
uB;:QAC9C,qBAAO,I;:MA5FP,yB;K;uFAGJ,gC;MAkOoB,Q;MADhB,WAAe,I;MACC,2B;MAAhB,OAAGB,cA
AhB,C;QAAgB,yB;QACZ,IA1Nc,SA0NV,CAAU,OA AV,CAAJ,C;UACI,OAAO,O;:MA3Nf,OA8NO,I;K;IA3NX,
6B;MAOI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAA d,C;QACI,MAAM,2BAAuB,oBAAvB,C;MACV,OAAO,QA
AS,O;K;IFAGpB,yB;MAAA,iE;MAAA,uC;QAOoB,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UAAM
,IAAI,UAAU,OA AV,CAAJ,C;YAAwB,OAAO,O;:QACrD,MAAM,gCAAuB,sDAAvB,C;O;KARV,C;kGAWA,yB
;MAAA,iE;MAAA,uC;QAWW,Q;QAAA,+B;:UAcS,U;UAAA,6B;UAAhB,OAAGB,gBAAhB,C;YAAgB,2B;YAC
Z,aAfwB,SAeX,CAAU,OA AV,C;YAcB,IAAI,cAAJ,C;cACI,8BAAO,M;cAAP,gC;:UAGR,8BAAO,I;:QApBA,k
C;QAAA,iB;UAAMC,MAAM,gCAAuB,iEAAvB,C;:QAAhD,OAAO,I;O;KAXX,C;8GAcA,gC;MAWoB,Q;MAA
A,2B;MAAhB,OAAGB,cAAhB,C;QAAgB,yB;QACZ,aAAa,UAAU,OA AV,C;QAcB,IAAI,cAAJ,C;UACI,OAAO,
M;:MAGf,OAAO,I;K;IAGX,mC;MAMI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAA d,C;QACI,OAAO,I;MACX,O
AAO,QAAS,O;K;6FAGpB,gC;MAMoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAgB,yB;QAAM,IAAI,UA
AU,OA AV,CAAJ,C;YAAwB,OAAO,O;:MACrD,OAAO,I;K;IAGX,wC;MAOiB,Q;MADb,YAAy,C;MACC,2B;M
AAb,OAAa,cAAb,C;QAAa,sB;QACT,mBAAmB,KAAAnB,C;QACA,IAAI,gBAAW,IAAX,CAAJ,C;UACI,OAAO,

K;QACX,qB;;MAEJ,OAAO,E;K;+FAGX,yB;MAAA,wE;MAAA,uC;QAOiB,Q;QADb,YAAY,C;QACC,2B;QAA b,OAAa,cAAb,C;UAAa,sB;UACT,mBAAmB,KAAAnB,C;UACA,IAAI,UAAU,IAAV,CAAJ,C;YACI,OAAO,K;UA CX,qB;;QAEJ,OAAO,E;O;KAbX,C;6FAGBA,yB;MAAA,wE;MAAA,uC;QAQiB,Q;QAFb,gBAAgB,E;QACbB,Y AAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,mBAAmB,KAAAnB,C;UACA,IAAI,UAAU,IAAV,CA AJ,C;YACI,YAAY,K;UACbB,qB;;QAEJ,OAAO,S;O;KAdX,C;IAiBA,4B;MAUI,eAAe,oB;MACf,IAAI,CAAC,Q AAS,UAAAd,C;QACI,MAAM,2BAAuB,oBAAvB,C;MACV,WAAW,QAAS,O;MACpB,OAAO,QAAS,UAAhB,C; QACI,OAAO,QAAS,O;MACpB,OAAO,I;K;+EAGX,yB;MAAA,iE;MAAA,gB;MAAA,8B;MAAA,uC;QAYoB,U AQT,M;QAVP,WAAe,I;QACf,YAAY,K;QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAAI,UAAU,O AAV,CAAJ,C;YACI,OAAO,O;YACP,QAAQ,I;;;QAGhB,IAAI,CAAC,KAAL,C;UAAy,MAAM,gCAAuB,sDAA vB,C;QAEIB,OAAO,2E;O;KApBX,C;IAuBA,4C;MAQiB,Q;MAFb,gBAAgB,E;MACHb,YAAY,C;MACC,2B;MA Ab,OAAa,cAAb,C;QAAa,sB;QACT,mBAAmB,KAAAnB,C;QACA,IAAI,gBAAW,IAAX,CAAJ,C;UACI,YAAY,K; QACbB,qB;;MAEJ,OAAO,S;K;IAGX,kC;MAQI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QACI,OAAO,I;M ACX,WAAW,QAAS,O;MACpB,OAAO,QAAS,UAAhB,C;QACI,OAAO,QAAS,O;MACpB,OAAO,I;K;2FAGX,g C;MASoB,Q;MADhB,WAAe,I;MACC,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI,UAAU,OAAV,C AAJ,C;UACI,OAAO,O;;;MAGf,OAAO,I;K;IAGX,8B;MAMI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAC I,MAAM,2BAAuB,oBAAvB,C;MACV,aAAa,QAAS,O;MACtB,IAAI,QAAS,UAAb,C;QACI,MAAM,gCAAyB,q CAAzB,C;MACV,OAAO,M;K;mFAGX,yB;MAAA,kF;MAAA,iE;MAAA,gB;MAAA,8B;MAAA,uC;QAQoB,UA ST,M;QAXP,aAAiB,I;QACjB,YAAY,K;QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAAI,UAAU,O AAV,CAAJ,C;YACI,IAAI,KAJ,C;cAAW,MAAM,8BAAYB,mDAAzB,C;YACjB,SAAS,O;YACT,QAAQ,I;;;QA GhB,IAAI,CAAC,KAAL,C;UAAy,MAAM,gCAAuB,sDAAvB,C;QAEIB,OAAO,6E;O;KAjBX,C;IAoBA,oC;MA MI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QACI,OAAO,I;MACX,aAAa,QAAS,O;MACtB,IAAI,QAAS,U AAb,C;QACI,OAAO,I;MACX,OAAO,M;K;+FAGX,gC;MAQoB,Q;MAFhB,aAAiB,I;MACjB,YAAY,K;MACI,2B ;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI,UAAU,OAAV,CAAJ,C;UACI,IAAI,KAJ,C;YAAY,OAA O,I;UACIB,SAAS,O;UACT,QAAQ,I;;;MAGhB,IAAI,CAAC,KAAL,C;QAAY,OAAO,I;MACnB,OAAO,M;K;IAG X,8B;MAWW,Q;MhBhXP,IAAI,EgB+WI,KAAC,ChB/WT,CAAJ,C;QACI,cgB8Wc,sD;QhB7Wd,MAAM,gCAAy B,OAAQ,WAAjC,C;;MgB+WN,UAAK,CAAL,C;QAAU,gB;WACV,+C;QAAiC,OAAAL,SAAK,cAAK,CAAL,C;; QACzB,wBAAa,SAAb,EAAmB,CAAnB,C;MAHZ,W;K;IAOJ,2C;MAQI,OAAO,sBAAkB,SAAB,EAAwB,SAAX B,C;K;IAGX,wC;MAQI,OAAO,sBAAkB,SAAB,EAAwB,IAAxB,EAA8B,SAA9B,C;K;IacqE,iD;MAAA,qB;QA AE,yBAAU,EAAG,MAAb,EAAoB,EAAG,MAAvB,C;O;K;IAAkC,oC;MAAE,OAAA,EAAG,M;K;IAXzH,+C;M AWI,OAAO,yBAAqB,sBAAkB,qBAAiB,SAAJB,CAAIB,EAA0C,IAA1C,EAAgD,+BAAhD,CAArB,EAAyG,sBA AzG,C;K;oGAGX,yB;MA80BA,wE;MA90BA,oD;QAU1BiB,gB;QADb,YAAY,C;QACC,2B;QAAb,OAAa,cAAb, C;UAAa,sB;UA50BT,IAAI,UA40BkB,oBAAmB,cAAAnB,EAAmB,sBAAnB,UA50BIB,EA40B+C,IA50B/C,CAAJ, C;YAA2C,sBA40BQ,IA50BR,C;;QAE/C,OAAO,W;O;KAbX,C;sGAGBA,yB;MAAA,8C;MAAA,0C;MAAA,8B; MASKb,qD;QAAA,qB;UAAE,c;S;O;MATpB,sC;QASW,Q;QAAP,OAAO,uCAAo,iCAAP,gC;O;KATX,C;0GAY A,4C;MAQoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,YAAJ,C;UAAkB,WAAy,WA AI,OAAJ,C;;MACpD,OAAO,W;K;IAGX,2C;MAQI,OAAO,sBAAkB,SAAB,EAAwB,KAAxB,EAA+B,SAA/B,C; K;IAYU,kC;MAAE,iB;K;IATvB,oC;MASW,Q;MAAP,OAAO,4CAAU,oBAAV,kC;K;IAGX,mD;MAQoB,Q;MA AA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,eAAJ,C;UAAqB,WAAy,WAAI,OAAJ,C;;MACvD, OAAO,W;K;4FAGX,6C;MAQoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,CAAC,UA AU,OAAV,CAAL,C;UAAyB,WAAy,WAAI,OAAJ,C;;MAC3D,OAAO,W;K;sFAGX,6C;MAQoB,Q;MAAA,2B; MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,WAAy,WAAI,OAAJ,C;; MAC1D,OAAO,W;K;IAGX,8B;MAWW,Q;MhBzgpB,IAAI,EgBwgBI,KAAC,ChBxgBT,CAAJ,C;QACI,cgBugBc ,sD;QhBtgBd,MAAM,gCAAyB,OAAQ,WAAjC,C;;MgBwgBN,UAAK,CAAL,C;QAAU,sB;WACV,+C;QAAiC,O AAL,SAAK,cAAK,CAAL,C;;QACzB,wBAAa,SAAb,EAAmB,CAAnB,C;MAHZ,W;K;IAOJ,2C;MAQI,OAAO,sB AAKB,SAAB,EAAwB,SAAXB,C;K;IAWA,2C;MAAA,8B;K;8CACH,Y;MACI,iBAA6B,iBAAZ,gBAAy,C;MAC IB,QAAAX,UAAW,C;MACX,OAAO,UAAW,W;K;;IAZ9B,6B;MAQI,0C;K;sFASJ,yB;MAAA,sD;MdfjA,sC;MAA A,oC;MAAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA5Dd,cAAc,SA4DgB,CA 5DhB,CAAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;Mc0ef,sC;QAUI,OAAO,sBdpfP,eAAW,iBcofiB,

QdpfjB,CAAX,CcofO,C;O;KAVX,C;0GAaA,yB;MAAA,sD;Md3eA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D; eAwFe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN, CAA3B,C;W;S;OA+EI,C;Mcoef,sC;QAQI,OAAO,sBd5eP,eAAW,2Bc4e2B,Qd5e3B,CAAX,Cc4eO,C;O;KARX,C; IAWA,uC;MAQI,OAAO,wBAAW,cAAX,C;K;IAWA,uE;MAAA,sC;MAAA,4C;K;kDACH,Y;MACI,iBAAiC,iBA AhB,oBAAgB,C;MACtB,WAAx,UAAW,EAAS,uBAAT,C;MACX,OAAO,UAAW,W;K;;IAZ9B,6C;MAQI,0D;K; wFASJ,yB;MAAA,wE;MAAA,uC;QAaW,kBAAY,oB;QAI FH,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,y B;UACZ,WAlFsC,SAkFvB,CAAU,OAAV,C;UvBnEnB,wBAAI,IAAK,MAAT,EAAGB,IAAK,OAArB,C;;QuBfA, OAoFO,W;O;KAjGX,C;6FAGBA,yB;MAAA,wE;MAAA,yC;QAaW,kBAAc,oB;QA8BL,Q;QAAA,2B;QAAhB,O AAgB,cAAhB,C;UAAgB,yB;UACZ,WAAy,aA/B4B,WA+BxB,CAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;QA/Bh B,OAIcO,W;O;KA9CX,C;6FAGBA,yB;MAAA,wE;MAAA,yD;QAYW,kBAAc,oB;QAIcL,Q;QAAA,2B;QAAhB, OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAAy,aAlC4B,WakCxB,CAAY,OAAZ,CAAJ,EAlCyC,cAkCf,CAAE,OA Af,CAA1B,C;;QAIChB,OAoCO,W;O;KAhDX,C;iGAeA,+C;MAYoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;Q AAgB,yB;QACZ,WAAy,aAAI,YAAy,OAAZ,CAAJ,EAA0B,OAA1B,C;;MAEhB,OAAO,W;K;iGAGX,+D;MAY oB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,WAAy,aAAI,YAAy,OAAZ,CAAJ,EAA0B,eA Ae,OAaf,CAA1B,C;;MAEhB,OAAO,W;K;4FAGX,6C;MAWoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAA gB,yB;QACZ,WAAe,UAAU,OAAV,C;QvBnEnB,wBAAI,IAAK,MAAT,EAAGB,IAAK,OAArB,C;;MuBqEA,OA AO,W;K;gGAGX,yB;MAAA,wE;MAAA,2C;QAcI,aAAa,oB;QAgBG,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;U AAgB,yB;UafO,MAgBP,aAAI,OAAJ,EAhBe,aAgBF,CAAc,OAAd,CAAb,C;;QAhhB,OAAuB,M;O;Kaf3B,C;o GakBA,iD;MAYoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,WAAy,aAAI,OAAJ,EAAa,cA Ac,OAAd,CAAb,C;;MAEhB,OAAO,W;K;IAGX,gD;MAMiB,Q;MAAA,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QA CT,WAAy,WAAI,IAAJ,C;;MAEhB,OAAO,W;K;IAGX,gC;MAMI,OAAO,0BAAa,cAAb,C;K;IAGX,8B;MAMI,O AA4B,qBAAhB,iBAAL,SAAK,CAAgB,C;K;IAGhC,qC;MAMI,OAAO,0BAAa,gBAAb,C;K;IAGX,4B;MAQI,OA AwC,oBAAjC,0BAAa,sBAAb,CAAI,C;K;IAG5C,0C;MAYI,OAAO,uBAAmB,SAAnB,EAAyB,SAAzB,6BAAo C,qB;;OAApC,E;K;IAGX,0C;MAQI,OAAO,uBAAmB,SAAnB,EAAyB,SAAzB,6BAAoC,qB;;OAApC,E;K;IAGX ,iD;MAaI,OAAO,kBA Ae,SAaf,EAAqB,SAArB,6BAAgC,qB;;OAAhC,E;K;IAGX,iD;MAaI,OAAO,kBA Ae,SAaf ,EAAqB,SAArB,6BAAgC,qB;;OAAhC,E;K;sGAGX,yB;MAAA,wE;MAAA,gD;MAAA,oD;QAaoB,UAC4B,M;Q AF5C,YAAy,C;QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAAW,UAAU,oBAAmB,cAAnB,EAA mB,sBAAnB,UAAV,EAAuC,OAAvC,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KAjBX,C;u GAoBA,yB;MAAA,wE;MAAA,gD;MAAA,oD;QAaoB,UAC4B,M;QAF5C,YAAy,C;QACI,2B;QAAhB,OAAgB, cAAhB,C;UAAgB,yB;UACZ,WAAW,UAAU,oBAAmB,cAAnB,EAAmB,sBAAnB,UAAV,EAAuC,OAAvC,C;U ACC,OAAZ,WAAy,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KAjBX,C;yFAoBA,yB;MAAA,gD;MAAA,oD;QAuO B,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAy,E AAO,IAAP,C;;QAEhB,OAAO,W;O;KAdX,C;yFAiBA,yB;MAAA,gD;MAAA,oD;QAMoB,Q;QAAA,2B;QAAhB, OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAy,EAAO,IAAP,C;;QAEhB, OAAO,W;O;KAVX,C;qFAaA,yB;MAAA,wE;MA6BA,+D;MA7BA,yC;QAWW,kBAAU,oB;QA6BD,Q;QAAA,2 B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UA9BiD,WA8BvC,CAAY,OAAZ,C;UvBjoBP,U;UADP,YuBm oBe,WvBnoBH,WuBmoBwB,GvBnoBxB,C;UACL,IAAI,aAAJ,C;YACH,auBioBuC,gB;YAA5B,WvBhoBX,auBg oBgC,GvBhoBhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UuB6nBA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAhCT,OAK CO,W;O;KA7CX,C;qFAcA,yB;MAAA,wE;MAkCA,+D;MAICA,yD;QAYW,kBAAU,oB;QAKCD,Q;QAAA,2B;Q AAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UAnCiD,WAmCvC,CAAY,OAAZ,C;UvBrpBP,U;UADP,YuBupBe, WvBvpBH,WuBupBwB,GvBvpBxB,C;UACL,IAAI,aAAJ,C;YACH,auBqpBuC,gB;YAA5B,WvBppBX,auBopBg C,GvBppBhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UuBipBA,iB;UACA,IAAK,WArCyD,cAqCrD,CAAE,OAaf,C AAJ,C;;QArCT,OAuCO,W;O;KAnDX,C;yFAeA,yB;MAAA,+D;MAAA,sD;QAWoB,Q;QAAA,2B;QAAhB,OAA gB,cAAhB,C;UAAgB,yB;UACZ,UAAU,YAAy,OAAZ,C;UvBjoBP,U;UADP,YuBmoBe,WvBnoBH,WuBmoBwB ,GvBnoBxB,C;UACL,IAAI,aAAJ,C;YACH,auBioBuC,gB;YAA5B,WvBhoBX,auBgoBgC,GvBhoBhC,EAAS,MA AT,C;YACA,e;;YAEA,c;;UuB6nBA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KAhBX,C;yFAmBA,y B;MAAA,+D;MAAA,sE;QAYoB,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UAAU,YAAy,OA AZ,C;UvBrpBP,U;UADP,YuBupBe,WvBvpBH,WuBupBwB,GvBvpBxB,C;UACL,IAAI,aAAJ,C;YACH,auBqpBu

C,gB;YAA5B,WvBppBX,auBopBgC,GvBppBhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UuBipBA,iB;UACA,IAAK ,WAAI,eAAe,OAAf,CAAJ,C;;QAET,OAAO,W;O;KAjBX,C;OFAoBA,yB;MAAA,kC;MAAA,4C;MAAA,wE;QA UW,sC;QAAA,8C;O;MAVX,oDAWQ,Y;QAA6C,OAAA,oBAAgB,W;O;MAXrE,iDAYQ,mB;QAAoC,gCAAY,O AAZ,C;O;MAZ5C,gF;MAAA,yC;QAUI,2D;O;KAVJ,C;IAGBA,sC;MASI,OAAO,yBAAqB,SAArB,EAA2B,SAA3 B,C;K;IAGX,4C;MASI,OAAO,gCAA4B,SAA5B,EAakC,SAaIC,C;K;IAGX,mD;MASI,OAAoD,gBAA7C,gCAA 4B,SAA5B,EAakC,SAaIC,CAA6C,C;K;4GAGxD,yB;MAuNA,wE;MAvNA,oD;QAgOiB,gB;QADb,YAAY,C;Q ACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UAvNsB,U;UAAA,wBAuNT,oBAAmB,cAAAnB,EAAMb,sBAAnB,UA vNS,EAuNoB,IAvNpB,W;YAA6C,6B;;;QACHf,OAAO,W;O;KAVX,C;8FAaA,yB;MAAA,wE;MAAA,oD;QAUI B,UACoC,M;QAFjD,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,WAAy,WAAI,UAAU,oBAA mB,cAAAnB,EAAMb,sBAAnB,UAAV,EAAC,IAAvC,CAAJ,C;;QACHB,OAAO,W;O;KAZX,C;IAeA,4C;MASI,O AA6C,gBAAtC,yBAAqB,SAArB,EAA2B,SAA3B,CAAsC,C;K;8FAGjD,yB;MAAA,oD;QA4KoB,Q;QAAA,2B;Q AAhB,OAAgB,cAAhB,C;UAAgB,yB;UArKK,U;UAAA,wBAqKQ,OArKR,W;YAA5C,6B;;;QAC3D,OAAO,W;O; KARX,C;iFAWA,6C;MAOiB,Q;MAAA,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,WAAy,WAAI,UAAU,IAA V,CAAJ,C;;MACHB,OAAO,W;K;IAGX,gC;MAOI,OAAO,qBAAiB,SAAjB,C;K;IAGB,6B;MAAE,S;K;IAX7B,+ B;MAWI,OAAY,aAAL,SAAK,EAAW,eAAX,C;K;IAGhB,2C;MAYI,OAAO,qBAAiB,SAAjB,EAuB,QAAvB,C; K;IAGX,mC;MASiB,Q;MADb,UAAU,sB;MACG,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QAAM,GAAL,WAAI,IAA J,C;;MACvB,OAAO,G;K;6EAGX,gC;MAQoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IA AI,CAAC,UAAU,OAAV,CAAL,C;UAAyB,OAAO,K;;MACtD,OAAO,I;K;IAGX,2B;MAQI,OAAO,oBAAW,U;K ;6EAGtB,gC;MAQoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ, C;UAAwB,OAAO,I;;MACrD,OAAO,K;K;IAGX,6B;MAOoB,Q;MADhB,YAAY,C;MACI,2B;MAAhB,OAAgB,c AAhB,C;QAAgB,yB;QAAM,oBAAmB,qBAAnB,EAAMb,KAAAnB,E;;MACtB,OAAO,K;K;iFAGX,yB;MAAA,w E;MAAA,uC;QAOoB,Q;QADhB,YAAY,C;QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,UAA U,OAAV,CAAJ,C;YAAwB,oBAAmB,qBAAnB,EAAMb,KAAAnB,E;;QAC9C,OAAO,K;O;KARX,C;8EAWA,yC; MAYoB,Q;MADhB,kBAAkB,O;MACF,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,cAAc,UAAU,WAA V,EAuB,OAAvB,C;;MACpC,OAAO,W;K;4FAGX,yB;MAAA,wE;MAAA,gD;QAcOB,UAAiD,M;QAFjE,YAA Y,C;QACZ,kBAAkB,O;QACF,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,cAAc,UAAU,oBAAmB,cAAAn B,EAAMb,sBAAnB,UAAV,EAAC,WAAvC,EAAD,OAApD,C;;QACpC,OAAO,W;O;KAFX,C;qFAkBA,6B;M AMoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,OAAO,OAAP,C;;K;kGAG1B,yB;MAAA,w E;MAAA,oC;QASiB,UAAgC,M;QAD7C,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UAAM,OAAO,o BAAmB,cAAAnB,EAAMb,sBAAnB,UAAP,EAAC,IAAP,C,C;;O;KATvB,C;IAYA,2B;MAII,OAAO,uB;K;IAGX,2 B;MAII,OAAO,uB;K;IAGX,2B;MAGI,OAAO,uB;K;iFAGX,+B;MAGW,sB;;QAYP,eAAe,oB;QACf,IAAI,CAAC, QAAS,UAAAd,C;UAAyB,qBAAO,I;UAAp,uB;;QACzB,cAAc,QAAS,O;QACvB,IAAI,CAAC,QAAS,UAAAd,C;UA AyB,qBAAO,O;UAAp,uB;;QACzB,eAhBmB,QAgBJ,CAAS,OAAT,C;;UAEX,QAAQ,QAAS,O;UACjB,QAnBe, QAmBP,CAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;QAED,QAA T,QAAS,W;QACIB,qBAAO,O;;MAzBP,yB;K;6FAGJ,+B;MASI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;Q AAyB,OAAO,I;MACHC,cAAc,QAAS,O;MACvB,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,O;MACHC,eAAe, SAAS,OAAT,C;;QAEX,QAAQ,QAAS,O;QACjB,QAAQ,SAAS,CAAT,C;QACR,IAAI,2BAAW,CAAX,KAAJ,C; UACI,UAAU,C;UACV,WAAW,C;;MAED,QAAT,QAAS,W;MACIB,OAAO,O;K;iFAGX,yB;MAAA,sE;MZpwC A,iB;MYowCA,sC;QAEI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eAAe,SAAS, QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,WZhxCG,MAA O,KYgxCO,QZhxCP,EYgxCiB,CZhxCjB,C;;QYkxCd,OAAO,Q;O;KATBX,C;iFAyBA,yB;MAAA,sE;MZxyCA,iB ;MYwyCA,sC;QAEI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eAAe,SAAS,QA AS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,WZpzCG,MAAO,K YozCO,QZpzCP,EYozCiB,CZpzCjB,C;;QYszCd,OAAO,Q;O;KATBX,C;iFAyBA,yB;MAAA,sE;MAAA,sC;QAAI, eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eAAe,SAAS,QAAS,OAAIB,C;QACf, OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,W AAW,C;;QAGnB,OAAO,Q;O;KATBX,C;6FAyBA,yB;MZ/0CA,iB;MY+0CA,sC;QAAI,eAAe,oB;QACf,IAAI,CA AC,QAAS,UAAAd,C;UAAyB,OAAO,I;QACHC,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UA

CI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,WZz1CG,MAAO,KYy1CO,QZz1CP,EYy1CiB,CZz1CjB,C;;QY21Cd,
OAAO,Q;O;KApBX,C;6FAuBA,yB;MZj3CA,iB;MYi3CA,sC;QAaI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C
;UAAyB,OAAO,I;QACf,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,Q
AAS,OAAIB,C;UACR,WZ33CG,MAAO,KY23CO,QZ33CP,EY23CiB,CZ33CjB,C;;QY63Cd,OAAO,Q;O;KApB
X,C;6FAuBA,+B;MAWI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACf,eAAe,SAAS,Q
AAS,OAAIB,C;MACf,OAAO,QAAS,UAAhB,C;QACI,QAAQ,SAAS,QAAS,OAAIB,C;QACR,IAAI,2BAAW,CA
AX,KAAJ,C;UACI,WAAW,C;;;MAGnB,OAAO,Q;K;yFAGX,yB;MAAA,sE;MAAA,kD;QAaI,eAAe,oB;QACf,IA
AI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAA
hB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAAkB,CAAIB,CAAX,GAakC,
CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAtBX,C;qGAYBA,2C;MAWI,eAAe,oB;MACf,IAAI,CAAC,Q
AAS,UAAAd,C;QAAyB,OAAO,I;MACf,eAAe,SAAS,QAAS,OAAIB,C;MACf,OAAO,QAAS,UAAhB,C;QACI,Q
AAQ,SAAS,QAAS,OAAIB,C;QACR,IAAI,UAAW,SAAQ,QAAR,EAAkB,CAAIB,CAAX,GAakC,CAAtC,C;UA
CI,WAAW,C;;;MAGnB,OAAO,Q;K;IAGX,iC;MASI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OA
AO,I;MACf,eAAe,SAAS,QAAS,OAAIB,C;MACf,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,MZ18CG,MA
AO,KY08CE,GZ18CF,EY08CO,CZ18CP,C;;MY48Cd,OAAO,G;K;IAGX,iC;MASI,eAAe,oB;MACf,IAAI,CAAC,
QAAS,UAAAd,C;QAAyB,OAAO,I;MACf,eAAe,SAAS,QAAS,OAAIB,C;MACf,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QA
AS,O;QACjB,MZx+CG,MAAO,KYw+CE,GZx+CF,EYw+CO,CZx+CP,C;;MY0+Cd,OAAO,G;K;IAGX,iC;MAOI
,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACf,eAAe,SAAS,QAAS,OAAIB,C;MACf,OAAO,QA
AS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,sBAAM,CAAN,KAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;
K;IAGX,2C;MAGI,OAAO,4BAAc,UAAAd,C;K;IAGX,iD;MAOI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;Q
AAyB,OAAO,I;MACf,eAAe,SAAS,QAAS,OAAIB,C;MACf,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAA
I,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,
2B;MAII,OAAO,uB;K;IAGX,2B;MAII,OAAO,uB;K;IAGX,2B;MAGI,OAAO,uB;K;iFAGX,+B;MAGW,sB;;QAY
P,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,qBAAO,I;UAAp,uB;;QACzB,cAAc,QAAS,O;QACvB,IA
AAI,CAAC,QAAS,UAAAd,C;UAAyB,qBAAO,O;UAAp,uB;;QACzB,eAhBmB,QAgBJ,CAAS,OAAT,C;;UAEX,Q
AAQ,QAAS,O;UACjB,QAnBe,QAmBP,CAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;
YACV,WAAW,C;;;QAED,QAAT,QAAS,W;QACIB,qBAAO,O;;MAzBP,yB;K;6FAGJ,+B;MASI,eAAe,oB;MAC
f,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACf,cAAc,QAAS,O;MACvB,IAAI,CAAC,QAAS,UAAAd,C;
QAAyB,OAAO,O;MACf,eAAe,SAAS,OAAT,C;;QAEX,QAAQ,QAAS,O;QACjB,QAAQ,SAAS,CAAT,C;QAC
R,IAAI,2BAAW,CAAX,KAAJ,C;UACI,UAAU,C;UACV,WAAW,C;;;MAED,QAAT,QAAS,W;MACIB,OAAO,O
;K;iFAGX,yB;MAAA,sE;MZj3CA,iB;MYi3CA,sC;QAeI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,
MAAM,6B;QAC/B,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,O
AAIB,C;UACR,WZ73CG,MAAO,KY63CO,QZ73CP,EY63CiB,CZ73CjB,C;;QY+3Cd,OAAO,Q;O;KAtBX,C;iFA
yBA,yB;MAAA,sE;MZr5CA,iB;MYq5CA,sC;QAeI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAA
M,6B;QAC/B,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB
,C;UACR,WZj6CG,MAAO,KYi6CO,QZj6CP,EYi6CiB,CZj6CjB,C;;QYm6Cd,OAAO,Q;O;KAtBX,C;iFAyBA,yB;
MAAA,sE;MAAA,sC;QAaI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eAAe,SA
AS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,IAAI,2BAA
W,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAtBX,C;6FAyBA,yB;MZ57CA,iB;MY47CA,sC;QA
aI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,I;QACf,eAAe,SAAS,QAAS,OAAIB,C;QACf,
OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,WZt8CG,MAAO,KYs8CO,QZt8CP,EYs8
CiB,CZt8CjB,C;;QYw8Cd,OAAO,Q;O;KApBX,C;6FAuBA,yB;MZ99CA,iB;MY89CA,sC;QAaI,eAAe,oB;QACf,IA
AAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,I;QACf,eAAe,SAAS,QAAS,OAAIB,C;QACf,OAAO,QAAS,UAA
hB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,WZx+CG,MAAO,KYw+CO,QZx+CP,EYw+CiB,CZx+CjB,C;
;QY0+Cd,OAAO,Q;O;KApBX,C;6FAuBA,+B;MAWI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OA
AO,I;MACf,eAAe,SAAS,QAAS,OAAIB,C;MACf,OAAO,QAAS,UAAhB,C;QACI,QAAQ,SAAS,QAAS,OAAI
B,C;QACR,IAAI,2BAAW,CAAX,KAAJ,C;UACI,WAAW,C;;;MAGnB,OAAO,Q;K;yFAGX,yB;MAAA,sE;MAA
A,kD;QAaI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,6B;QAC/B,eAAe,SAAS,QAAS,OAAI

B,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAIB,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAAkB,CAAIB,CAAX,GAaKc,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KATbX,C;qGAYBA,2C;MAWI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHc,eAAe,SAAS,QAAS,OAAIB,C;MACf,OAAO,QAAS,UAAhB,C;QACI,QAAQ,SAAS,QAAS,OAAIB,C;QACR,IAAI,UAAW,SAAQ,QAAR,EAAkB,CAAIB,CAAX,GAaKc,CAAtC,C;UACI,WAAW,C;;;MAGnB,OAAO,Q;K;IAGX,iC;MASI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHc,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAAS,O;QACjB,MZvjDG,MAAO,KYujDE,GZvjDF,EYujDO,CZvjDP,C;;MYyjDd,OAAO,G;K;IAGX,iC;MASI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHc,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,MZrIDG,MAAO,KYqlDE,GZrIDF,EYqlDO,CZrIDP,C;;MYulDd,OAAO,G;K;IAGX,iC;MAOI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHc,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,sBAAM,CAAN,KAaJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,2C;MAGI,OAAO,4BAAc,UAAAd,C;K;IAGX,iD;MAOI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHc,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,sBAAM,CAAN,KAaJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,2C;MAGI,OAAO,4BAAc,UAAAd,C;K;IAGX,iD;MAOI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHc,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,4B;MAQI,OAAO,CAAC,oBAAW,U;K;+EAGvB,gC;MAQoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,K;;MACrD,OAAO,I;K;IAUI,uC;MAAA,qB;QACP,eAAO,EAAP,C;QAAA,OACA,E;O;K;IATR,sC;MAOI,OAAO,kBAAL,qBAAL,C;K;IAeW,8C;MAAA,iC;QACd,eAAO,KAAP,EAAC,OAAd,C;QAAA,OACA,O;O;K;IAXR,6C;MASI,OAAO,wBAAW,4BAAX,C;K;kFAMX,yB;MAAA,4F;MAAA,uC;QAEI,eAAe,SAAK,W;QACpB,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,mCAA8B,kCAA9B,C;QAC/B,kBAAqB,QAAS,O;QAC9B,OAAO,QAAS,UAAhB,C;UACI,cAAc,UAAU,WAAV,EAAuB,QAAS,OAAhC,C;;QAEIB,OAAO,W;O;KArBX,C;gGAWBA,yB;MAAA,4F;MAAA,wE;MAAA,uC;QAoBmD,Q;QAL/C,eAAe,SAAK,W;QACpB,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,MAAM,mCAA8B,kCAA9B,C;QAC/B,YAAy,C;QACZ,kBAAqB,QAAS,O;QAC9B,OAAO,QAAS,UAAhB,C;UACI,cAAc,UAAU,oBAAmB,YAAnB,EAAmB,oBAAnB,QAAV,EAAuC,WAAvC,EAAoD,QAAS,OAA7D,C;;QAEIB,OAAO,W;O;KATbX,C;4GAYBA,yB;MAAA,wE;MAAA,uC;QAoBmD,Q;QAL/C,eAAe,SAAK,W;QACpB,IAAI,CAAC,QAAS,UAAAd,C;UAAyB,OAAO,I;QACHc,YAAy,C;QACZ,kBAAqB,QAAS,O;QAC9B,OAAO,QAAS,UAAhB,C;UACI,cAAc,UAAU,oBAAmB,YAAnB,EAAmB,oBAAnB,QAAV,EAAuC,WAAvC,EAAoD,QAAS,OAA7D,C;;QAEIB,OAAO,W;O;KATbX,C;8FAyBA,gC;MAGBI,eAAe,SAAK,W;MACpB,IAAI,CAAC,QAAS,UAAAd,C;QAAyB,OAAO,I;MACHc,kBAAqB,QAAS,O;MAC9B,OAAO,QAAS,UAAhB,C;QACI,cAAc,UAAU,WAAV,EAAuB,QAAS,OAAhC,C;;MAEIB,OAAO,W;K;IAoBS,2I;MAAA,wC;MAAA,6B;MAAA,yB;MAAA,8C;MAAA,gD;MAAA,kD;MAAA,wB;MAAA,+B;MAAA,kC;K;;;sDAAA,Y;;;cACZ,gB;8BAAA,iCAAM,0BAAN,O;kBAAA,2C;uBAAA,yB;cAAA,Q;;;uCACkB,0B;cACF,wD;cAAhB,gB;;;cAAA,KAAGB,yBAAhB,C;gBAAA,gB;;;cAAGB,oC;cACZ,yBAAc,6BAAU,sBAAV,EAAuB,OAAvB,C;cACd,gB;8BAAA,iCAAM,sBAAN,O;kBAAA,2C;uBAAA,yB;cAAA,Q;;;cAFJ,gB;;;cAIJ,W;;;K;IAPgB,wF;MAAA,yD;uBAAA,+H;YAAA,S;iBAAA,Q;;iBAAA,uB;O;K;IAjBpB,sD;MAiBLOAAO,SAAS,iDAAT,C;K;IA4BS,yJ;MAAA,wC;MAAA,6B;MAAA,yB;MAAA,8C;MAAA,8D;MAAA,kD;MAAA,wB;MAAA,yB;MAAA,+B;MAAA,kC;K;;;6DAAA,Y;;;kBAKmc,I;cAJ/C,gB;8BAAA,iCAAM,0BAAN,O;kBAAA,2C;uBAAA,yB;cAAA,Q;;;iCACyC,uCACM,0B;cACF,+D;cAAhB,gB;;;cAAA,KAAGB,yBAAhB,C;gBAAA,gB;;;cAAGB,oC;cACZ,yBAAc,6BAAU,oBAAmB,uBAAnB,EAAmB,+BAAnB,QAAV,EAAuC,sBAAvC,EAAoD,OAApD,C;cACd,gB;8BAAA,iCAAM,sBAAN,O;kBAAA,2C;uBAAA,yB;cAAA,Q;;;cAFJ,gB;;;cAIJ,W;;;K;IARgB,sG;MAAA,yD;uBAAA,6I;YAAA,S;iBAAA,Q;;iBAAA,uB;O;K;IAIBpB,6D;MAkBI,OAAO,SAAS,wDAAT,C;K;IA2BS,4H;MAAA,wC;MAAA,6B;MAAA,yB;MAAA,oD;MAAA,kD;MAAA,4B;MAAA,+B;MAAA,kC;K;;;wDAAA,Y;;;oCACG,wC;cACf,IAAI,mBAAS,UAAb,C;yCACyB,mBAAS,O;gBAC9B,gB;gCAAA,iCAAM,sBAAN,O;oBAAA,2C;yBAAA,yB;gBAAA,Q;;;gBAFJ,gB;;;cAGI,gB;;;cAAA,KA AO,mBAAS,UAAhB,C;gBAAA,gB;;;cACI,yBAAc,6BAAU,sBAAV,EAAuB,mBAAS,OAAhC,C;cACd,gB;8BAAA,iCAAM,sBAAN,O;kBAAA,2C;uBAAA,yB;cAAA,Q;;;cAFJ,gB;;;cAHJ,gB;;;cAQJ,W;;;K;IAVgB,yE;MAAA,yD;uBAAA,gH;YAAA,S;iBAAA,Q;;iBAAA,uB;O;K;IAhBpB,+C;MAGBI,OAAO,SAAS,0CAAT,C;K;IA6BS,0I;MAAA,wC;MAAA,6B;MAAA,yB;MAAA,kE;MAAA,kD;MAAA,4B;MAAA,+B;MAAA,yB;MAAA,kC;K;;;+DAAA,Y;;;cAOuC,Q;oCANpC,+C;cACf,IAAI,mBAAS,UAAb,C;yCACyB,mBAAS,O;gBAC9B,gB;gCAAA,iCAAM,sBAAN,O;oBAAA,2

C;yBAAA,yB;gBAAA,Q;;gBAFJ,gB;;;;;;;iCAGgB,C;cACZ,gB;;;cAAA,KAAO,mBAAS,UAAhB,C;gBAAA,gB;;;
cACI,yBAAC,6BAAU,oBAAMb,uBAAnB,EAAMb,+BAAnB,QAAY,EAAC,sBAAvC,EAAD,mBAAS,OAA7D
,C;cACd,gB;8BAAA,iCAAM,sBAAN,O;kBAAA,2C;uBAAA,yB;cAAA,Q;;cAFJ,gB;;;cAJJ,gB;;;CASJ,W;;;;;
;K;IAXgB,uF;MAAA,yD;uBAAA,8H;YAAA,S;iBAAA,Q;;iBAAA,uB;O;K;IAhBpB,sD;MAGBI,OAAO,SAAS,iD
AAT,C;K;IAcX,+C;MAKBI,OAAO,yBAAY,OAAZ,EAAGb,SAARb,C;K;IAGX,sD;MAMBI,OAAO,gCAAMb,OA
AnB,EA4B,SA5B,C;K;gFAGX,+B;MASoB,Q;MADhB,UAAe,C;MACC,2B;MAAhB,OAAGb,cAAhB,C;QAA
gB,yB;QACZ,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;4FAGX,+B;MASoB,Q;MADhB,UAAkB,G;MAC
F,2B;MAAhB,OAAGb,cAAhB,C;QAAGb,yB;QACZ,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;iFAGX,+B;MA
YoB,Q;MADhB,UAAoB,C;MACJ,2B;MAAhB,OAAGb,cAAhB,C;QAAGb,yB;QACZ,OAAO,SAAS,OAAT,C;;M
AEX,OAAO,G;K;iFAGX,+B;MAYoB,Q;MADhB,UAAe,C;MACC,2B;MAAhB,OAAGb,cAAhB,C;QAAGb,yB;Q
ACZ,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;iFAGX,yB;MAAA,SAWoB,gB;MAXpB,sC;QAYoB,Q;Q
ADhB,Y;QACgB,2B;QAAhB,OAAGb,cAAhB,C;UAAgB,yB;UACZ,cAAO,SAAS,OAAT,CAAP,C;;QAEJ,OAAO
,G;O;KafX,C;iFAkBA,yB;M3B15DA,6B;M2B05DA,sC;QAaoB,Q;QADhB,U3B55DmC,c2B45DnB,C3B55DmB,
C;Q2B65DnB,2B;QAAhB,OAAGb,cAAhB,C;UAAgB,yB;UACZ,M3BhuEiD,c2BguEjD,G3BhuE2D,KAACK,G2Bg
uEzD,SAAS,OAAT,C3BhuEoE,KAAX,IAAf,C;;Q2BkuErD,OAAO,G;O;KAhBX,C;iFAMBA,yB;MX16DA,+B;M
W06DA,sC;QAaoB,Q;QADhB,UX36DqC,eAAW,oBW26D/B,CX36D+B,CAAX,C;QW46DrB,2B;QAAhB,OAAG
b,cAAhB,C;UAAgB,yB;UACZ,MXhvEmD,eWgvEnD,GXhvE8D,KAACK,KWgvE5D,SAAS,OAAT,CXhvEuE,KA
AX,cAAhB,C;;QWkvEvD,OAAO,G;O;KAhBX,C;IAyBe,oD;MAAA,qB;QAAE,e;UAAM,MAAM,gCAAYb,2BA
AwB,mBAAXb,MAAZb,C;;QAAZ,S;O;K;IANjB,qC;MAMI,OAAO,kBAAI,gCAAJ,C;K;IAGX,oC;MAAI,OAAO,
sBAAS,IAAT,EAAe,IAAf,EAAsC,IAAtC,C;K;IAGX,+C;MAKBI,OAAO,sBAAS,IAAT,EAAe,IAAf,EAAsC,IAAt
C,EAAwD,SAAXD,C;K;IASA,0D;MAAA,4B;MAAA,sC;K;IAG0B,+E;MAAA,qB;QAAE,IAAI,CAAC,iBAAD,IA
AY,WAAM,eAAN,cAAhB,C;UAAiC,oBAAU,I;UAA3C,OAAiD,K;;UAAjD,OAA8D,I;O;K;6CAF7F,Y;MACI,k
BAAC,KAAd,C;MACA,OAakB,SAAX,eAAW,EAAO,kEAAP,CAA8E,W;K;;IAT5G,qC;MAMI,kD;K;IASBO,6D;
MAAA,wC;MAAA,4B;K;IAG6B,8D;MAAA,qB;QAAE,OAAM,aAAN,mB;O;K;+CAFIC,Y;MACI,YAAqB,8BA
AT,qBAAS,C;MACrB,OAakB,YAAX,eAAW,EAAU,4CAAV,CAA0B,W;K;;IAjBxD,sC;MAAI,IAAI,Q9B80KG,
YAAQ,C8B90Kf,C;QAAwB,OAAO,S;MAC/B,qD;K;IAqBO,6D;MAAA,wC;MAAA,4B;K;IAMiC,8D;MAAA,qB;
QAAE,OAAM,aAAN,mB;O;K;+CALtC,Y;MACI,YAAqB,4BAAT,qBAAS,C;MACrB,IAAI,KAAM,UAAV,C;QA
CI,OAAO,eAAW,W;;QAEIB,OAakB,YAAX,eAAW,EAAU,4CAAV,CAA0B,W;K;;IANB5D,sC;MAAI,qD;K;IAw
BO,6D;MAAA,wC;MAAA,4B;K;IAMiC,8D;MAAA,qB;QAAE,OAAM,aAAN,mB;O;K;+CALtC,Y;MACI,YAAq
B,8BAAT,qBAAS,C;MACrB,IAAI,KAAM,UAAV,C;QACI,OAAO,eAAW,W;;QAEIB,OAakB,YAAX,eAAW,E
AAU,4CAAV,CAA0B,W;K;;IANB5D,sC;MAAI,qD;K;8FAWJ,yB;MAAA,4C;MAAA,qC;QAOI,OAAO,iBAAM,O
AAN,C;O;KAPX,C;wFAUA,yB;MAAA,+D;MAAA,6B;MAAA,uC;QAYoB,Q;QAFhB,YAAY,gB;QACZ,aAAa,g
B;QACG,2B;QAAhB,OAAGb,cAAhB,C;UAAgB,yB;UACZ,IAAI,UAAU,OAAV,CAAJ,C;YACI,KAAM,WAII,
OAAJ,C;;YAEN,MAAO,WAII,OAAJ,C;;QAGf,OAAO,cAAK,KAAL,EAAY,MAAZ,C;O;KANBX,C;IASBA,oC;
MAMI,OAA6C,UAAtC,YAAW,SAAX,EAaiB,YAAW,OAAX,EAajB,EAAsC,C;K;IAGjD,qC;MASI,OAAy,OA
AL,SAAK,EAAc,OAAT,QAAS,CAAd,C;K;IAGhB,qC;MASI,OAA+C,UAAx,C,YAAW,SAAX,EAa0B,aAAT,QA
AS,CAA1B,EAAwC,C;K;IAGnD,sC;MASI,OAakC,UAA3B,YAAW,SAAX,EAaiB,QAajB,EAA2B,C;K;4FAGt
C,yB;MAAA,0C;MAAA,qC;QAOI,OAAO,gBAAK,OAAL,C;O;KAPX,C;IAUA,2D;MAGb+C,oB;QAAA,OAAy,
C;MAAG,8B;QAAA,iBAA0B,K;MACpF,OAAO,8BAAiB,IAAjB,EAauB,IAAvB,EAA6B,cAA7B,EAA2D,KAA3
D,C;K;IAGX,sE;MAKbKD,oB;QAAA,OAAy,C;MAAG,8B;QAAA,iBAA0B,K;MACvF,OAAwE,OAAjE,8BAAiB
,IAAjB,EAauB,IAAvB,EAA6B,cAA7B,EAA2D,IAA3D,CAAiE,EAAL,SAAJ,C;K;IAYpC,4B;MAAY,cAAM,EA
AN,C;K;IATpD,kC;MASI,OAAO,oBAAGb,SAAhB,EAAsB,KAAtB,EAA6B,UAA7B,C;K;IAGX,6C;MAUI,OAA
O,oBAAGb,SAAhB,EAAsB,KAAtB,EAA6B,SAA7B,C;K;IAcY,kC;MAAU,aAAK,CAAL,C;K;IAXjC,kC;MAWI,
OAAO,yBAAY,kBAAZ,C;K;IAeiB,wH;MAAA,wC;MAAA,6B;MAAA,yB;MAAA,gD;MAAA,kD;MAAA,4B;M
AAA,2B;MAAA,wB;MAAA,kC;K;;sDAAA,Y;;;oCACL,sC;cACf,IAAI,CAAC,mBAAS,UAAAd,C;gBAAYb,M;;
gBAAZb,gB;;;;;mCACc,mBAAS,O;cACvB,gB;;;cAAA,KAAO,mBAAS,UAAhB,C;gBAAA,gB;;;gCACe,mBAA
S,O;cACpB,gB;8BAAA,iCAAM,6BAAU,kBAAV,EAAMb,eAANb,CAAN,O;kBAAA,2C;uBAAA,yB;cAAA,Q;;c
ACA,qBAAU,e;cAHd,gB;;;cAKJ,W;;;;;;;K;IATwB,uE;MAAA,yD;uBAAA,4G;YAAA,S;iBAAA,Q;;iBAAA,

C;QAOI,OAAO,gBAAK,OAAL,C;O;KAPX,C;InBnIA,oD;MAMuF,wC;K;IANvF,8CAOI,Y;MAAuC,8B;K;IAP3
C,gF;ICGA,oD;MAQuF,wC;K;IARvF,8CASI,Y;MAAuC,8B;K;IAT3C,gF;gGmBYA,yB;MAAA,uD;MAAA,gC;M
AAA,iD;QAOI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAcC,qBAAI,KA AJ,CAAtC,GAAcD,uBAAa,KA
Ab,E;O;KAPjE,C;gGAUA,yB;MAAA,+C;MAAA,mC;QAOI,OAA Y,UAAL,SAAK,EAAU,KA AV,C;O;KAPhB,C;
0EAUA,yB;MA4EA,6C;MAAA,oC;MAAA,gC;MA5EA,uC;QAOW,sB;;UAYES,Q;UAAA,0B;UAAhB,OAAGB,c
AAhB,C;YAAgB,oC;YAAM,IAzEH,SAyEO,CAAU,oBAAV,CAAJ,C;cAAwB,qBAAO,O;cAAP,uB;;;UAC9C,qB
AAO,I;;;QA1EP,yB;O;KAPJ,C;kFAUA,yB;MAwJA,mD;MAAA,+C;MAAA,oC;MAxJA,uC;QAOW,qB;;UAUJO,
Q;UAAA,OAAa,SAAR,sBAAQ,CAAb,W;UAA d,OAAC,cAA d,C;YAAc,uB;YACV,cAAc,qBAAK,KAAL,C;YAC
d,IAzJc,SAyJV,CAAU,oBAAV,CAAJ,C;cAAwB,oBAAO,O;cAAP,sB;;;UAE5B,oBAAO,I;;;QA3JP,wB;O;KAPJ,
C;IAUA,6B;MAKI,ICkOgD,qBAAU,CDIO1D,C;QACI,MAAM,2BA AuB,yBA AvB,C;MACV,OAAO,qBAAK,CA
AL,C;K;4EAGX,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,iE;MAAA,uC;QAKoB,Q;QAAA,0B;QAAhB,OA
AgB,cAAhB,C;UAAgB,oC;UAAM,IAAI,UAAU,oBAAV,CAAJ,C;YAAwB,OAAO,O;;QACrD,MAAM,gCAAuB,
6DAAvB,C;O;KANV,C;6FASA,yB;MAAA,iE;MAY A,6C;MAAA,oC;MAAA,gC;MAZA,uC;QASW,Q;QAAA,+B
;;UAYS,U;UAAA,4B;UAAhB,OAAGB,gBAAhB,C;YAAgB,sC;YACZ,aAbwB,SAaX,CAAU,oBAAV,C;YACb,IA
AI,cAAJ,C;cACI,8BAAO,M;cAAP,gC;;;UAGR,8BAAO,I;;;QAIBA,kC;QAAA,iB;UAAmC,MAAM,gCAAuB,sEA
AvB,C;;QAAhD,OAAO,I;O;KATX,C;yGAYA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,uC;QASoB,Q;QAA
A,0B;QAAhB,OAAGB,cAAhB,C;UAAgB,oC;UACZ,aAAa,UAAU,oBAAV,C;UACb,IAAI,cAAJ,C;YACI,OAAO,
M;;;QAGf,OAAO,I;O;KafX,C;IAkBA,mC;MAII,OCKLgD,qBAAU,CDILnD,GAAe,IAAf,GAAYB,qBAAK,CAAL
,C;K;wFAGpC,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,uC;QAIoB,Q;QAAA,0B;QAAhB,OAAGB,cAAhB,C
;UAAgB,oC;UAAM,IAAI,UAAU,oBAAV,CAAJ,C;YAAwB,OAAO,O;;QACrD,OAAO,I;O;KALX,C;mFAQA,yB
;MAAA,uD;MAAA,gC;MAAA,iD;QAKI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAcC,qBAAI,KA AJ,CA
AtC,GAAcD,uBAAa,KAAb,E;O;KALjE,C;IAQA,uC;MAMI,OAAW,SAAS,CAAT,IAAc,SAAS,2BAA3B,GAAcC
,qBAAI,KA AJ,CAAtC,GAAcD,I;K;0FAGjE,yB;MAAA,mD;MAAA,oC;MAAA,uC;QAIkB,gC;QAAA,6B;QAAA,
mB;QAAA,kB;QAAA,kB;QAAd,0D;UACI,IAAI,UAAU,iCAAK,KAAL,EAAV,CAAJ,C;YACI,OAAO,K;;;QAGf,
OAAO,E;O;KATX,C;wFAYA,yB;MAAA,mD;MAAA,+C;MAAA,oC;MAAA,uC;QAIkB,Q;QAAA,OAAQ,SAAR
,sBAAQ,CAAR,W;QAAd,OAAC,cAA d,C;UAAc,uB;UACV,IAAI,UAAU,iCAAK,KAAL,EAAV,CAAJ,C;YACI,O
AAO,K;;;QAGf,OAAO,E;O;KATX,C;IAYA,4B;MAQI,ICsHgD,qBAAU,CDtH1D,C;QACI,MAAM,2BA AuB,yB
AAvB,C;MACV,OAAO,qBAAK,2BAAL,C;K;0EAGX,yB;MAAA,mD;MAAA,+C;MAAA,oC;MAAA,iE;MAAA,
uC;QAQkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAAC,cAA d,C;UAAc,uB;UACV,cAA
c,qBAAK,KAAL,C;UACd,IAAI,UAAU,oBAAV,CAAJ,C;YAAwB,OAAO,O;;QAE nC,MAAM,gCAAuB,6DAAv
B,C;O;KAZV,C;IAeA,kC;MAMI,OC4FgD,qBAAU,CD5FnD,GAAe,IAAf,GAAYB,qBAAK,mBAAS,CAAT,IAAL
,C;K;sFAGpC,yB;MAAA,mD;MAAA,+C;MAAA,oC;MAAA,uC;QAMkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,
CAAQ,CAAb,W;QAAd,OAAC,cAA d,C;UAAc,uB;UACV,cAAc,qBAAK,KAAL,C;UACd,IAAI,UAAU,oBAAV,C
AAJ,C;YAAwB,OAAO,O;;QAE nC,OAAO,I;O;KAVX,C;8EAaA,yB;MAAA,mC;MAAA,yC;MAAA,4B;QAQI,O
AAO,kBAAO,cAAP,C;O;KARX,C;IAWA,sC;MAOI,IC0DgD,qBAAU,CD1D1D,C;QACI,MAAM,2BA AuB,yBA
AvB,C;MACV,OAAO,qBAAI,MAAO,iBAAQ,gBAAR,CAAX,C;K;0FAGX,yB;MAAA,mC;MAAA,qD;MAAA,4
B;QAOI,OAAO,wBAAa,cAAb,C;O;KAPX,C;IAUA,4C;MAMI,ICqCgD,qBAAU,CDrC1D,C;QACI,OAAO,I;MA
CX,OAAO,qBAAI,MAAO,iBAAQ,gBAAR,CAAX,C;K;IAGX,8B;MAIiB,IAAN,I;MAAA,QAAM,gBAAN,C;aA
CH,C;UAAK,MAAM,2BA AuB,yBA AvB,C;aACX,C;UAAK,4BAAK,CAAL,C;UAAL,K;;UACQ,MAAM,gCAAy
B,0CAAzB,C;;MAHIB,W;K;8EAOJ,yB;MAAA,6C;MAAA,oC;MAAA,kF;MAAA,gC;MAAA,iE;MAAA,8B;MA
AA,uC;QAMoB,UAST,M;QAXP,aAAoB,I;QACpB,YAAY,K;QACI,0B;QAAhB,OAAGB,cAAhB,C;UAAgB,oC;U
ACZ,IAAI,UAAU,oBAAV,CAAJ,C;YACI,IAAI,KA AJ,C;cAAW,MAAM,8BAAYB,wDAAzB,C;YACjB,SAAS,O;
YACT,QAAQ,I;;;QAGhB,IAAI,CAAC,KAAL,C;UAAY,MAAM,gCAAuB,6DAAvB,C;QAEIB,OAAO,4E;O;Kaf
X,C;IAkBA,oC;MAII,OAAW,qBAAU,CAAd,GAAiB,qBAAK,CAAL,CAAjB,GAA8B,I;K;0FAGzC,yB;MAAA,6
C;MAAA,oC;MAAA,gC;MAAA,uC;QAMoB,Q;QAFhB,aAAoB,I;QACpB,YAAY,K;QACI,0B;QAAhB,OAAGB,c
AAhB,C;UAAgB,oC;UACZ,IAAI,UAAU,oBAAV,CAAJ,C;YACI,IAAI,KA AJ,C;cAAW,OAAO,I;YACIB,SAAS,
O;YACT,QAAQ,I;;;QAGhB,IAAI,CAAC,KAAL,C;UAAY,OAAO,I;QACnB,OAAO,M;O;KadX,C;IAiBA,+B;MI
BzRI,IAAI,EkBiSI,KA AK,CIBjST,CAAJ,C;QACI,ckBgSc,wD;QIB/Rd,MAAM,gCAAyB,OAAQ,WAAjC,C;;MkB

gSV,OAAO,8BAAC,eAAF,CAAE,EAAa,gBAAb,CAAd,EAAoC,gBAAPC,C;K;IAGX,+B;MIBrSI,IAAI,EkB6SI,K
AAK,CIB7ST,CAAJ,C;QACI,ckB4Sc,wD;QIB3Sd,MAAM,gCAAyB,OAAQ,WAAjC,C;;MkB4SV,OLhH6E,oBKg
H1D,eAAF,CAAE,EAAa,gBAAb,CLhH0D,C;K;IKmHjF,kC;MIBjTI,IAAI,EkByTI,KAAK,CIBzTT,CAAJ,C;QAC
I,ckBwTc,wD;QIBvTd,MAAM,gCAAyB,OAAQ,WAAjC,C;;MkBwTV,OAAO,mBAAKB,gBAAZ,mBAAS,CAAT
,IAAY,EAAc,CAAd,CAAIB,C;K;IAGX,mC;MIB7TI,IAAI,EkBqUI,KAAK,CIBrUT,CAAJ,C;QACI,ckBoUc,wD;
QIBnUd,MAAM,gCAAyB,OAAQ,WAAjC,C;;MkB0UV,OAAO,mBAAKB,gBAAZ,mBAAS,CAAT,IAAY,EAAc,
CAAd,CAAIB,C;K;2FAGX,yB;MAAA,uD;MAAA,oC;MAAA,uC;QAMI,iBAAc,wBAAd,WAA+B,CAA/B,U;UA
CI,IAAI,CAAC,UAAU,iCAAK,KAAL,EAAV,CAAL,C;YACI,OAAO,8BAAY,CAAZ,EAAe,QAAQ,CAAR,IAAf,
C;QACf,OAAO,E;O;KATX,C;4FAYA,yB;MAAA,uD;MAAA,oC;MAAA,uC;QAMI,iBAAc,wBAAd,WAA+B,CA
A/B,U;UACI,IAAI,CAAC,UAAU,iCAAK,KAAL,EAAV,CAAL,C;YACI,OL5JoF,oBK4JnE,CL5JmE,EK4JhE,QA
AQ,CAAR,IL5JgE,C;;QK6J5F,OAAO,E;O;KATX,C;oFAYA,yB;MAAA,mD;MAAA,oC;MAAA,uC;QAMuB,UA
AL,MAAK,EAAL,MAAK,EAAL,M;QAAK,mBAAL,SAAK,C;QAAL,mB;QAAA,kB;QAAA,kB;QAAd,0D;UACI
,IAAI,CAAC,UAAU,iCAAK,KAAL,EAAV,CAAL,C;YACI,OAAO,8BAAY,KAAZ,EAAmB,gBAAnB,C;QACf,O
AAO,E;O;KATX,C;oFAYA,yB;MAAA,mD;MAAA,oC;MAAA,uC;QAMuB,UAAL,MAAK,EAAL,MAAK,EAAL
,M;QAAK,mBAAL,SAAK,C;QAAL,mB;QAAA,kB;QAAA,kB;QAAd,0D;UACI,IAAI,CAAC,UAAU,iCAAK,KA
AL,EAAV,CAAL,C;YACI,OLvLqE,oBKuLpD,KLvLoD,C;;QKwL7E,OAAO,E;O;KATX,C;8EAYA,yB;MAAA,y
D;MAkFA,oC;MAIFA,uC;QAMW,kBAAS,oB;QAKFM,Q;QAAA,uB;QAAtB,iBAAc,CAAd,wB;UACI,cAAc,qBA
AI,KAAJ,C;UACd,IAPf6B,SAoFzB,CAAU,oBAAV,CAAJ,C;YAAwB,WAAy,gBAAO,OAAP,C;;QApFxC,OAsF
O,W;O;KA5FX,C;8EASA,yB;MAAA,yD;MAyEA,oC;MAzEA,uC;QAMW,kBAAS,oB;QAYEM,Q;QAAA,uB;QA
AtB,iBAAc,CAAd,wB;UACI,cAAc,qBAAI,KAAJ,C;UACd,IA3E6B,SA2EzB,CAAU,oBAAV,CAAJ,C;YAAwB,
WAAy,gBAAO,OAAP,C;;QA3ExC,OA6EO,WA7EqC,W;O;KANhD,C;4FASA,yB;MAAA,yD;MASBA,gC;MA+
sBA,6C;MAAA,oC;MARuBA,uC;QAQW,kBAAGB,oB;QAouBV,gB;QADb,YAAy,C;QACC,0B;QAAb,OAAa,cA
Ab,C;UAAa,iC;UAAM,eAAO,cAAP,EAAO,sBAAP,S;UAAA,cAAGB,iB;UA7sB/B,IAvBoC,SAuBhC,CAAU,OA
AV,EAAiB,OAAjB,CAAJ,C;YAA2C,2BAAO,kBAAP,C;;QAvB/C,OAYBO,W;O;KAjCX,C;4FAWA,yB;MAAA,y
D;MAWA,gC;MA+sBA,6C;MAAA,oC;MA1tBA,uC;QAQW,kBAAGB,oB;QAYtBV,gB;QADb,YAAy,C;QACC,0
B;QAAb,OAAa,cAAb,C;UAAa,iC;UAAM,eAAO,cAAP,EAAO,sBAAP,S;UAAA,cAAGB,iB;UA7sB/B,IAZOC,SA
YhC,CAAU,OA AV,EAAiB,OAAjB,CAAJ,C;YAA2C,2BAAO,kBAAP,C;;QAZ/C,OAco,WAd4C,W;O;KARvD,C
;gGAWA,yB;MAAA,gC;MA+sBA,6C;MAAA,oC;MA/sBA,oD;QAsBiB,gB;QADb,YAAy,C;QACC,0B;QAAb,O
AAa,cAAb,C;UAAa,iC;UAAM,eAAO,cAAP,EAAO,sBAAP,S;UAAA,cAAGB,iB;UA7sB/B,IAAI,UAAU,OA AV,
EAAiB,OAAjB,CAAJ,C;YAA2C,2BAAO,kBAAP,C;;QAE/C,OAAO,W;O;KAXX,C;oFAcA,yB;MAAA,yD;MAK
BA,6C;MAAA,oC;MAAA,gC;MAIBA,uC;QAMW,kBAAY,oB;QAKBH,Q;QAAA,0B;QAAhB,OAAGB,cAAhB,C;
UAAgB,oC;UAAM,IAAI,CAIBU,SAkBT,CAAU,oBAAV,CAAL,C;YAAyB,WAAy,gBAAO,OAAP,C;;QAIb3D,
OAmBO,W;O;KAZBX,C;oFASA,yB;MAAA,yD;MASA,6C;MAAA,oC;MAAA,gC;MATA,uC;QAMW,kBAAY,o
B;QASH,Q;QAAA,0B;QAAhB,OAAGB,cAAhB,C;UAAgB,oC;UAAM,IAAI,CATU,SAST,CAAU,oBAAV,CAAL
,C;YAAyB,WAAy,gBAAO,OAAP,C;;QAT3D,OAuO,WAVwC,W;O;KANnD,C;wFASA,yB;MAAA,6C;MAAA,
oC;MAAA,gC;MAAA,oD;QAMoB,Q;QAAA,0B;QAAhB,OAAGB,cAAhB,C;UAAgB,oC;UAAM,IAAI,CAAC,U
AAU,oBAAV,CAAL,C;YAAyB,WAAy,gBAAO,OAAP,C;;QAC3D,OAAO,W;O;KAPX,C;kFAUA,yB;MAAA,o
C;MAAA,oD;QAM0B,Q;QAAA,uB;QAAtB,iBAAc,CAAd,wB;UACI,cAAc,qBAAI,KAAJ,C;UACd,IAAI,UAAU,
oBAAV,CAAJ,C;YAAwB,WAAy,gBAAO,OAAP,C;;QAExC,OAAO,W;O;KAVX,C;IAaA,sC;MAII,IAAI,OAAQ
,UAAZ,C;QAAuB,OAAO,E;MAC9B,OAAO,yBAAY,OAAZ,C;K;IAGX,sC;MAII,IAAI,OAAQ,UAAZ,C;QAAuB
,OAAO,E;MAC9B,OAAO,uBAAU,OA AV,C;K;IAGX,sC;MAOc,Q;MAHV,WAAmB,wBAAR,OAAQ,EAAwB,E
AAxB,C;MACnB,IAAI,SAAQ,CAAZ,C;QAae,OAAO,E;MAcIB,aAAa,mBAAc,IAAd,C;MACH,yB;MAAV,OA
AU,cAAV,C;QAAU,mB;QACN,MAAO,gBAAO,qBAAI,CAAJ,CAAP,C;;MAEX,OAAO,M;K;4EAGX,yB;MAA
A,8B;MAAA,uC;MAAA,qC;QAKY,Q;QAAR,OAA8B,MAAtB,2DAAsB,EAAM,OAAN,CAAE,W;O;KALjD,C;I
AQA,+B;MIB7fI,IAAI,EkBqgBI,KAAK,CIBrgBT,CAAJ,C;QACI,ckBogBc,wD;QIBngBd,MAAM,gCAAyB,OAA
Q,WAAjC,C;;MkBogBV,OAAO,8BAAY,CAAZ,EAAiB,eAAF,CAAE,EAAa,gBAAb,CAAjB,C;K;IAGX,+B;MIB
zgBI,IAAI,EkBihBI,KAAK,CIBjhBT,CAAJ,C;QACI,ckBghBc,wD;QIB/gBd,MAAM,gCAAyB,OAAQ,WAAjC,C;
;MkBghBV,OLjV4F,oBKiv3E,CLjV2E,EKivtE,eAAF,CAAE,EAAa,gBAAb,CLjVsE,C;K;IKoVhG,kC;MIBrhBI,I

AAI,EkB6hBI,KAAK,CIB7hBT,CAAJ,C;QACI,ckB4hBc,wD;QIB3hBd,MAAM,gCAAYB,OAAQ,WAAjC,C;;Mk
B4hBV,aAAa,gB;MACb,OAAO,8BAAY,SAAW,eAAF,CAAE,EAAa,MAAb,CAAX,IAAZ,EAA6C,MAA7C,C;K;
IAGX,mC;MIBliBI,IAAI,EkB0iBI,KAAK,CIB1iBT,CAAJ,C;QACI,ckByiBc,wD;QIBxiBd,MAAM,gCAAYB,OAA
Q,WAAjC,C;;MkByiBV,aAAa,gB;MACb,OL9W6E,oBK8W5D,SAAW,eAAF,CAAE,EAAa,MAAb,CAAX,IL9W
4D,C;K;2FKiXjF,yB;MAAA,uD;MAAA,oC;MAAA,uC;QAMI,iBAAc,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,C
AAC,UAAU,iCAAK,KAAL,EAAV,CAAL,C;YACI,OAAO,8BAAY,QAAQ,CAAR,IAAZ,EAAuB,gBAAvB,C;;;Q
AGf,OAAO,8BAAY,CAAZ,EAAe,gBAAf,C;O;KAXX,C;4FAcA,yB;MAAA,uD;MAAA,oC;MAAA,uC;QAMI,iB
AAc,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,iCAAK,KAAL,EAAV,CAAL,C;YACI,OLvYqE,oB
KuYpD,QAAQ,CAAR,ILvYoD,C;;;QK0Y7E,OAAO,S;O;KAXX,C;oFAcA,yB;MAAA,oC;MAAA,uC;QAM0B,Q;
QAAA,uB;QAAiB,iBAAc,CAAd,wB;UACI,IAAI,CAAC,UAAU,iCAAI,KAAJ,EAAV,CAAL,C;YACI,OAAO,8B
AAY,CAAZ,EAAe,KAaf,C;;QAEf,OAAO,8BAAY,CAAZ,EAAe,gBAAf,C;O;KAVX,C;oFAaA,yB;MAAA,oC;M
AAA,uC;QAM0B,Q;QAAA,uB;QAAiB,iBAAc,CAAd,wB;UACI,IAAI,CAAC,UAAU,iCAAI,KAAJ,EAAV,CAA
L,C;YACI,OL/ZoF,oBK+ZnE,CL/ZmE,EK+ZhE,KL/ZgE,C;;QKia5F,OAAO,S;O;KAVX,C;IAaA,gC;MAII,OAAO
,qBAAc,SAAd,CAAoB,U;K;kFAG/B,yB;MAAA,8B;MAAA,6C;MAAA,4B;QAKY,Q;QAAR,OAA8B,SAAtB,2D
AAsB,CAAW,W;O;KAL7C,C;oFAQA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MA4EA,6C;MAAA,oC;MAAA,gC;
MA5EA,uC;QAWI,eAAmC,cAApB,YAAY,gBAAZ,CAAoB,EAAc,EAAAd,C;QAC5B,kBAAY,mBAAoB,QAApB,
C;QAyEH,Q;QAAA,0B;QAAhB,OAAGB,cAAhB,C;UAAgB,oC;UACZ,WA1E8C,SA0E/B,CAAU,oBAAV,C;UzB
9EnB,wBAAI,IAAK,MAAT,EAAGB,IAAK,OAAR,B,C;;QyBIA,OA4EO,W;O;KAXFX,C;wFAeA,yB;MAAA,0D;M
AAA,yD;MAAA,uE;MA6BA,6C;MAAA,oC;MAAA,gC;MA7BA,yC;QAWI,eAAmC,cAApB,YAAY,gBAAZ,CA
AoB,EAAc,EAAAd,C;QAC5B,kBAAc,mBAAuB,QAAvB,C;QA2BL,Q;QAAA,0B;QAAhB,OAAGB,cAAhB,C;UA
AgB,oC;UACZ,WAAY,aA5BuC,WA4BnC,CAAY,oBAAZ,CAAJ,EAA0B,oBAA1B,C;;QA5BhB,OA8BO,W;O;K
A1CX,C;wFAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MA8BA,6C;MAAA,oC;MAAA,gC;MA9BA,yD;QAUl,e
AAmC,cAApB,YAAY,gBAAZ,CAAoB,EAAc,EAAAd,C;QAC5B,kBAAc,mBAAoB,QAApB,C;QA6BL,Q;QAAA,
0B;QAAhB,OAAGB,cAAhB,C;UAAgB,oC;UACZ,WAAY,aA9BoC,WA8BhC,CAAY,oBAAZ,CAAJ,EA9BiD,cA
8BvB,CAAe,oBAAf,CAA1B,C;;QA9BhB,OAAGB,cAAhB,C;UAAgB,oC;UACZ,WAAY,aAAI,YAAY,oBAAZ,C
AAJ,EAA0B,oBAA1B,C;;QAEhB,OAAO,W;O;KAbX,C;4FAGBA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,
sE;QAUoB,Q;QAAA,0B;QAAhB,OAAGB,cAAhB,C;UAAgB,oC;UACZ,WAAY,aAAI,YAAY,oBAAZ,CAAJ,EA
A0B,eAAe,oBAAf,CAA1B,C;;QAEhB,OAAO,W;O;KAbX,C;wFAGBA,yB;MAAA,6C;MAAA,oC;MAAA,gC;M
AAA,oD;QASoB,Q;QAAA,0B;QAAhB,OAAGB,cAAhB,C;UAAgB,oC;UACZ,WAAe,UAAU,oBAAV,C;UzB9En
B,wBAAI,IAAK,MAAT,EAAGB,IAAK,OAAR,B,C;;QyBgFA,OAAO,W;O;KAZX,C;4FAeA,yB;MAAA,uD;MAA
A,0D;MAAA,yD;MAAA,uE;MAGBA,6C;MAAA,oC;MAAA,gC;MAhBA,2C;QAYI,aAAa,mBAA6D,cAAtC,YA
AmB,aAAP,gBAAO,EAAa,GAAb,CAAnB,CAAsC,EAAc,EAAAd,CAA7D,C;QAcG,Q;QAAA,0B;QAAhB,OAAGB
,cAAhB,C;UAAgB,oC;UAbO,MAcP,aAAI,oBAAJ,EAAe,aAcF,CAAc,oBAAAd,CAAb,C;;QAdhB,OAAuB,M;O;KA
b3B,C;+FAGBA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,wD;QAUoB,Q;QAAA,0B;QAAhB,OAAGB,cAAh
B,C;UAAgB,oC;UACZ,WAAY,aAAI,oBAAJ,EAAa,cAAc,oBAAAd,CAAb,C;;QAEhB,OAAO,W;O;KAbX,C;IAGB
A,iD;MAIiB,Q;MAAA,4B;MAAb,OAAa,cAAb,C;QAaa,iC;QACT,WAAY,WAAI,iBAAJ,C;;MAEhB,OAAO,W;
K;IAGX,iC;MAII,OAAO,2BAAa,eAAc,YAAmB,eAAP,gBAAO,EAAa,GAAb,CAAnB,CAAd,CAAb,C;K;IAGX,8
B;MAIiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,kB;UAAL,K;aACA,C;UAAK,cAAO,iCAAK,CAAL
,EAAP,C;UAAL,K;;UACa,wBAAL,SAAK,C;UAHV,K;;MAAP,W;K;IAOJ,qC;MAII,OAAO,2BAAa,iBAAgB,gB
AAhB,CAAb,C;K;IAGX,6B;MAMiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,iB;UAAL,K;aACA,C;U
AAK,aAAM,iCAAK,CAAL,EAAN,C;UAAL,K;;UACQ,kCAAa,qBAAoB,YAAmB,eAAP,gBAAO,EAAa,GAAb,
CAAnB,CAApB,CAAb,C;UAHL,K;;MAAP,W;K;gFAOJ,yB;MAAA,+D;MA0CA,6C;MAAA,oC;MAAA,gD;MA
AA,gC;MA1CA,uC;QAMW,kBAAU,gB;QAwCD,Q;QAAA,0B;QAAhB,OAAGB,cAAhB,C;UAAgB,oC;UACZ,W
AzC6B,SAyCIB,CAAU,oBAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QA1ChB,OA4CO,W;O;KAIDX,C;8FA
SA,yB;MAAA,+D;MAeA,6C;MAAA,oC;MAAA,gD;MAAA,gC;MAfA,uC;QAYW,kBAAiB,gB;QAcR,gB;QADh
B,YAAY,C;QACI,0B;QAAhB,OAAGB,cAAhB,C;UAAgB,oC;UACZ,WAFoC,SAezB,EAAU,cAAV,EAAU,sBAA
V,WAAmB,oBAAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAhBhB,OAkBO,W;O;KA9BX,C;kGAeA,yB;M

AAA,6C;MAAA,oC;MAAA,gD;MAAA,gC;MAAA,oD;QAWoB,UACS,M;QAFzB,YAA Y,C;QACI,0B;QAAhB,O
AAgB,cAAhB,C;UAAgB,oC;UACZ,WAAW,WAAU,cAAV,EAAU,sBAAV,WAAmB,oBAAnB,C;UACC,OAAZ,
WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KAFx,C;oFAkBA,yB;MAAA,6C;MAAA,oC;MAAA,gD;MAAA,g
C;MAAA,oD;QAIoB,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,WAAW,UAAU,oBAAV,C;UA
CC,OAAZ,WAA Y,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KARX,C;gFAWA,yB;MAAA,wE;MAyBA,6C;MAAA,o
C;MAAA,+D;MAAA,gC;MAzBA,yC;QASW,kBAAU,oB;QAYBD,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAA
gB,oC;UACZ,UA1BoD,WA0B1C,CAAY,oBAAZ,C;UzBrjBP,U;UADP,YyBujBe,WzBvjBH,WyBujBwB,GzBvjB
xB,C;UACL,IAAI,aAAJ,C;YACH,ayBqjBuC,gB;YAA5B,WzBpjBX,ayBojBgC,GzBpjBhC,EAAS,MAAT,C;YAC
A,e;;YAEA,c;;UyBijBA,iB;UACA,IAAK,WAAI,oBAAJ,C;;QA5BT,OA8BO,W;O;KA vCX,C;gFAYA,yB;MAAA,
wE;MA8BA,6C;MAAA,oC;MAAA,+D;MAAA,gC;MA9BA,yD;QAUW,kBAAU,oB;QA8BD,Q;QAAA,0B;QAAh
B,OAAgB,cAAhB,C;UAAgB,oC;UACZ,UA/BiD,WA+BvC,CAAY,oBAAZ,C;UzBvkBP,U;UADP,YyBykBe,WzB
zkBH,WyBykBwB,GzBzkBxB,C;UACL,IAAI,aAAJ,C;YACH,ayBukBuC,gB;YAA5B,WzBtkBX,ayBskBgC,GzBt
kBhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UyBmkBA,iB;UACA,IAAK,WAjCyD,cAiCrD,CAAe,oBAAf,CAAJ,C;
;QAJCT,OAmCO,W;O;KA7CX,C;oFAaA,yB;MAAA,6C;MAAA,oC;MAAA,+D;MAAA,gC;MAAA,sD;QASoB,Q
;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,UAAU,YAA Y,oBAAZ,C;UzBrjBP,U;UADP,YyBujBe
,WzBvjBH,WyBujBwB,GzBvjBxB,C;UACL,IAAI,aAAJ,C;YACH,ayBqjBuC,gB;YAA5B,WzBpjBX,ayBojBgC,G
zBpjBhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UyBijBA,iB;UACA,IAAK,WAAI,oBAAJ,C;;QAET,OAAO,W;O;K
AdX,C;oFAiBA,yB;MAAA,6C;MAAA,oC;MAAA,+D;MAAA,gC;MAAA,sE;QAUoB,Q;QAAA,0B;QAAhB,OA
AgB,cAAhB,C;UAAgB,oC;UACZ,UAAU,YAA Y,oBAAZ,C;UzBvkBP,U;UADP,YyBykBe,WzBzkBH,WyBykBw
B,GzBzkBxB,C;UACL,IAAI,aAAJ,C;YACH,ayBukBuC,gB;YAA5B,WzBtkBX,ayBskBgC,GzBtkBhC,EAAS,MA
AT,C;YACA,e;;YAEA,c;;UyBmkBA,iB;UACA,IAAK,WAAI,eAAe,oBAAf,CAAJ,C;;QAET,OAAO,W;O;KAFx,
C;qFAkBA,yB;MAAA,6C;MAAA,oC;MAAA,kC;MAAA,4C;MAAA,wE;QAQW,sC;QAAA,8C;O;MARX,oDAS
Q,Y;QAAgD,OAAgB,SAAhB,oBAAgB,C;O;MATxE,iDAUQ,mB;QAAuC,gCAAY,oBAAZ,C;O;MAV/C,gF;MA
AA,yC;QAQI,2D;O;KARJ,C;wEAca,yB;MAAA,gE;MAyEA,6C;MAAA,oC;MAAA,gC;MAzEA,uC;QAOW,kBA
AM,eAAa,gBAAb,C;QAuEA,Q;QAAA,0B;QAAb,OAAa,cAAb,C;UAAa,iC;UACT,WAA Y,WAXEmB,SAwEf,CA
AU,iBAAV,CAAJ,C;;QAxEhB,OAYEO,W;O;KAhFX,C;sFAUA,yB;MAAA,gE;MA+BA,6C;MAAA,oC;MAAA,g
C;MA/BA,uC;QAOW,kBAAa,eAAa,gBAAb,C;QA gCP,gB;QADb,YAA Y,C;QACC,0B;QAAb,OAAa,cAAb,C;UA
Aa,iC;UACT,WAA Y,WAJC0B,SAiCtB,EAAU,cAAV,EAAU,sBAAV,WAAmB,iBAAnB,CAAJ,C;;QAJChB,OAk
CO,W;O;KAZCX,C;mGAUA,yB;MAAA,+D;MAUA,gC;MAoLA,6C;MAAA,oC;MA9LA,uC;QAOW,kBAAoB,g
B;QA8Ld,gB;QADb,YAA Y,C;QACC,0B;QAAb,OAAa,cAAb,C;UAAa,iC;UApLsB,U;UAAA,cAVQ,SAUR,EAo
LT,cApLS,EAoLT,sBApLS,WAO LA,iBApLA,W;YAA6C,6B;;;QAVhF,OAwo,W;O;KAIBX,C;uGAUA,yB;MAA
A,gC;MAoLA,6C;MAAA,oC;MApLA,oD;QA2LiB,gB;QADb,YAA Y,C;QACC,0B;QAAb,OAAa,cAAb,C;UAAa,i
C;UApLsB,U;UAAA,yBAoLT,cApLS,EAoLT,sBApLS,WAO LA,iBApLA,W;YAA6C,6B;;;QACHF,OAAO,W;O;K
ARX,C;0FAWA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,oD;QAQiB,UACiB,M;QAF9B,YAA Y,C;QACC,0
B;QAAb,OAAa,cAAb,C;UAAa,iC;UACT,WAA Y,WAAI,WAAU,cAAV,EAAU,sBAAV,WAAmB,iBAAnB,CAAJ
,C;;QACHB,OAAO,W;O;KAVX,C;qFAaA,yB;MAAA,+D;MAUA,gC;MA2IA,6C;MAAA,oC;MARJA,uC;QAOW,
kBAAa,gB;QAKJJ,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UA1IK,U;UAAA,cARe,SAQf,CA0IQ,oB
A1IR,W;YAA sC,6B;;;QAR3D,OASO,W;O;KAhBX,C;yFAUA,yB;MAAA,gC;MA2IA,6C;MAAA,oC;MA3IA,oD;
QA+IoB,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UA1IK,U;UAAA,wBA0IQ,oBA1IR,W;YAA sC,6B
;;;QAC3D,OAAO,W;O;KANX,C;4EASA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,oD;QAKiB,Q;QAAA,0B;
QAAb,OAAa,cAAb,C;UAAa,iC;UACT,WAA Y,WAAI,UAAU,iBAAV,CAAJ,C;;QACHB,OAAO,W;O;KAPX,C;I
Ae4B,4C;MAAA,mB;QAAE,iC;O;K;IAL9B,iC;MAKI,OAAO,qBAaiB,6BAajB,C;K;wEAGX,yB;MAAA,6C;MA
AA,oC;MAAA,gC;MAAA,uC;QAMoB,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UAAM,IAAI,CAA
C,UAAU,oBAAV,CAAL,C;YAAyB,OAAO,K;;QACtD,OAAO,I;O;KAPX,C;IAUA,2B;MAMI,OAAO,ECrwByC,
qBAAU,CDqwBnD,C;K;wEAGX,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,uC;QAMoB,Q;QAAA,0B;QAAh
B,OAAgB,cAAhB,C;UAAgB,oC;UAAM,IAAI,UAAU,oBAAV,CAAJ,C;YAAwB,OAAO,I;;QACrD,OAAO,K;O;
KAPX,C;4EAUA,qB;MAKI,OAAO,gB;K;4EAGX,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,uC;QAKoB,Q;Q
ADhB,YAA Y,C;QACI,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UAAM,IAAI,UAAU,oBAAV,CAAJ,C;YAAw

B,qB;;QAC9C,OAAO,K;O;KANX,C;0EASA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,gD;QAUoB,Q;QADh
B,kBAaKb,O;QACF,0B;QAAhB,OAAGb,cAAhB,C;UAAgB,oC;UAAM,cAAc,UAAU,WAAV,EAAuB,oBAAvB,
C;;QACpC,OAAO,W;O;KAXX,C;wFAcA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,gD;QAYoB,UAA8B,M;
QAF9C,YAAY,C;QACZ,kBAaKb,O;QACF,0B;QAAhB,OAAGb,cAAhB,C;UAAgB,oC;UAAM,cAAc,WAAU,cA
AV,EAAU,sBAAV,WAAmB,WAAhB,EAAgC,oBAAhC,C;;QACpC,OAAO,W;O;KAbX,C;mFAgBA,yB;MAAA,
uD;MAAA,oC;MAAA,gD;QAYoC,Q;QAHhC,YAAY,wB;QACZ,kBAaKb,O;QACIB,OAAO,SAAS,CAAhB,C;U
ACI,cAAc,UAAU,kCAAI,YAAJ,EAAI,oBAAJ,SAAV,EAAwB,WAAxB,C;;QAEIB,OAAO,W;O;KAdX,C;GaiB
A,yB;MAAA,uD;MAAA,oC;MAAA,gD;QAU,I,YAAY,wB;QACZ,kBAaKb,O;QACIB,OAAO,SAAS,CAAhB,C;U
ACI,cAAc,UAAU,KAAV,EAAiB,iCAAI,KAAJ,EAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAhB
X,C;gFamBA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,oC;QAIoB,Q;QAAA,0B;QAAhB,OAAGb,cAAhB,C;
UAAgB,oC;UAAM,OAAO,oBAAP,C;;O;KAJ1B,C;8FAOA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,oC;QA
OiB,UAAa,M;QAD1B,YAAY,C;QACC,0B;QAAb,OAAa,cAAb,C;UAAa,iC;UAAM,QAAO,cAAP,EAAO,sBAAP
,WAAgB,iBAAhB,C;;O;KAPvB,C;IAUA,2B;MAGI,OAAO,uB;K;4EAGX,yB;MAMA,uD;MAAA,oC;MANA,sC;
QAGW,sB;;UAUP,ICz4BgD,qBAAU,CDy4B1D,C;YAAe,qBAAO,I;YAAP,uB;;UACf,cAAc,qBAAK,CAAL,C;U
ACd,gBAAqB,wB;UACrB,IAAI,cAAa,CAAjB,C;YAAoB,qBAAO,O;YAAP,uB;;UACpB,eAdmB,QAcJ,CAAS,oB
AAT,C;UACf,aAAU,CAAV,OAAa,SAAb,M;YACI,QAAQ,qBAAK,CAAL,C;YACR,QAjBe,QAiBP,CAAS,cAAT
,C;YACR,IAAI,2BAAW,CAAX,KAAJ,C;cACI,UAAU,C;cACV,WAAW,C;;;UAGnB,qBAAO,O;;;QAvBP,yB;O;
KAHJ,C;wFAMA,yB;MAAA,uD;MAAA,oC;MAAA,sC;QAOI,ICz4BgD,qBAAU,CDy4B1D,C;UAAe,OAAO,I;Q
ACtB,cAAc,qBAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;
QAC3B,eAAe,SAAS,oBAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,qBAAK,CAAL,C;UACR,QA
AQ,SAAS,cAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O
;O;KApBX,C;4EAuBA,yB;MAAA,sE;MAAA,oC;MAAA,uD;MdzNCA,iB;McyNCA,sC;QAEiB,Q;QAFb,ICt6BgD,
qBAAU,CDs6B1D,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,C
AAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,WdloCG,MAAO,KckoCO,QdloCP,EckoCiB,CdloCj
B,C;;QcooCd,OAAO,Q;O;KAnBX,C;4EAsBA,yB;MAAA,sE;MAAA,oC;MAAA,uD;Md1pCA,iB;Mc0pCA,sC;Q
AeiB,Q;QAFb,IC57BgD,qBAAU,CD47B1D,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;Q
ACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,WdnqCG,MAAO,KcmqC
O,QdnqCP,EcmqCiB,CdnqCjB,C;;QcqqCd,OAAO,Q;O;KAnBX,C;4EAsBA,yB;MAAA,sE;MAAA,oC;MAAA,uD
;MAAA,sC;QAaiB,Q;QAFb,IC9BgD,qBAAU,CDg9B1D,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,iCAAK,CA
AL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,IAAI,2BA
AW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;wFAsBA,yB;MAAA,oC;MAAA,uD;Md3r
CA,iB;Mc2rCA,sC;QAaiB,Q;QAFb,ICt+BgD,qBAAU,CDs+B1D,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,iCAAK,
CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,WdlsCG
,MAAO,KcksCO,QdlsCP,EcksCiB,CdlsCjB,C;;QcosCd,OAAO,Q;O;KAjBX,C;wFAoBA,yB;MAAA,oC;MAAA,u
D;Md1tCA,iB;Mc0tCA,sC;QAaiB,Q;QAFb,IC1/BgD,qBAAU,CD0/B1D,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,i
CAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,
WdjuCG,MAAO,KciuCO,QdjuCP,EciuCiB,CdjuCjB,C;;QcmuCd,OAAO,Q;O;KAjBX,C;wFAoBA,yB;MAAA,oC;
MAAA,uD;MAAA,sC;QAWiB,Q;QAFb,IC5gCgD,qBAAU,CD4gC1D,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,iC
AAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,IA
AI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;oFAoBA,yB;MAAA,sE;MAAA,o
C;MAAA,uD;MAAA,kD;QAaiB,Q;QAFb,ICliCgD,qBAAU,CDkiC1D,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,i
CAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,I
AAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;K
AnBX,C;gGAsBA,yB;MAAA,oC;MAAA,uD;MAAA,kD;QAWiB,Q;QAFb,ICtjCgD,qBAAU,CDsjC1D,C;UAAe,
OAAO,I;QACtB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iC
AAK,CAAL,EAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WA
AW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;IAoBA,iC;MAOiB,Q;MAFb,ICtkCgD,qBAAU,CDskC1D,C;QAAe,OAA
O,I;MACtB,UAAU,qBAAK,CAAL,C;MACG,kC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,qBAAK,CAAL,C;QACR

,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,2C;MAGI,OAAO,4BAAc,UAAAd,C;K;IAGX,iD;MAOiB,Q;MAFb,IC11CgD,qBAAU,CD0IC1D,C;QAAe,OAAO,I;MACTb,UAAU,qBAAK,CAAL,C;MACG,kC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,qBAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,gBAAR,EAAa,cAAb,C AAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,2B;MAGI,OAAO,uB;K;4EAGX,yB;MA MA,uD;MAAA,oC;MANA,sC;QAGW,sB;;UAUP,ICtnCgD,qBAAU,CDsnC1D,C;YAAe,qBAAO,I;YAAP,uB;;UA Cf,cAAc,qBAAK,CAAL,C;UACd,gBAAqB,wB;UACrB,IAAI,cAAa,CAAjB,C;YAAoB,qBAAO,O;YAAP,uB;;UA CpB,eAdmB,QAcJ,CAAS,oBAAT,C;UACf,aAAU,CAAV,OAAa,SAAb,M;YACI,QAAQ,qBAAK,CAAL,C;YAC R,QAjBe,QAiBP,CAAS,cAAT,C;YACR,IAAI,2BAAW,CAAX,KAAJ,C;cACI,UAAU,C;cACV,WAAW,C;;;UAG nB,qBAAO,O;;;QAvBP,yB;O;KAHJ,C;wFAMA,yB;MAAA,uD;MAAA,oC;MAAA,sC;QAOI,ICtnCgD,qBAAU,C DsnC1D,C;UAAe,OAAO,I;QACtB,cAAc,qBAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa, CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,oBAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ, qBAAK,CAAL,C;UACR,QAAQ,SAAS,cAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV, WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;4EAuBA,yB;MAAA,sE;MAAA,oC;MAAA,uD;MdlpCA,iB;MckpCA ,sC;QAEiB,Q;QAFb,ICnpCgD,qBAAU,CDmpC1D,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,iCAAK,CAAL,EA AT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,Wd3pCG,MAAO, Kc2pCO,Qd3pCP,Ec2pCiB,Cd3pCjB,C;;;Qc6pCd,OAAO,Q;O;KAnBX,C;4EAsBA,yB;MAAA,sE;MAAA,oC;MA AAA,uD;MdnrCA,iB;McmrCA,sC;QAEiB,Q;QAFb,ICzqCgD,qBAAU,CDyqC1D,C;UAAe,MAAM,6B;QACrB,eAA e,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C ;UACR,Wd5rCG,MAAO,Kc4rCO,Qd5rCP,Ec4rCiB,Cd5rCjB,C;;;Qc8rCd,OAAO,Q;O;KAnBX,C;4EAsBA,yB;MA AAA,sE;MAAA,oC;MAAA,uD;MAAA,sC;QAaiB,Q;QAFb,IC7rCgD,qBAAU,CD6rC1D,C;UAAe,MAAM,6B;QAC rB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL, EAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;wFAsBA,yB; MAAA,oC;MAAA,uD;MdptCA,iB;McotCA,sC;QAaiB,Q;QAFb,ICntCgD,qBAAU,CDmtC1D,C;UAAe,OAAO,I; QACtB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CA AL,EAAT,C;UACR,Wd3tCG,MAAO,Kc2tCO,Qd3tCP,Ec2tCiB,Cd3tCjB,C;;;Qc6tCd,OAAO,Q;O;KAjBX,C;wFAo BA,yB;MAAA,oC;MAAA,uD;MdnvCA,iB;McmvCA,sC;QAaiB,Q;QAFb,ICvuCgD,qBAAU,CDuuC1D,C;UAAe, OAAO,I;QACtB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iC AAK,CAAL,EAAT,C;UACR,Wd1vCG,MAAO,Kc0vCO,Qd1vCP,Ec0vCiB,Cd1vCjB,C;;;Qc4vCd,OAAO,Q;O;KAj BX,C;wFAoBA,yB;MAAA,oC;MAAA,uD;MAAA,sC;QAWiB,Q;QAFb,ICzvCgD,qBAAU,CDyvC1D,C;UAAe,O AAO,I;QACtB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iCA AK,CAAL,EAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;oF AoBA,yB;MAAA,sE;MAAA,oC;MAAA,uD;MAAA,kD;QAaiB,Q;QAFb,IC/wCgD,qBAAU,CD+wC1D,C;UAAe, MAAM,6B;QACrB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS ,iCAAK,CAAL,EAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAiC,C;YACI,W AAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;gGAsBA,yB;MAAA,oC;MAAA,uD;MAAA,kD;QAWiB,Q;QAFb,ICny CgD,qBAAU,CDmyC1D,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAA U,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,C AAX,GAakC,CAAiC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;IAoBA,iC;MAOiB,Q;MAFb,ICnzCgD ,qBAAU,CDmzC1D,C;QAAe,OAAO,I;MACTb,UAAU,qBAAK,CAAL,C;MACG,kC;MAAb,aAAU,CAAV,iB;QA CI,QAAQ,qBAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,2C;M AGI,OAAO,4BAAc,UAAAd,C;K;IAGX,iD;MAOiB,Q;MAFb,ICv0CgD,qBAAU,CDu0C1D,C;QAAe,OAAO,I;MA CtB,UAAU,qBAAK,CAAL,C;MACG,kC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,qBAAK,CAAL,C;QACR,IAAI, UAAW,SAAQ,gBAAR,EAAa,cAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,4 B;MAMI,OCt1CgD,qBAAU,C;K;0EDy1C9D,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,uC;QAMoB,Q;QAA A,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UAAM,IAAI,UAAU,oBAAV,CAAJ,C;YAAwB,OAAO,K;;QACrD, OAAO,I;O;KAPX,C;8EAU,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,oC;QAKmC,Q;QAAA,0B;QAAhB,O AAgB,cAAhB,C;UAAgB,oC;UAAM,OAAO,oBAAP,C;;QAARc,gB;O;KALJ,C;4FAQA,yB;MAAA,6B;MAAA,sC ;MA/fa,6C;MAAA,oC;MAAA,gC;MA+fa,2BAQiB,yB;QAvGBjB,6C;QAAA,oC;QAAA,gC;eAugBiB,0B;UAAA,

4B;YAAE,aAAe,c;YAhgBjB,gB;YADb,YAAy,C;YACC,0B;YAAb,OAAa,cAAb,C;cAAa,iC;cAAM,QAAO,cAAP ,EAAO,sBAAP,WAAgB,iBAAhB,C;;YAggBmB,W;W;S;OAAzB,C;MARjB,oC;QAxfiB,gB;QADb,YAAy,C;QA CC,0B;QAAb,OAAa,cAAb,C;UAAa,iC;UAAM,QAAO,cAAP,EAAO,sBAAP,WAAgB,iBAAhB,C;;QAggBnB,gB; O;KARJ,C;8EAWA,yB;MAAA,4F;MAAA,uD;MAAA,oC;MAAA,gC;MAAA,uC;QAgBqB,Q;QAHjB,ICn4CgD,q BAAU,CDm4C1D,C;UACI,MAAM,mCAA8B,uCAA9B,C;QACV,kBAakB,qBAAK,CAAL,C;QACD,+B;QAAjB ,iBAAc,CAAd,yB;UACI,cAAc,oBAAU,wBAAV,EAAuB,iCAAK,KAAL,EAAvB,E;;QAEIB,OAAO,W;O;KAnBX ,C;4FAsBA,yB;MAAA,4F;MAAA,uD;MAAA,oC;MAAA,gC;MAAA,uC;QAgBqB,Q;QAHjB,ICz5CgD,qBAAU, CDy5C1D,C;UACI,MAAM,mCAA8B,uCAA9B,C;QACV,kBAakB,qBAAK,CAAL,C;QACD,+B;QAAjB,iBAAc, CAAd,yB;UACI,cAAc,oBAAU,KAAV,EAAiB,wBAAjB,EAA8B,iCAAK,KAAL,EAA9B,E;;QAEIB,OAAO,W;O; KAnBX,C;wGAsBA,yB;MAAA,uD;MAAA,oC;MAAA,gC;MAAA,uC;QAgBqB,Q;QAHjB,IC/6CgD,qBAAU,CD +6C1D,C;UACI,OAAO,I;QACX,kBAakB,qBAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,o BAAU,KAAV,EAAiB,wBAAjB,EAA8B,iCAAK,KAAL,EAA9B,E;;QAEIB,OAAO,W;O;KAnBX,C;0FAsBA,yB; MAAA,uD;MAAA,oC;MAAA,gC;MAAA,uC;QAIbqB,Q;QAHjB,ICt8CgD,qBAAU,CDs8C1D,C;UACI,OAAO,I; QACX,kBAakB,qBAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,oBAAU,wBAAV,EAAuB,i CAAK,KAAL,EAAvB,E;;QAEIB,OAAO,W;O;KApBX,C;uFAuBA,yB;MAAA,uD;MAAA,4F;MAAA,oC;MAAA, gC;MAAA,uC;QAE0B,UAEU,M;QAJhC,YAAy,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,uCA A9B,C;QACrB,kBAakB,sBAAI,YAAJ,EAAI,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,oBAAU,k CAAI,cAAJ,EAAI,sBAAJ,WAAV,EAAwB,wBAAxB,E;;QAEIB,OAAO,W;O;KAnBX,C;qGAsBA,yB;MAAA,uD ;MAAA,4F;MAAA,oC;MAAA,gC;MAAA,uC;QAE0B,Q;QAFtB,YAAy,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe, MAAM,mCAA8B,uCAA9B,C;QACrB,kBAakB,sBAAI,YAAJ,EAAI,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C; UACI,cAAc,oBAAU,KAAV,EAAiB,iCAAi,KAAJ,EAAjB,EAA6B,wBAA7B,E;UACd,qB;;QAEJ,OAAO,W;O;K ApBX,C;iHAuBA,yB;MAAA,uD;MAAA,oC;MAAA,gC;MAAA,uC;QAE0B,Q;QAFtB,YAAy,wB;QACZ,IAAI,Q AAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAakB,sBAAI,YAAJ,EAAI,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C; UACI,cAAc,oBAAU,KAAV,EAAiB,iCAAi,KAAJ,EAAjB,EAA6B,wBAA7B,E;UACd,qB;;QAEJ,OAAO,W;O;K ApBX,C;mGAuBA,yB;MAAA,uD;MAAA,oC;MAAA,gC;MAAA,uC;QAgB0B,UAEU,M;QAJhC,YAAy,wB;QA CZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAakB,sBAAI,YAAJ,EAAI,oBAAJ,Q;QACIB,OAAO,SAAS ,CAAhB,C;UACI,cAAc,oBAAU,kCAAi,cAAJ,EAAI,sBAAJ,WAAV,EAAwB,wBAAxB,E;;QAEIB,OAAO,W;O; KApBX,C;wFAuBA,yB;MAAA,gD;MAAA,gE;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,gD;QAgBoB,Q;QAHh B,ICvjDgD,qBAAU,CDujD1D,C;UAAe,OAAO,OAAO,OAAP,C;QACgB,kBAAzB,eAAa,mBAAS,CAAT,IAAb,C ;QAAiC,8B;QAA9C,af5wDO,W;Qe6wDP,kBAakB,O;QACF,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,c AAc,UAAU,WAAV,EAAuB,oBAAvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KApBX,C;sGAuB A,yB;MAAA,gD;MAAA,gE;MAAA,mD;MAAA,oC;MAAA,gD;QAIbKb,gC;QAHd,IC/kDgD,qBAAU,CD+kD1D ,C;UAAe,OAAO,OAAO,OAAP,C;QACgB,kBAAzB,eAAa,mBAAS,CAAT,IAAb,C;QAAiC,8B;QAA9C,afpyDO, W;QeqyDP,kBAakB,O;QACJ,6B;QAAA,mB;QAAA,kB;QAAA,kB;QAAAd,0D;UACI,cAAc,UAAU,KAAV,EAAi B,WAAjB,EAA8B,iCAAK,KAAL,EAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KArBX,C;4FA wBA,yB;MAAA,qD;MAAA,gE;MAAA,oC;MAAA,gC;MAAA,uC;QAgB0B,Q;QAHtB,ICtmDgD,qBAAU,CDsm D1D,C;UAAe,OAAO,W;QACtB,sBAakB,qBAAK,CAAL,CAAIB,C;QACqC,kBAAxB,eAAgB,gBAAhB,C;QAA gC,sBAAI,0BAAJ,C;QAA7C,af5zDO,W;Qe6zDe,uB;QAAtB,iBAAc,CAAd,wB;UACI,gBAAc,oBAAU,0BAAV,E AAuB,iCAAK,KAAL,EAAvB,E;UACd,MAAO,WAAI,0BAAJ,C;;QAEX,OAAO,M;O;KApBX,C;0GAuBA,yB;M AAA,qD;MAAA,gE;MAAA,oC;MAAA,gC;MAAA,uC;QAIb0B,Q;QAHtB,IC9nDgD,qBAAU,CD8nD1D,C;UAA e,OAAO,W;QACtB,sBAakB,qBAAK,CAAL,CAAIB,C;QACqC,kBAAxB,eAAgB,gBAAhB,C;QAAgC,sBAAI,0B AAJ,C;QAA7C,afp1DO,W;Qeq1De,uB;QAAtB,iBAAc,CAAd,wB;UACI,gBAAc,oBAAU,KAAV,EAAiB,0BAAj B,EAA8B,iCAAK,KAAL,EAA9B,E;UACd,MAAO,WAAI,0BAAJ,C;;QAEX,OAAO,M;O;KArBX,C;0EAwBA,y B;MA9FA,gD;MAAA,gE;MAAA,6C;MAAA,oC;MAAA,gC;MA9FA,gD;QAcW,sB;;UA5FS,Q;UAHhB,ICvjDgD, qBAAU,CDujD1D,C;YAAe,qBAAO,OA+FH,OA/FG,C;YAAP,uB;;UACuB,kBAAzB,eAAa,mBAAS,CAAT,IAAb ,C;UAAiC,sBA8F3B,OA9F2B,C;UAA9C,af5wDO,W;Ue6wDP,kBA6FmB,O;UA5FH,0B;UAAhB,OAAgB,cAAh B,C;YAAgB,oC;YACZ,cA2FwB,SA3FV,CAAU,WAAV,EAAuB,oBAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UA EX,qBAAO,M;;QAwFP,yB;O;KAdJ,C;wFAiBA,yB;MAxFA,gD;MAAA,gE;MAAA,mD;MAAA,oC;MAwFA,gD;

QAeW,6B;;UAiFO,gC;UAHd,IC/kDgD,qBAAU,CD+kD1D,C;YAAe,4BAAO,OAYFI,OAzFJ,C;YAAP,8B;;UACu
B,kBAAzB,eAAa,mBAAS,CAAT,IAAb,C;UAAiC,sBAwFpB,OAxFoB,C;UAA9C,afpyDO,W;UeqyDP,kBAuF0B,
O;UAiFZ,6B;UAAA,mB;UAAA,kB;UAAA,kB;UAAAd,0D;YACI,cAqF+B,SArFjB,CAAU,KAAV,EAAiB,WAAjB
,EAA8B,iCAAk,KAAL,EAA9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;;QAKFP,gC;O;KAFJ,C;4E
AkBA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,sC;QAOoB,Q;QADhB,UAAe,C;QACC,0B;QAAhB,OAAgB
,cAAhB,C;UAAgB,oC;UACZ,YAAO,SAAS,oBAAT,CAAP,I;;QAEJ,OAAO,G;O;KAVX,C;wFAaA,yB;MAAA,6
C;MAAA,oC;MAAA,gC;MAAA,sC;QAOoB,Q;QADhB,UAAkB,G;QACF,0B;QAAhB,OAAgB,cAAhB,C;UAAg
B,oC;UACZ,OAAO,SAAS,oBAAT,C;;QAEX,OAAO,G;O;KAVX,C;4EAaA,yB;MAAA,6C;MAAA,oC;MAAA,g
C;MAAA,sC;QAUoB,Q;QADhB,UAAoB,C;QACJ,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,OAAO,SA
AS,oBAAT,C;;QAEX,OAAO,G;O;KAbX,C;4EAgBA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,sC;QAUoB,
Q;QADhB,UAAe,C;QACC,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,YAAO,SAAS,oBAAT,CAAP,I;;Q
AEJ,OAAO,G;O;KAbX,C;4EAgBA,yB;MAAA,SASoB,gB;MATpB,6C;MAAA,oC;MAAA,gC;MAAA,sC;QAUo
B,Q;QADhB,Y;QACgB,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,cAAO,SAAS,oBAAT,CAAP,C;;QAEJ
,OAAO,G;O;KAbX,C;4EAgBA,yB;MAAA,6C;MAAA,oC;MAAA,gC;M7BppDA,6B;M6BopDA,sC;QAWoB,Q;Q
ADhB,U7BppDmC,c6BopDnB,C7BppDmB,C;Q6BqpDnB,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,M7
Bx9DiD,c6Bw9DjD,G7Bx9D2D,KAAK,G6Bw9DzD,SAAS,oBAAT,C7Bx9DoE,KAAK,IAAf,C;;Q6B09DrD,OA
AO,G;O;KAdX,C;4EAiBA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MblqDA,+B;MakqDA,sC;QAWoB,Q;QADhB,
UbjqDqC,eAAW,oBaiqD/B,CbjqD+B,CAAX,C;QakqDrB,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,Mbt
+DmD,eas+DnD,Gbt+D8D,KAAK,Kas+D5D,SAAS,oBAAT,Cbt+DuE,KAAK,CAAhB,C;;Qaw+DvD,OAAO,G;O
;KAdX,C;IAiBA,oC;MAWI,OAAO,sBAAS,IAAT,EAAe,IAAf,EAAc,IAAtC,C;K;IAGX,+C;MAGBI,OAAO,sBA
AS,IAAT,EAAe,IAAf,EAAc,IAAtC,EAAwD,SAAXD,C;K;IACsB,oC;MAAE,OAAA,EAAG,W;K;IAXtC,0C;MA
WI,OAAO,6BAAgB,IAAhB,EAAhB,sBAAtB,C;K;IAGX,uD;MAGBI,OAAO,8BAAiB,IAAjB,EAAuB,IAAvB,EA
A8C,IAA9C,EAAG,E,SAAhE,C;K;oFAGX,yB;MAAA,yD;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,6B;MAAA,u
C;QAUoB,Q;QAFhB,YAAY,oB;QACZ,aAAa,oB;QACG,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,IAAI
,UAAU,oBAAV,CAAJ,C;YACI,KAAM,gBAAO,OAAP,C;;YAEN,MAAO,gBAAO,OAAP,C;;;QAGf,OAAO,cAA
K,KAAL,EAAy,MAAZ,C;O;KAjBX,C;oFAoBA,yB;MAAA,yD;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,6B;M
AAA,uC;QAUoB,Q;QAFhB,YAAY,oB;QACZ,aAAa,oB;QACG,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UAC
Z,IAAI,UAAU,oBAAV,CAAJ,C;YACI,KAAM,gBAAO,OAAP,C;;YAEN,MAAO,gBAAO,OAAP,C;;;QAGf,OAA
O,cAAK,KAAM,WAAW,EAAB,MAAO,WAA9B,C;O;KAjBX,C;IAqCgD,6B;MAAE,OAAA,EAAG,W;K;IAjBr
D,2D;MAGB4C,oB;QAAA,OAAY,C;MAAG,8B;QAAA,iBAA0B,K;MACjF,OAAO,sBAAS,IAAT,EAAe,IAAf,E
AAqB,cAArB,EAAqC,eAArC,C;K;IAGX,sE;MAkBgD,oB;QAAA,OAAY,C;MAAG,8B;QAAA,iBAA0B,K;MAQ
hE,Q;MAPrB,oBAAoB,IAApB,EAA0B,IAA1B,C;MACA,eAAe,SAAK,O;MACpB,qBAAqB,YAAW,IAAX,SAAS
B,WAAW,IAAX,KAAmB,CAAvB,GAA0B,CAA1B,GAAiC,CAAnD,K;MACrB,aAAa,iBAAa,cAAAb,C;MACb,Y
AAy,C;MACZ,OAAgB,CAAT,qBAAiB,QAAxB,C;QACI,UAAU,QAAQ,IAAR,I;QACO,IAAI,MAAM,CAAN,IA
AW,MAAM,QAArB,C;UAAiC,IAAI,cAAJ,C;YAAoB,e;;YAAc,K;;UAAa,U;QAAjG,qB;QACA,MAAO,WAAI,U
AAU,8BAAy,KAAZ,EAAMB,UAAAnB,CAAV,CAAJ,C;QACP,gBAAS,IAAT,I;;MAEJ,OAAO,M;K;IAoB6C,qC;
MAAE,OAAA,EAAG,W;K;IAjB7D,iE;MAGBoD,oB;QAAA,OAAY,C;MAAG,8B;QAAA,iBAA0B,K;MACzF,O
AAO,8BAAiB,IAAjB,EAAuB,IAAvB,EAA6B,cAA7B,EAA6C,uBAA7C,C;K;IAwByB,2F;MAAA,wB;QAC5B,U
AAU,QAAQ,YAAR,I;QACV,iBAAqB,MAAM,CAAN,IAAW,MAAM,4BAArB,GAA6B,4BAA7B,GAAyC,G;QA
D1D,OAEA,kBAAU,0CAAy,KAAZ,EAAMB,UAAAnB,CAAV,C;O;K;IAxBR,gF;MAkBWd,sB;QAAA,SAAY,C;
MAAG,8B;QAAA,iBAA0B,K;MAC7F,oBAAoB,IAApB,EAA0B,MAA1B,C;MACA,cAAc,KAAK,cAAJ,GAAoB,
yBAApB,GAAiC,WAAQ,mBAAS,IAAT,GAAgB,CAAhB,IAAR,CAAIc,EAAkE,MAAIE,C;MACd,OAA4B,OA
Ab,aAAR,OAAQ,CAAA,EAAI,qDAAJ,C;K;IAOhC,kC;MAkBI,ad3hEO,MAAO,Kc2hEU,gBd3hEV,EcghEH,KA
W2B,Od3hExB,C;Mc4hEd,WAAW,iBAAa,MAAb,C;MACX,aAAU,CAAV,MAAkB,MAAIB,M;QACI,IAAK,WA
dqB,GAcP,iCAAk,CAAL,EAdO,EAcE,YAdR,KAcqB,YAAM,CAAN,EAdF,CACrB,C;;MAdT,OAgBO,I;K;wEA
bX,yB;MAAA,gE;MAAA,oC;MdzHEA,iB;McyHEA,8C;QAQI,ad3hEO,MAAO,Kc2hEK,SAAK,Od3hEV,Ec2hEk
B,KAAM,Od3hExB,C;Qc4hEd,WAAW,eAAa,MAAb,C;QACX,aAAU,CAAV,MAAkB,MAAIB,M;UACI,IAAK,
WAAI,UAAU,iCAAk,CAAL,EAAV,EAAMB,6BAAM,CAAN,EAAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;IAG

BA,kC;MASW,sB;;QAaP,WAAW,mBAAS,CAAT,I;QACX,IAAI,OAAO,CAAX,C;UAAc,qBAAO,W;UAAP,uB;;
QACd,aAAa,iBAAa,IAAb,C;QACb,iBAAc,CAAd,UAAsB,IAAtB,U;UACI,MAAO,WajBkB,GaiBJ,iCAAK,KA
AL,EAjBI,EAiBS,iCAAK,QAAQ,CAAR,IAAL,EAjBT,CAiBIB,C;;QAEX,qBAAO,M;;;MANBP,yB;K;uFAGJ,yB;
MAAA,qD;MAAA,gE;MAAA,oC;MAAA,uC;QAUI,WAAW,mBAAS,CAAT,I;QACX,IAAI,OAAO,CAAX,C;UA
Ac,OAAO,W;QACrB,aAAa,eAAa,IAAb,C;QACb,iBAAc,CAAd,UAAsB,IAAtB,U;UACI,MAAO,WAAI,UAAU,i
CAAK,KAAL,EAAV,EAAuB,iCAAK,QAAQ,CAAR,IAAL,EAAvB,CAAJ,C;;QAEX,OAAO,M;O;KahBX,C;IA
wBoB,8C;MAAA,mB;QAAE,OAAK,WAAI,eAAK,C;O;K;IAL3B,kC;MAIQ,wC;MAAA,S;QAAkB,OCniE0B,qB
AAU,C;;MDmiE1D,S;QAAiC,OAAO,W;MACxC,oCAAgB,8BAAhB,C;K;IAQgB,8C;MAAA,mB;QAAE,OAAK,
WAAI,eAAK,C;O;K;IAL3B,kC;MAIQ,wC;MAAA,S;QAAkB,OC3iE0B,qBAAU,C;;MD2iE1D,S;QAAiC,OAAO,
e;MACxC,oCAAgB,8BAAhB,C;K;IEpwEkC,yC;MAAA,wB;QAAW,OAAA,aAAK,KAAL,ChCsLV,K;O;K;liCiL
H,wC;MAAA,wB;QAAW,OAAA,aAAK,KAAL,ChC8NV,K;O;K;liC9NC,yC;MAAA,wB;QAAW,OAAA,aAAK,
KAAL,CjByOV,K;O;K;IkBzOC,0C;MAAA,wB;QAAW,OAAA,aAAK,KAAL,CjCiMV,K;O;K;4FkC5PzC,qB;MA
UI,OAAO,sBAAI,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAI,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAI,CA
AJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAI,CAAJ,C;K;4FAGX,qB;MAUI,OAAO,sBAAI,CAAJ,C;K;6FAGX,qB;M
AUI,OAAO,sBAAI,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAI,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAI,C
AAJ,C;K;4FAGX,qB;MAUI,OAAO,sBAAI,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAI,CAAJ,C;K;6FAGX,qB;
MAUI,OAAO,sBAAI,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAI,CAAJ,C;K;4FAGX,qB;MAUI,OAAO,sBAAI,
CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAI,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAI,CAAJ,C;K;6FAGX,q
B;MAUI,OAAO,sBAAI,CAAJ,C;K;4FAGX,qB;MAUI,OAAO,sBAAI,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBA
AI,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAI,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAI,CAAJ,C;K;uGAuC
X,yB;MA8gHI,8D;MA9gHJ,iD;QASe,oBAAS,C;QAAT,S;UAAc,gBAqgHT,cAAR,iBAAQ,C;;QArgHhB,OAAO,
OAAcS,sBAAI,KAJ,CAATC,GAAsD,aAAa,KAAb,C;O;KATjE,C;uGAYA,yB;MA0gHI,8D;MA1gHJ,iD;QASe,o
BAAS,C;QAAT,S;UAAc,gBAigHT,cAAR,iBAAQ,C;;QAjgHhB,OAAO,OAAcS,sBAAI,KAJ,CAATC,GAAsD,a
AAa,KAAb,C;O;KATjE,C;uGAYA,yB;MAsgHI,8D;MATgHJ,iD;QASe,oBAAS,C;QAAT,S;UAAc,gBA6/GT,cAA
R,iBAAQ,C;;QA7/GhB,OAAO,OAAcS,sBAAI,KAJ,CAATC,GAAsD,aAAa,KAAb,C;O;KATjE,C;uGAYA,yB;M
AkgHI,8D;MA1gHJ,iD;QASe,oBAAS,C;QAAT,S;UAAc,gBAy/GT,cAAR,iBAAQ,C;;QAz/GhB,OAAO,OAAcS,s
BAAI,KAJ,CAATC,GAAsD,aAAa,KAAb,C;O;KATjE,C;uGAYA,yB;MAAA,sD;MAAA,mC;QASI,OAAy,UAA
L,SAAK,EAAU,KAAY,C;O;KAThB,C;uGAYA,yB;MAAA,sD;MAAA,mC;QASI,OAAy,UAAAL,SAAK,EAAU,KAAY,C;O;KAThB
,C;uGAYA,yB;MAAA,sD;MAAA,mC;QASI,OAAy,UAAAL,SAAK,EAAU,KAAY,C;O;KAThB,C;iFAYA,gC;MA
SW,sB;;QA8NS,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UAAM,IA9NH,SA8NO,CAAU,OAAV,CA
AJ,C;YAAwB,qBAAO,O;YAAP,uB;;QAC9C,qBAAO,I;;MA/NP,yB;K;iFAGJ,gC;MASW,sB;;QA6NS,Q;QAAA,
2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UAAM,IA7NH,SA6NO,CAAU,OAAV,CAAJ,C;YAAwB,qBAAO,O;
YAAP,uB;;QAC9C,qBAAO,I;;MA9NP,yB;K;iFAGJ,gC;MASW,sB;;QA4NS,Q;QAAA,2B;QAAhB,OAAGB,cAA
hB,C;UAAgB,yB;UAAM,IA5NH,SA4NO,CAAU,OAAV,CAAJ,C;YAAwB,qBAAO,O;YAAP,uB;;QAC9C,qBAA
AO,I;;MA7NP,yB;K;iFAGJ,gC;MASW,sB;;QA2NS,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UAA
M,IA3NH,SA2NO,CAAU,OAAV,CAAJ,C;YAAwB,qBAAO,O;YAAP,uB;;QAC9C,qBAAO,I;;MA5NP,yB;K;yF
AGJ,yB;MA4nBA,+C;MAkuFI,0D;MA91GJ,uC;QASW,qB;;UA4nBO,Q;UAAA,OAAa,SAyFX,YAAR,iBAAQ,C
AztFW,CAAb,W;UAAAd,OAAc,cAAAd,C;YAAc,uB;YACV,cAAc,sBAAK,KAAL,C;YACd,IA9nBc,SA8nBV,CAA
U,OAAV,CAAJ,C;cAAwB,oBAAO,O;cAAP,sB;;UAE5B,oBAAO,I;;QAhBP,wB;O;KATJ,C;yFAYA,yB;MAgo
BA,+C;MA0tFI,0D;MA11GJ,uC;QASW,qB;;UAgoBO,Q;UAAA,OAAa,SAitFX,YAAR,iBAAQ,CAjFW,CAAb,
W;UAAAd,OAAc,cAAAd,C;YAAc,uB;YACV,cAAc,sBAAK,KAAL,C;YACd,IAloBc,SAkoBV,CAAU,OAAV,CAAJ
,C;cAAwB,oBAAO,O;cAAP,sB;;UAE5B,oBAAO,I;;QApoBP,wB;O;KATJ,C;yFAYA,yB;MAooBA,+C;MAktFI,
0D;MAT1GJ,uC;QASW,qB;;UAooBO,Q;UAAA,OAAa,SAysFX,YAAR,iBAAQ,CAzsFW,CAAb,W;UAAAd,OAAc,
cAAAd,C;YAAc,uB;YACV,cAAc,sBAAK,KAAL,C;YACd,IAtoBc,SAsoBV,CAAU,OAAV,CAAJ,C;cAAwB,oBA
AO,O;cAAP,sB;;UAE5B,oBAAO,I;;QAxoBP,wB;O;KATJ,C;yFAYA,yB;MAwoBA,+C;MA0sFI,0D;MA11GJ,uC
;QASW,qB;;UAwoBO,Q;UAAA,OAAa,SAisFX,YAAR,iBAAQ,CAjsFW,CAAb,W;UAAAd,OAAc,cAAAd,C;YAAc,
uB;YACV,cAAc,sBAAK,KAAL,C;YACd,IA1oBc,SA0oBV,CAAU,OAAV,CAAJ,C;cAAwB,oBAAO,O;cAAP,sB

;;;UAE5B,oBAAO,I;;;QA5oBP,wB;O;KATJ,C;mFAYA,yB;MAAA,8C;MnCPHA,6B;MmCoHA,4B;QAQI,OnCIH
mC,cmCkHpB,MAAR,iBAAQ,CnCIHoB,C;O;KmC0GvC,C;mFAWA,yB;MAAA,8C;MnBhHA,+B;MmBgHA,4B;
QAQI,OnB9GsC,emB8GvB,MAAR,iBAAQ,CnB9GuB,C;O;KmBsG1C,C;mFAWA,yB;MAAA,8C;MpCxLA,+B;
MoCwLA,4B;QAQI,OpCtLsC,eoCsLvB,MAAR,iBAAQ,CpCtLuB,C;O;KoC8K1C,C;mFAWA,yB;MAAA,8C;MI
CtLA,iC;MkCsLA,4B;QAQI,OICpLyC,gBkCoL1B,MAAR,iBAAQ,CICpL0B,C;O;KkC4K7C,C;mFAWA,yB;MA
AA,iE;MAAA,uC;QAQoB,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,UAAU,OAAV,CA
AJ,C;YAAwB,OAAO,O;;QACrD,MAAM,gCAAuB,mDAAvB,C;O;KATV,C;mFAYA,yB;MAAA,iE;MAAA,uC;
QAQoB,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OA
AO,O;;QACrD,MAAM,gCAAuB,mDAAvB,C;O;KATV,C;mFAYA,yB;MAAA,iE;MAAA,uC;QAQoB,Q;QAAA,
2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QACrD,MA
AM,gCAAuB,mDAAvB,C;O;KATV,C;mFAYA,yB;MAAA,iE;MAAA,uC;QAQoB,Q;QAAA,2B;QAAhB,OAAgB
,cAAhB,C;UAAgB,yB;UAAM,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QACrD,MAAM,gCAAuB,mDA
AvB,C;O;KATV,C;IAYA,mC;MAMI,OAAW,mBAAJ,GAAe,IAAf,GAAyB,sBAAK,CAAL,C;K;IAGpC,mC;MA
MI,OAAW,mBAAJ,GAAe,IAAf,GAAyB,sBAAK,CAAL,C;K;IAGpC,mC;MAMI,OAAW,mBAAJ,GAAe,IAAf,G
AAyB,sBAAK,CAAL,C;K;IAGpC,mC;MAMI,OAAW,mBAAJ,GAAe,IAAf,GAAyB,sBAAK,CAAL,C;K;+FAGp
C,gC;MAOoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAA
wB,OAAO,O;;MACrD,OAAO,I;K;+FAGX,gC;MAOoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;Q
AAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,O;;MACrD,OAAO,I;K;+FAGX,gC;MAOoB,Q;MAAA,2B;
MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,O;;MACrD,OAA
O,I;K;+FAGX,gC;MAOoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QAAM,IAAI,UAAU,OAAV,C
AAJ,C;UAAwB,OAAO,O;;MACrD,OAAO,I;K;2FAGX,yB;MAkqGI,8D;MAIqGJ,iD;QAOe,oBAAS,C;QAAT,S;U
AAc,gBA2pGT,cAAR,iBAAQ,C;;QA3pGhB,OAAO,OAAcS,sBAAI,KAAJ,CAAtC,GAAsD,aAAa,KAAb,C;O;K
APjE,C;2FAUA,yB;MAGqGI,8D;MAhGJ,iD;QAOe,oBAAS,C;QAAT,S;UAAc,gBAypGT,cAAR,iBAAQ,C;;QAz
pGhB,OAAO,OAAcS,sBAAI,KAAJ,CAAtC,GAAsD,aAAa,KAAb,C;O;KAPjE,C;2FAUA,yB;MA8pGI,8D;MA9p
GJ,iD;QAOe,oBAAS,C;QAAT,S;UAAc,gBAupGT,cAAR,iBAAQ,C;;QAvpGhB,OAAO,OAAcS,sBAAI,KAAJ,C
AAtC,GAAsD,aAAa,KAAb,C;O;KAPjE,C;2FAUA,yB;MA4pGI,8D;MA5pGJ,iD;QAOe,oBAAS,C;QAAT,S;UAA
c,gBAqpGT,cAAR,iBAAQ,C;;QArpGhB,OAAO,OAAcS,sBAAI,KAAJ,CAAtC,GAAsD,aAAa,KAAb,C;O;KAPjE
,C;IAUA,wC;MAQe,oBAAS,C;MAAT,S;QAAC,gBAknGT,gBAAR,iBAAQ,C;;MAlnGhB,OAAO,OAAcS,sBAAI,
KAAJ,CAAtC,GAAsD,I;K;IAGjE,wC;MAQe,oBAAS,C;MAAT,S;QAAC,gBA+mGT,gBAAR,iBAAQ,C;;MA/mG
hB,OAAO,OAAcS,sBAAI,KAAJ,CAAtC,GAAsD,I;K;IAGjE,wC;MAQe,oBAAS,C;MAAT,S;QAAC,gBA4mGT,g
BAAR,iBAAQ,C;;MA5mGhB,OAAO,OAAcS,sBAAI,KAAJ,CAAtC,GAAsD,I;K;IAGjE,wC;MAQe,oBAAS,C;M
AAT,S;QAAC,gBAymGT,gBAAR,iBAAQ,C;;MAzmGhB,OAAO,OAAcS,sBAAI,KAAJ,CAAtC,GAAsD,I;K;uFA
GjE,yB;MAAA,kD;MAAA,qC;QAOI,OAAe,QAAR,iBAAQ,EAAQ,OnCtdU,KmCsdIB,C;O;KAPnB,C;uFAUA,yB
;MAAA,kD;MAAA,qC;QAOI,OAAe,QAAR,iBAAQ,EAAQ,OpClhBY,KoCkhBpB,C;O;KAPnB,C;uFAUA,yB;MAAA,k
D;MAAA,qC;QAOI,OAAe,QAAR,iBAAQ,EAAQ,OICjhBc,KkCihBtB,C;O;KAPnB,C;IGAUA,yB;MAAA,sC;Mn
C5ZA,6B;MmC4ZA,0BAOgC,yB;QnCnahC,6B;emCmagC,6B;UAAA,qB;YAAE,yBnCzZK,cmCyZK,EnCzZL,C
mCyZL,C;W;S;OAAF,C;MAPhC,uC;QAOMb,kBAAR,iB;QAAQ,uB;;UtC40Bf,0D;YACI,IsC70B0B,UnCzZK,cH
suCjB,YAAK,KAAL,CGtuCiB,CmCyZL,CtC60B1B,C;cACI,sBAAO,K;cAAP,wB;;;UAGR,sBAAO,E;;;QsCj1BP,
0B;O;KAPJ,C;IGAUA,yB;MAAA,sC;MnBvZA,+B;MmBuZA,0BAOgC,yB;QnB9ZhC,+B;emB8ZgC,6B;UAAA,q
B;YAAE,yBnBpZQ,emBoZE,EnBpZF,CmBoZR,C;W;S;OAAF,C;MAPhC,uC;QAOMb,kBAAR,iB;QAAQ,uB;;Ut
C80Bf,0D;YACI,IsC/0B0B,UnBpZQ,enBmuCpB,YAAK,KAAL,CmBnuCoB,CmBoZR,CtC+0B1B,C;cACI,sBAA
O,K;cAAP,wB;;;UAGR,sBAAO,E;;;QsCn1BP,0B;O;KAPJ,C;IGAUA,yB;MAAA,sC;MpC9dA,+B;MoC8dA,0BA
OgC,yB;QpCrehC,+B;eoCqegC,6B;UAAA,qB;YAAE,yBpC3dQ,eoC2dE,EpC3dF,CoC2dR,C;W;S;OAAF,C;MAP
hC,uC;QAOMb,kBAAR,iB;QAAQ,uB;;UtCgyBf,0D;YACI,IsCjyB0B,UpC3dQ,eF4vCpB,YAAK,KAAL,CE5vCo
B,CoC2dR,CtCiyB1B,C;cACI,sBAAO,K;cAAP,wB;;;UAGR,sBAAO,E;;;QsCryBP,0B;O;KAPJ,C;IGAUA,yB;MA
AA,sC;MIC3dA,iC;MkC2dA,0BAOgC,yB;QIClehC,iC;ekCkegC,6B;UAAA,qB;YAAE,yBICxdW,gBkCwdD,EICx
dC,CkCwdX,C;W;S;OAAF,C;MAPhC,uC;QAOMb,kBAAR,iB;QAAQ,uB;;UtCkyBf,0D;YACI,IsCnyB0B,UICxd

W,gBJ2vCvB,YAAK,KAAL,Ci3vCuB,CkCwdX,CtCmyB1B,C;cACI,sBAAO,K;cAAP,wB;;;UAGR,sBAAO,E;;;Q
sCvyBP,0B;O;KAPJ,C;+FAUA,yB;MAAA,sC;MtCm5BA,0D;MAAA,+C;MGv1CA,6B;MmCocA,yBAO+B,yB;Q
nC3c/B,6B;emC2c+B,6B;UAAA,qB;YAAE,yBnCjcm,cmCicI,EnCjcJ,CmCicN,C;W;S;OAAF,C;MAP/B,uC;QAO
mB,kBAAR,iB;QAAQ,sB;;;UtCg5BD,Q;UAAA,OAAQ,SAAR,wBAAQ,CAAR,W;UAAAd,OAAc,cAAAd,C;YAAc,u
B;YACV,IsCj5ByB,UnCjcM,cHk1CjB,YAAK,KAAL,CG11CiB,CmCicN,CtCi5BzB,C;cACI,qBAAO,K;cAAP,uB;
;UAGR,qBAAO,E;;;QsCr5BP,yB;O;KAPJ,C;+FAUA,yB;MAAA,sC;MtCq5BA,0D;MAAA,+C;MmBp1CA,+B;M
mB+bA,yBAO+B,yB;QnBtc/B,+B;emBsc+B,6B;UAAA,qB;YAAE,yBnB5bS,emB4bC,EnB5bD,CmB4bT,C;W;S;
OAAF,C;MAP/B,uC;QAOmB,kBAAR,iB;QAAQ,sB;;;UtCk5BD,Q;UAAA,OAAQ,SAAR,wBAAQ,CAAR,W;UA
Ad,OAAc,cAAAd,C;YAAc,uB;YACV,IsCn5ByB,UnB5bS,enB+0CpB,YAAK,KAAL,CmB/0CoB,CmB4bT,CtCm5
BzB,C;cACI,qBAAO,K;cAAP,uB;;;UAGR,qBAAO,E;;;QsCv5BP,yB;O;KAPJ,C;+FAUA,yB;MAAA,sC;MtCu2B
A,0D;MAAA,+C;ME72CA,+B;MoCsgBA,yBAO+B,yB;QpC7gB/B,+B;eoC6gB+B,6B;UAAA,qB;YAAE,yBpCng
BS,eoCmgBC,EpCngBD,CoCmgBT,C;W;S;OAAF,C;MAP/B,uC;QAOmB,kBAAR,iB;QAAQ,sB;;;UtCo2BD,Q;U
AAA,OAAQ,SAAR,wBAAQ,CAAR,W;UAAAd,OAAc,cAAAd,C;YAAc,uB;YACV,IsCr2ByB,UpCngBS,eFw2CpB,
YAAK,KAAL,CEX2CoB,CoCmgBT,CtCq2BzB,C;cACI,qBAAO,K;cAAP,uB;;;UAGR,qBAAO,E;;;QsCz2BP,yB;
O;KAPJ,C;+FAUA,yB;MAAA,sC;MtCy2BA,0D;MAAA,+C;MI52CA,iC;MkCmgBA,yBAO+B,yB;QlC1gB/B,iC;e
kC0gB+B,6B;UAAA,qB;YAAE,yBlChgBY,gBkCggBF,ElChgBE,CkCggBZ,C;W;S;OAAF,C;MAP/B,uC;QAOmB
,kBAAR,iB;QAAQ,sB;;;UtCs2BD,Q;UAAA,OAAQ,SAAR,wBAAQ,CAAR,W;UAAAd,OAAc,cAAAd,C;YAAc,uB;Y
ACV,IsCv2ByB,UlChgBY,gBJu2CvB,YAAK,KAAL,CiV2CuB,CkCggBZ,CtCu2BzB,C;cACI,qBAAO,K;cAAP,u
B;;;UAGR,qBAAO,E;;;QsC32BP,yB;O;KAPJ,C;+FAUA,yB;MAAA,4C;MnC5eA,6B;MmC4eA,4B;QAWI,OnC7e
mC,cmC6epB,KAAR,iBAAQ,CnC7eoB,C;O;KmCkevC,C;+FAUA,yB;MAAA,4C;MnB3eA,+B;MmB2eA,4B;QAW
I,OnB5esC,emB4evB,KAAR,iBAAQ,CnB5euB,C;O;KmBie1C,C;+FAUA,yB;MAAA,4C;MpCtjBA,+B;MoCsjBA,4
B;QAWI,OpCvjBsC,eoCujBvB,KAAR,iBAAQ,CpCvjBuB,C;O;KoC4iB1C,C;+FAUA,yB;MAAA,4C;MICvjBA,iC;
MkCujBA,4B;QAWI,OICxjByC,gBkCwjB1B,KAAR,iBAAQ,CiCxB0B,C;O;KkC6iB7C,C;+FAUA,yB;MAAA,+C;
MAAA,iE;MA83FI,0D;MA93FJ,uC;QAWkB,Q;QAAA,OAAa,SAm3FX,YAn3FF,SAm3FN,QAAQ,CAn3FW,CA
Ab,W;QAAd,OAAc,cAAAd,C;UAAc,uB;UACV,cAAc,sBAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YA
AwB,OAAO,O;QAEnC,MAAM,gCAAuB,mDAAvB,C;O;KafV,C;+FAUA,yB;MAAA,+C;MAAA,iE;MAo3FI,0
D;MAp3FJ,uC;QAWkB,Q;QAAA,OAAa,SAy2FX,YAz2FF,SAy2FN,QAAQ,CAz2FW,CAAb,W;QAAd,OAAc,cA
Ad,C;UAAc,uB;UACV,cAAc,sBAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;QAEnC,
MAAM,gCAAuB,mDAAvB,C;O;KafV,C;+FAUA,yB;MAAA,+C;MAAA,iE;MA02FI,0D;MA12FJ,uC;QAWkB,
Q;QAAA,OAAa,SA+1FX,YA/1FF,SA+1FN,QAAQ,CA/1FW,CAAb,W;QAAd,OAAc,cAAAd,C;UAAc,uB;UACV,c
AAc,sBAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;QAEnC,MAAM,gCAAuB,mDA
AvB,C;O;KafV,C;+FAUA,yB;MAAA,+C;MAAA,iE;MAg2FI,0D;MAh2FJ,uC;QAWkB,Q;QAAA,OAAa,SAq1F
X,YAr1FF,SAq1FN,QAAQ,CAr1FW,CAAb,W;QAAd,OAAc,cAAAd,C;UAAc,uB;UACV,cAAc,sBAAK,KAAL,C;
UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;QAEnC,MAAM,gCAAuB,mDAAvB,C;O;KafV,C;+FAU
A,yB;MAAA,0D;MAAA,qC;QAOI,OAAe,YAAR,iBAAQ,EAAY,OnC9sBM,KmC8sBIB,C;O;KAPnB,C;+FAU
A,yB;MAAA,0D;MAAA,qC;QAOI,OAAe,YAAR,iBAAQ,EAAY,OnB7sBQ,KmB6sBpB,C;O;KAPnB,C;+FAUA,
yB;MAAA,0D;MAAA,qC;QAOI,OAAe,YAAR,iBAAQ,EAAY,OpC1wBQ,KoC0wBpB,C;O;KAPnB,C;+FAUA,y
B;MAAA,0D;MAAA,qC;QAOI,OAAe,YAAR,iBAAQ,EAAY,OICzwBU,KkCywBtB,C;O;KAPnB,C;IAUA,kC;M
AQI,OAAW,mBAAJ,GAAe,IAAf,GAAyB,sBAAK,iBAAO,CAAP,IAAL,C;K;IAGpC,kC;MAQI,OAAW,mBAAJ,
GAAe,IAAf,GAAyB,sBAAK,iBAAO,CAAP,IAAL,C;K;IAGpC,kC;MAQI,OAAW,mBAAJ,GAAe,IAAf,GAAyB,s
BAAK,iBAAO,CAAP,IAAL,C;K;IAGpC,kC;MAQI,OAAW,mBAAJ,GAAe,IAAf,GAAyB,sBAAK,iBAAO,CAAP
,IAAL,C;K;6FAGpC,yB;MAAA,+C;MAkuFI,0D;MALuFJ,uC;QASkB,Q;QAAA,OAAa,SAytFX,YAz2FF,SAytFN,
QAAQ,CAztFW,CAAb,W;QAAd,OAAc,cAAAd,C;UAAc,uB;UACV,cAAc,sBAAK,KAAL,C;UACd,IAAI,UAAU,
OAAV,CAAJ,C;YAAwB,OAAO,O;QAEnC,OAAO,I;O;KAbX,C;6FAGBA,yB;MAAA,+C;MA0tFI,0D;MA1tFJ,u
C;QASkB,Q;QAAA,OAAa,SAitFX,YAjtFF,SAitFN,QAAQ,CAjtFW,CAAb,W;QAAd,OAAc,cAAAd,C;UAAc,uB;U
ACV,cAAc,sBAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;QAEnC,OAAO,I;O;KAbX
,C;6FAGBA,yB;MAAA,+C;MAktFI,0D;MAItFJ,uC;QASkB,Q;QAAA,OAAa,SAysFX,YAzsFF,SAysFN,QAAQ,C
AzsFW,CAAb,W;QAAd,OAAc,cAAAd,C;UAAc,uB;UACV,cAAc,sBAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,C

AAJ,C;YAAwB,OAAO,O;;QAEnC,OAAO,I;O;KAbX,C;6FAgBA,yB;MAAA,+C;MA0sFI,0D;MA1sFJ,uC;QASk
B,Q;QAAA,OAAa,SAisFX,YAjsFF,SAisFN,QAAQ,CAjsFW,CAAb,W;QAAd,OAAc,cAAAd,C;UAAc,uB;UACV,c
AAc,sBAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,OAAO,I;O;KAbX,C;qF
AgBA,yB;MAAA,mC;MAAA,gD;MAAA,4B;QASI,OAAO,kBAAO,cAAP,C;O;KATX,C;qFAYA,yB;MAAA,mC;
MAAA,gD;MAAA,4B;QASI,OAAO,kBAAO,cAAP,C;O;KATX,C;qFAYA,yB;MAAA,mC;MAAA,gD;MAAA,4B
;QASI,OAAO,kBAAO,cAAP,C;O;KATX,C;qFAYA,yB;MAAA,mC;MAAA,gD;MAAA,4B;QASI,OAAO,kBAAO
,cAAP,C;O;KATX,C;IAYA,sC;MAQI,IAAI,mBAAJ,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO,sBAA
I,MAAO,iBAAQ,cAAR,CAAX,C;K;IAGX,sC;MAQI,IAAI,mBAAJ,C;QACI,MAAM,2BAAuB,iBAAvB,C;MAC
V,OAAO,sBAAI,MAAO,iBAAQ,cAAR,CAAX,C;K;IAGX,sC;MAQI,IAAI,mBAAJ,C;QACI,MAAM,2BAAuB,iB
AAvB,C;MACV,OAAO,sBAAI,MAAO,iBAAQ,cAAR,CAAX,C;K;IAGX,sC;MAQI,IAAI,mBAAJ,C;QACI,MAA
M,2BAAuB,iBAAvB,C;MACV,OAAO,sBAAI,MAAO,iBAAQ,cAAR,CAAX,C;K;iGAGX,yB;MAAA,mC;MAA
A,4D;MAAA,4B;QAQI,OAAO,wBAAa,cAAb,C;O;KARX,C;iGAWA,yB;MAAA,mC;MAAA,4D;MAAA,4B;QA
QI,OAAO,wBAAa,cAAb,C;O;KARX,C;iGAWA,yB;MAAA,mC;MAAA,4D;MAAA,4B;QAQI,OAAO,wBAAa,c
AAb,C;O;KARX,C;iGAWA,yB;MAAA,mC;MAAA,4D;MAAA,4B;QAQI,OAAO,wBAAa,cAAb,C;O;KARX,C;I
AWA,4C;MAOI,IAAI,mBAAJ,C;QACI,OAAO,I;MACX,OAAO,sBAAI,MAAO,iBAAQ,cAAR,CAAX,C;K;IAGX
,4C;MAOI,IAAI,mBAAJ,C;QACI,OAAO,I;MACX,OAAO,sBAAI,MAAO,iBAAQ,cAAR,CAAX,C;K;IAGX,4C;
MAOI,IAAI,mBAAJ,C;QACI,OAAO,I;MACX,OAAO,sBAAI,MAAO,iBAAQ,cAAR,CAAX,C;K;IAGX,4C;MAO
I,IAAI,mBAAJ,C;QACI,OAAO,I;MACX,OAAO,sBAAI,MAAO,iBAAQ,cAAR,CAAX,C;K;qFAGX,yB;MAAA,g
D;MnCh8BA,6B;MmCg8BA,4B;QAOI,OnC77BmC,cmC67BpB,OAAR,iBAAQ,CnC77BoB,C;O;KmCs7BvC,C;q
FAUA,yB;MAAA,gD;MnB37BA,+B;MmB27BA,4B;QAOI,OnBx7BsC,emBw7BvB,OAAR,iBAAQ,CnBx7BuB,C
;O;KmBi7B1C,C;qFAUA,yB;MAAA,gD;MpClgCA,+B;MoCkgCA,4B;QAOI,OpC/BsC,eoC+/BvB,OAAR,iBAA
Q,CpC/BuB,C;O;KoCw/B1C,C;qFAUA,yB;MAAA,gD;MIC//BA,iC;MkC+/BA,4B;QAOI,OIC5/ByC,gBkC4/B1B,
OAAR,iBAAQ,CIC5/B0B,C;O;KkCq/B7C,C;qFAUA,yB;MAAA,kF;MAAA,iE;MAAA,wB;MAAA,8B;MAAA,uC
;QASoB,UAST,M;QAXP,aAAoB,I;QACpB,YAAY,K;QACI,2B;QAaHb,OAAGb,cAAhB,C;UAAgB,yB;UACZ,I
AAI,UAAU,OAAV,CAAJ,C;YACI,IAAI,KAJ,C;cAAW,MAAM,8BAAyB,gDAAzB,C;YACjB,SAAS,O;YACT,
QAAQ,I;;;QAGhB,IAAI,CAAC,KAAL,C;UAAy,MAAM,gCAAuB,mDAAvB,C;QAEIB,OAAO,0D;O;KAlBX,C;
qFAqBA,yB;MAAA,kF;MAAA,iE;MAAA,0B;MAAA,8B;MAAA,uC;QASoB,UAST,M;QAXP,aAAqB,I;QACrB,
YAAY,K;QACI,2B;QAaHb,OAAGb,cAAhB,C;UAAgB,yB;UACZ,IAAI,UAAU,OAAV,CAAJ,C;YACI,IAAI,KA
AJ,C;cAAW,MAAM,8BAAyB,gDAAzB,C;YACjB,SAAS,O;YACT,QAAQ,I;;;QAGhB,IAAI,CAAC,KAAL,C;UA
AY,MAAM,gCAAuB,mDAAvB,C;QAEIB,OAAO,2D;O;KAlBX,C;qFAqBA,yB;MAAA,kF;MAAA,iE;MAAA,0B
;MAAA,8B;MAAA,uC;QASoB,UAST,M;QAXP,aAAqB,I;QACrB,YAAY,K;QACI,2B;QAaHb,OAAGb,cAAhB,
C;UAAgB,yB;UACZ,IAAI,UAAU,OAAV,CAAJ,C;YACI,IAAI,KAJ,C;cAAW,MAAM,8BAAyB,gDAAzB,C;Y
ACjB,SAAS,O;YACT,QAAQ,I;;;QAGhB,IAAI,CAAC,KAAL,C;UAAy,MAAM,gCAAuB,mDAAvB,C;QAEIB,O
AAO,2D;O;KAlBX,C;qFAqBA,yB;MAAA,kF;MAAA,iE;MAAA,4B;MAAA,8B;MAAA,uC;QASoB,UAST,M;Q
AXP,aAAsB,I;QACtB,YAAY,K;QACI,2B;QAaHb,OAAGb,cAAhB,C;UAAgB,yB;UACZ,IAAI,UAAU,OAAV,C
AAJ,C;YACI,IAAI,KAJ,C;cAAW,MAAM,8BAAyB,gDAAzB,C;YACjB,SAAS,O;YACT,QAAQ,I;;;QAGhB,IA
AI,CAAC,KAAL,C;UAAy,MAAM,gCAAuB,mDAAvB,C;QAEIB,OAAO,4D;O;KAlBX,C;IAqBA,oC;MAMI,OA
AW,mBAAQ,CAAZ,GAAe,sBAAK,CAAL,CAAF,GAA4B,I;K;IAGvC,oC;MAMI,OAAW,mBAAQ,CAAZ,GAAe,sBAAK,CAAL,CAAF,GAA
4B,I;K;IAGvC,oC;MAMI,OAAW,mBAAQ,CAAZ,GAAe,sBAAK,CAAL,CAAF,GAA4B,I;K;iGAGvC,gC;MASoB
,Q;MAFhB,aAAoB,I;MACpB,YAAY,K;MACI,2B;MAAhB,OAAGb,cAAhB,C;QAAGb,yB;QACZ,IAAI,UAAU,O
AAV,CAAJ,C;UACI,IAAI,KAJ,C;YAAW,OAAO,I;UACIB,SAAS,O;UACT,QAAQ,I;;;MAGhB,IAAI,CAAC,K
AAL,C;QAAY,OAAO,I;MACnB,OAAO,M;K;iGAGX,gC;MASoB,Q;MAFhB,aAAqB,I;MACrB,YAAY,K;MACI,
2B;MAAhB,OAAGb,cAAhB,C;QAAGb,yB;QACZ,IAAI,UAAU,OAAV,CAAJ,C;UACI,IAAI,KAJ,C;YAAW,O
AAO,I;UACIB,SAAS,O;UACT,QAAQ,I;;;MAGhB,IAAI,CAAC,KAAL,C;QAAY,OAAO,I;MACnB,OAAO,M;K;i
GAGX,gC;MASoB,Q;MAFhB,aAAqB,I;MACrB,YAAY,K;MACI,2B;MAAhB,OAAGb,cAAhB,C;QAAGb,yB;QA
CZ,IAAI,UAAU,OAAV,CAAJ,C;UACI,IAAI,KAJ,C;YAAW,OAAO,I;UACIB,SAAS,O;UACT,QAAQ,I;;;MAG
hB,IAAI,CAAC,KAAL,C;QAAY,OAAO,I;MACnB,OAAO,M;K;iGAGX,gC;MASoB,Q;MAFhB,aAAsB,I;MACTB

,YAAy,K;MACI,2B;MAAhB,OAAGB,cAAhB,C;QAAgB,yB;QACZ,IAAI,UAAU,OAaV,CAAJ,C;UACI,IAAI,KAAJ,C;YAAW,OAaO,I;UACIB,SAAS,O;UACT,QAAQ,I;;;MAGhB,IAAI,CAAC,KAAL,C;QAAy,OAaO,I;MAcNB,OAaO,M;K;IAGX,+B;MxBRhDI,IAAI,EwB+hDI,KAAK,CxB/hDT,CAAJ,C;QACI,cwB8hDc,sD;QxB7hDd,MAAM,gCAAYB,OAaQ,WAAjC,C;;MwB8hDV,OAaO,uBAAoB,gBAAV,iBAAO,CAAP,IAAU,EAAC,CAAd,CAApB,C;K;IAGX,+B;MxBniDI,IAAI,EwB6iDI,KAAK,CxB7iDT,CAAJ,C;QACI,cwB4iDc,sD;QxB3iDd,MAAM,gCAAYB,OAaQ,WAAjC,C;;MwB4iDV,OAaO,uBAAoB,gBAAV,iBAAO,CAAP,IAAU,EAAC,CAAd,CAApB,C;K;IAGX,+B;MxBjjDI,IAAI,EwB2jDI,KAAK,CxB3jDT,CAAJ,C;QACI,cwB0jDc,sD;QxBzjDd,MAAM,gCAAYB,OAaQ,WAAjC,C;;MwB0jDV,OAaO,uBAAoB,gBAAV,iBAAO,CAAP,IAAU,EAAC,CAAd,CAApB,C;K;IAGX,+B;MxB/jDI,IAAI,EwBykDI,KAAK,CxBzkDT,CAAJ,C;QACI,cwBwkDc,sD;QxBvkDd,MAAM,gCAAYB,OAaQ,WAAjC,C;;MwBwkDV,OAaO,uBAAoB,gBAAV,iBAAO,CAAP,IAAU,EAAC,CAAd,CAApB,C;K;IAGX,mC;MxB7kDI,IAAI,EwBulDI,KAAK,CxBvlDT,CAAJ,C;QACI,cwBslDc,sD;QxBrlDd,MAAM,gCAAYB,OAaQ,WAAjC,C;;MwBslDV,OAaO,mBAAgB,gBAAV,iBAAO,CAAP,IAAU,EAAC,CAAd,CAAhB,C;K;IAGX,mC;MxB3lDI,IAAI,EwBqmDI,KAAK,CxBrmDT,CAAJ,C;QACI,cwBomDc,sD;QxBnmDd,MAAM,gCAAYB,OAaQ,WAAjC,C;;MwBomDV,OAaO,mBAAgB,gBAAV,iBAAO,CAAP,IAAU,EAAC,CAAd,CAAhB,C;K;IAGX,mC;MxBzmDI,IAAI,EwBmnDI,KAAK,CxBnnDT,CAAJ,C;QACI,cwBknDc,sD;QxBjnDd,MAAM,gCAAYB,OAaQ,WAAjC,C;;MwBknDV,OAaO,mBAAgB,gBAAV,iBAAO,CAAP,IAAU,EAAC,CAAd,CAAhB,C;K;IAGX,mC;MxBvnDI,IAAI,EwBioDI,KAAK,CxBjoDT,CAAJ,C;QACI,cwBgoDc,sD;QxB/nDd,MAAM,gCAAYB,OAaQ,WAAjC,C;;MwBgoDV,OAaO,mBAAgB,gBAAV,iBAAO,CAAP,IAAU,EAAC,CAAd,CAAhB,C;K;mGAGX,yB;MAAA,4C;MAAA,qD;MAkqEI,8D;MAIqEJ,uC;QASI,iBAypEgB,cAAR,iBAAQ,CAzpEhB,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,sBAAK,KAAL,CAAV,CAAL,C;YACI,OAaO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAaO,W;O;KAdX,C;mGAiBA,yB;MAAA,4C;MAAA,qD;MAypEI,8D;MAzpEJ,uC;QASI,iBAgpEgB,cAAR,iBAAQ,CAhpEhB,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,sBAAK,KAAL,CAAV,CAAL,C;YACI,OAaO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAaO,W;O;KAdX,C;mGAiBA,yB;MAAA,4C;MAAA,qD;MAGpEI,8D;MAhpEJ,uC;QASI,iBAuoEgB,cAAR,iBAAQ,CAvoEhB,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,sBAAK,KAAL,CAAV,CAAL,C;YACI,OAaO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAaO,W;O;KAdX,C;mGAiBA,yB;MAAA,4C;MAAA,qD;MAuoEI,8D;MAvoEJ,uC;QASI,iBA8nEgB,cAAR,iBAAQ,CA9nEhB,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,sBAAK,KAAL,CAAV,CAAL,C;YACI,OAaO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAaO,W;O;KAdX,C;2FAiBA,yB;MAAA,+D;MAAA,uC;QAWiB,Q;QAFb,eAAe,K;QACf,WAAW,gB;QACE,2B;QAAb,OAaA,cAAb,C;UAAa,sB;UACT,IAAI,QAAJ,C;YACI,IAAK,WAAI,IAAJ,C;eACJ,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACD,IAAK,WAAI,IAAJ,C;YACL,WAAW,I;;;QAEEnB,OAaO,I;O;KAIBX,C;2FAqBA,yB;MAAA,+D;MAAA,uC;QAWiB,Q;QAFb,eAAe,K;QACf,WAAW,gB;QACE,2B;QAAb,OAaA,cAAb,C;UAAa,sB;UACT,IAAI,QAAJ,C;YACI,IAAK,WAAI,IAAJ,C;eACJ,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACD,IAAK,WAAI,IAAJ,C;YACL,WAAW,I;;;QAEEnB,OAaO,I;O;KAIBX,C;2FAqBA,yB;MAAA,+D;MAAA,uC;QASW,kBAAS,gB;QAgRA,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UAAM,IAhRa,SAgRT,CAAU,OAaV,CAAJ,C;YAAwB,WAAy,WAAI,OAaJ,C;;QAhR1D,OAIRO,W;O;KA1RX,C;qFAYA,yB;MAAA,+D;MAAA,uC;QASW,kBAAS,gB;QAIrA,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UAAM,IAjRc,SAiRV,CAAU,OAaV,CAAJ,C;YAAwB,WAAy,WAAI,OAaJ,C;;QAJR1D,OAKRO,W;O;KA3RX,C;qFAYA,yB;MAAA,+D;MAAA,uC;QASW,kBAAS,gB;QAKrA,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UAAM,IAIRc,SAkRV,CAAU,OAaV,CAAJ,C;YAAwB,WAAy,WAAI,OAaJ,C;;QAIr1D,OAmRO,W;O;KA5RX,C;qFAYA,yB;MAAA,+D;MAAA,uC;QASW,kBAAS,gB;QAmRA,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UAAM,IAAnRe,SAmRX,CAAU,OAaV,CAAJ,C;YAAwB,WAAy,WAAI,OAaJ,C;;QAnR1D,OAoRO,W;O;KA7RX,C;kGAYA,yB;MAAA,+D;MAAA,uC;QAWW,kBAAGB,gB;QAm5HV,gB;QADb,YAAy,C;QACC,2B;QAAb,OAaA,cAAb,C;UAAa,sB;UA11HT,IAZDsC,SAyDIC,EA01HkB,cA11HIB,EA01HkB,sBA11HIB,WA01H2B,IA11H3B,CAAJ,C;YAA2C,sBA01

C;MACnB,IAAI,SAAQ,CAAZ,C;QA Ae,OAAO,W;MACtB,WAAW,iBA AiB,IAAjB,C;MACG,yB;MAAd,OAAc,cAAd,C;QA Ac,uB;QACV,IAAK,WAAI,sBA AI,KAAJ,CAAJ,C;;MAET,OAAO,I;K;IAGX,sC;MAskB,Q;MAHd,WAAmB,wBAAR,OAAQ,EAAwB,EAAxB,C;MACnB,IAAI,SAAQ,CAAZ,C;QA Ae,OAAO,W;MACtB,WAAW,iBA AiB,IAAjB,C;MACG,yB;MAAd,OAAc,cAAd,C;QA Ac,uB;QACV,IAAK,WAAI,sBA AI,KAAJ,CAAJ,C;;MAET,OAAO,I;K;IAGX,sC;MAskB,Q;MAHd,WAAmB,wBAAR,OAAQ,EAAwB,EAAxB,C;MACnB,IAAI,SAAQ,CAAZ,C;QA Ae,OAAO,W;MACtB,WAAW,iBA AiB,IAAjB,C;MACG,yB;MAAd,OAAc,cAAd,C;QA Ac,uB;QACV,IAAK,WAAI,sBA AI,KAAJ,CAAJ,C;;MAET,OAAO,I;K;IAGX,2C;MAMI,OAAO,cAAkB,aAAR,iBAAQ,EAAW,OAA X,CAAI B,C;K;IAGX,2C;MAMI,OAAO,eAAmB,aAAR,iBAAQ,EAAW,OAA X,CAAnB,C;K;IAGX,2C;MAMI,OAAO,eAAmB,aAAR,iBAAQ,EAAW,OAA X,CAAnB,C;K;IAGX,2C;MAMI,OAAO,gBAAoB,aAAR,iBAAQ,EAAW,OAA X,CAApB,C;K;IAGX,2C;MAMI,OAAO,cAAkB,cAAR,iBAAQ,EAAW,OAA X,CAAI B,C;K;IAGX,2C;MAMI,OAAO,eAAmB,cAAR,iBAAQ,EAAW,OAA X,CAAnB,C;K;IAGX,2C;MAMI,OAAO,eAAmB,aAAR,iBAAQ,EAAW,OAA X,CAAnB,C;K;IAGX,2C;MAMI,OAAO,gBAAoB,cAAR,iBAAQ,EAAW,OAA X,CAApB,C;K;IAGX,+B;MAGBiB,Q;MxB7xEb,IAAI,EwBuxEI,KAAK,CxBvxET,CAAJ,C;QACI,cwBsxEc,sD;QxBrxEd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MwBsxEV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,cAAT,C;QA Ae,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,sBAAK,CAAL,CAAP,C;MACnB,YAAY,C;MACZ,WAAW,iBA AgB,CAAhB,C;MACE,2B;MAAb,OAAa,cAAb,C;QA Aa,sB;QACT,IAAK,WAAI,IAAJ,C;QACL,IAAI,mCAAW,CAAf,C;UACI,K;;MAER,OAAO,I;K;IAGX,+B;MAGBiB,Q;MxBrzEb,IAAI,EwB+yEI,KAAK,CxB/yET,CAAJ,C;QACI,cwB8yEc,sD;QxB7yEd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MwB8yEV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,cAAT,C;QA Ae,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,sBAAK,CAAL,CAAP,C;MACnB,YAAY,C;MACZ,WAAW,iBA AiB,CAAjB,C;MACE,2B;MAAb,OAAa,cAAb,C;QA Aa,sB;QACT,IAAK,WAAI,IAAJ,C;QACL,IAAI,mCAAW,CAAf,C;UACI,K;;MAER,OAAO,I;K;IAGX,+B;MAGBiB,Q;MxB70Eb,IAAI,EwBu0EI,KAAK,CxBv0ET,CAAJ,C;QACI,cwBs0Ec,sD;QxB r0Ed,MAAM,gCAAYB,OAAQ,WAAjC,C;;MwBs0EV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,cAAT,C;QA Ae,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,sBAAK,CAAL,CAAP,C;MACnB,YAAY,C;MACZ,WAAW,iBA AiB,CAAjB,C;MACE,2B;MAAb,OAAa,cAAb,C;QA Aa,sB;QACT,IAAK,WAAI,IAAJ,C;QACL,IAAI,mCAAW,CAAf,C;UACI,K;;MAER,OAAO,I;K;IAGX,+B;MAGBiB,Q;MxB2Eb,IAAI,EwB+1EI,KAAK,CxB/1ET,CAAJ,C;QACI,cwB81Ec,sD;QxB71Ed,MAAM,gCAAYB,OAAQ,WAAjC,C;;MwB81EV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,cAAT,C;QA Ae,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,sBAAK,CAAL,CAAP,C;MACnB,YAAY,C;MACZ,WAAW,iBA AkB,CAAI B,C;MACE,2B;MAAb,OAAa,cAAb,C;QA Aa,sB;QACT,IAAK,WAAI,IAAJ,C;QACL,IAAI,mCAAW,CAAf,C;UACI,K;;MAER,OAAO,I;K;IAGX,mC;MxB72EI,IAAI,EwBu3EI,KAAK,CxBv3ET,CAAJ,C;QACI,cwBs3Ec,sD;QxB r3Ed,MAAM,gCAAYB,OAAQ,WAAjC,C;;MwBs3EV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,c;MACX,IAAI,KAAK,IAAT,C;QA Ae,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,sBAAK,OAAO,CAAP,IAAL,CAAP,C;MACnB,WAAW,iBA AgB,CAAhB,C;MACX,iBA Ac,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,WAAI,sBAAK,KAAL,CAAJ,C;MACT,OAAO,I;K;IAGX,mC;MxB14EI,IAAI,EwB44EI,KAAK,CxB54ET,CAAJ,C;QACI,cwB24Ec,sD;QxB14Ed,MAAM,gCAAYB,OAAQ,WAAjC,C;;MwB24EV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,c;MACX,IAAI,KAAK,IAAT,C;QA Ae,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,sBAAK,OAAO,CAAP,IAAL,CAAP,C;MACnB,WAAW,iBA AiB,CAAjB,C;MACX,iBA Ac,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,WAAI,sBAAK,KAAL,CAAJ,C;MACT,OAAO,I;K;IAGX,mC;MxBv5EI,IAAI,EwBi6EI,KAAK,CxBj6ET,CAAJ,C;QACI,cwBg6Ec,sD;QxB/5Ed,MAAM,gCAAYB,OAAQ,WAAjC,C;;MwBg6EV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,c;MACX,IAAI,KAAK,IAAT,C;QA Ae,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,sBAAK,OAAO,CAAP,IAAL,CAAP,C;MACnB,WAAW,iBA AkB,CAAI B,C;MACX,iBA Ac,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,WAAI,sBAAK,KAAL,CAAJ,C;MACT,

OAAO,I;K;mGAGX,yB;MAAA,4C;MAAA,gD;MA s2CI,8D;Mat2CJ,uC;QASI,iBA61CgB,cAAR,iBAAQ,CA71C
hB,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,sBAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,QAAQ
,CAAR,IAAL,C;;;QAGf,OAAO,iB;O;KAdX,C;mGAiBA,yB;MAAA,4C;MAAA,gD;MA61CI,8D;MA71CJ,uC;QA
SI,iBAo1CgB,cAAR,iBAAQ,Cap1ChB,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,sBAAK,KAAL,CAAV,C
AAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,iB;O;KAdX,C;mGAiBA,yB;MAAA,4C;M
AAA,gD;MAo1CI,8D;MAp1CJ,uC;QASI,iBA20CgB,cAAR,iBAAQ,CA30ChB,WAA+B,CAA/B,U;UACI,IAAI,C
AAC,UAAU,sBAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,iB;O
;KAdX,C;mGAiBA,yB;MAAA,4C;MAAA,gD;MA20CI,8D;MA30CJ,uC;QASI,iBAk0CgB,cAAR,iBAAQ,CA10C
hB,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,sBAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,QAAQ
,CAAR,IAAL,C;;;QAGf,OAAO,iB;O;KAdX,C;2FAiBA,yB;MAAA,+D;MAAA,uC;QAUiB,Q;QADb,WAAW,gB;
QACE,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACI,K;UACJ,IAAK,W
AAI,IAAJ,C;;QAET,OAAO,I;O;KAFX,C;2FAkBA,yB;MAAA,+D;MAAA,uC;QAUiB,Q;QADb,WAAW,gB;QAC
E,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACI,K;UACJ,IAAK,WAAI,I
AAJ,C;;QAET,OAAO,I;O;KAFX,C;2FAkBA,yB;MAAA,+D;MAAA,uC;QAUiB,Q;QADb,WAAW,gB;QACE,2B;
QAAb,OAAa,cAAb,C;UAAa,sB;UACT,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACI,K;UACJ,IAAK,WAAI,IAAJ,
C;;QAET,OAAO,I;O;KAFX,C;2FAkBA,yB;MAAA,+D;MAAA,uC;QAUiB,Q;QADb,WAAW,gB;QACE,2B;QAA
b,OAAa,cAAb,C;UAAa,sB;UACT,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACI,K;UACJ,IAAK,WAAI,IAAJ,C;;QA
ET,OAAO,I;O;KAFX,C;uFAkBA,yB;MAAA,kD;MAAA,4B;QAOY,QAAR,iBAAQ,C;O;KAPZ,C;uFAUA,yB;MA
AA,kD;MAAA,4B;QAOY,QAAR,iBAAQ,C;O;KAPZ,C;uFAUA,yB;MAAA,kD;MAAA,4B;QAOY,QAAR,iBAA
Q,C;O;KAPZ,C;uFAUA,yB;MAAA,kD;MAAA,4B;QAOY,QAAR,iBAAQ,C;O;KAPZ,C;uFAUA,yB;MAAA,kD;
MAAA,gD;QAaY,QAAR,iBAAQ,EAAQ,SAAR,EAAmB,OAA nB,C;O;KAbZ,C;uFAgBA,yB;MAAA,kD;MAAA,
gD;QAaY,QAAR,iBAAQ,EAAQ,SAAR,EAAmB,OAA nB,C;O;KAbZ,C;uFAgBA,yB;MAAA,kD;MAAA,gD;QAaY,QAA
R,iBAAQ,EAAQ,SAAR,EAAmB,OAA nB,C;O;KAbZ,C;IAgBA,gC;MAMI,IAAI,mBAAJ,C;QAAe,OAAO,W;MA
CtB,WAAW,0B;MACN,WAAL,IAAK,C;MACL,OAAO,I;K;IAGX,gC;MAMI,IAAI,mBAAJ,C;QAAe,OAAO,W;
MACtB,WAAW,0B;MACN,WAAL,IAAK,C;MACL,OAAO,I;K;IAGX,gC;MAMI,IAAI,mBAAJ,C;QAAe,OAAO,
W;MACtB,WAAW,0B;MACN,WAAL,IAAK,C;MACL,OAAO,I;K;IAGX,gC;MAMI,IAAI,mBAAJ,C;QAAe,OA
AO,W;MACtB,WAAW,0B;MACN,WAAL,IAAK,C;MACL,OAAO,I;K;kGAGX,yB;MAAA,8D;MAAA,uC;MAA
A,4B;QAOI,OAAO,mBAAkB,cAAR,iBAAQ,CAAIB,C;O;KAPX,C;kGAUA,yB;MAAA,8D;MAAA,yC;MAAA,4
B;QAOI,OAAO,oBAAmB,cAAR,iBAAQ,CAAnB,C;O;KAPX,C;mGAUA,yB;MAAA,8D;MAAA,yC;MAAA,4B;
QAOI,OAAO,oBAAmB,cAAR,iBAAQ,CAAnB,C;O;KAPX,C;mGAUA,yB;MAAA,8D;MAAA,2C;MAAA,4B;Q
AOI,OAAO,qBAAoB,cAAR,iBAAQ,CAApB,C;O;KAPX,C;IAUA,+B;MAMI,sBAAQ,4BAAR,C;K;IAGJ,+B;MA
MI,sBAAQ,4BAAR,C;K;IAGJ,+B;MAMI,sBAAQ,4BAAR,C;K;IAGJ,+B;MAMI,sBAAQ,4BAAR,C;K;IAGJ,uC;
MAQI,aA8+BgB,gBAAR,iBAAQ,CA9+BhB,OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAA
R,C;QACf,WAAW,sBAAK,CAAL,C;QACX,sBAAK,CAAL,EAAU,sBAAK,CAAL,CAAV,C;QACA,sBAAK,CA
AL,EAAU,IAAV,C;;K;IAIR,uC;MAQI,aA8+BgB,gBAAR,iBAAQ,CA9+BhB,OAA2B,CAA3B,M;QACI,QAAQ,M
AAO,iBAAQ,IAAI,CAAJ,IAAR,C;QACf,WAAW,sBAAK,CAAL,C;QACX,sBAAK,CAAL,EAAU,sBAAK,CAA
L,CAAV,C;QACA,sBAAK,CAAL,EAAU,IAAV,C;;K;IAIR,uC;MAQI,aA89BgB,gBAAR,iBAAQ,CA99BhB,OA
A2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAAR,C;QACf,WAAW,sBAAK,CAAL,C;QACX,sBA
AK,CAAL,EAAU,sBAAK,CAAL,CAAV,C;QACA,sBAAK,CAAL,EAAU,IAAV,C;;K;IAIR,uC;MAQI,aA9BgB,
gBAAR,iBAAQ,CA9BhB,OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAAR,C;QACf,WAAW
,sBAAK,CAAL,C;QACX,sBAAK,CAAL,EAAU,sBAAK,CAAL,CAAV,C;QACA,sBAAK,CAAL,EAAU,IAAV,C
;;K;IAIR,sC;MAMI,IAAI,iBAAO,CAAX,C;QACI,iB;QApSI,UAAR,iBAAQ,C;;K;IAySZ,sC;MAMI,IAAI,iBAAO
,CAAX,C;QACI,iB;QAtSI,UAAR,iBAAQ,C;;K;IA2SZ,sC;MAMI,IAAI,iBAAO,CAAX,C;QACI,iB;QAxSI,UAAR
,iBAAQ,C;;K;IA6SZ,sC;MAMI,IAAI,iBAAO,CAAX,C;QACI,iB;QA1SI,UAAR,iBAAQ,C;;K;IA+SZ,6B;MAMoB
,kBA+nBT,cAAU,iBvB58EO,QuB48EjB,C;MA/nBiB,mB;MAAxB,OAAiC,SrBv3F1B,WqBu3F0B,C;K;IAGrC,8B
;MAMoB,kBAkoBT,eAAmB,UAAR,iBAAQ,CAAnB,C;MAl0BiB,mB;MAAxB,OAAiC,SrBh4F1B,WqBg4F0B,C;
K;IAGrC,8B;MAMoB,kBAqoBT,eAAW,iBvBx/EM,QuBw/EjB,C;MAroBiB,mB;MAAxB,OAAiC,UrBz4F1B,Wq

By4F0B,C;K;IAGrC,8B;MAMoB,kBAwoBT,gBAAY,iBvB1/EK,QuB0/EjB,C;MAxoBiB,mB;MAAxB,OAAiC,Ur
Bl5F1B,WqBk5F0B,C;K;IAGrC,kC;MAMI,IAAI,mBAAJ,C;QAAe,OAAO,S;MACD,kBA01Bd,cA11BA,SA01BU,
QvB58EO,QuB48EjB,C;MA11BsB,mB;MAA7B,OrB55FO,W;K;IqB+5FX,kC;MAMI,IAAI,mBAAJ,C;QAAe,OA
AO,S;MACD,kBA41Bd,eAAmB,UA51BnB,SA41BW,QAAQ,CAAnB,C;MA51BsB,mB;MAA7B,OrBt6FO,W;K;Iq
By6FX,kC;MAMI,IAAI,mBAAJ,C;QAAe,OAAO,S;MACD,kBA81Bd,eA91BA,SA81BW,QvBx/EM,QuBw/EjB,C;
MA91BsB,mB;MAA7B,OrBh7FO,W;K;IqBm7FX,mC;MAMI,IAAI,mBAAJ,C;QAAe,OAAO,S;MACD,kBAgmbd
,gBAhmBA,SAGmBY,QvB1/EK,QuB0/EjB,C;MAhmBsB,mB;MAA7B,OrB17FO,W;K;IqB67FX,4C;MAMI,IAAI,
mBAAJ,C;QAAe,OAAO,S;MACD,kBAkjBd,cAljBA,SAkjBU,QvB58EO,QuB48EjB,C;MALjBsB,8B;MAA7B,Or
Bp8FO,W;K;IqBu8FX,4C;MAMI,IAAI,mBAAJ,C;QAAe,OAAO,S;MACD,kBAojBd,eAAmB,UApjBnB,SAojBW,
QAAQ,CAAnB,C;MApjBsB,8B;MAA7B,OrB98FO,W;K;IqBi9FX,4C;MAMI,IAAI,mBAAJ,C;QAAe,OAAO,S;M
ACD,kBASjBd,eAtjBA,SAsjBW,QvBx/EM,QuBw/EjB,C;MATjBsB,8B;MAA7B,OrBx9FO,W;K;IqB29FX,6C;MA
MI,IAAI,mBAAJ,C;QAAe,OAAO,S;MACD,kBAwjBd,gBAxjBA,SAwjBY,QvB1/EK,QuB0/EjB,C;MAxjBsB,8B;
MAA7B,OrB1+FO,W;K;IqBq+FX,uC;MAQoB,kBAygBT,cAAU,iBvB58EO,QuB48EjB,C;MAzgBiB,mB;MAAxB
,OAAiC,YrB7+F1B,WqB6+F0B,C;K;IAGrC,wC;MAQoB,kBA0gBT,eAAmB,UAR,iBAAQ,CAAnB,C;MA1gBi
B,mB;MAAxB,OAAiC,YrBx/F1B,WqBw/F0B,C;K;IAGrC,wC;MAQoB,kBA2gBT,eAAW,iBvBx/EM,QuBw/EjB,
C;MA3gBiB,mB;MAAxB,OAAiC,YrBngG1B,WqBmgG0B,C;K;IAGrC,wC;MAQoB,kBA4gBT,gBAAY,iBvB1/E
K,QuB0/EjB,C;MA5gBiB,mB;MAAxB,OAAiC,YrB9gG1B,WqB8gG0B,C;K;4FAGrC,qB;MAQI,OAAO,iB;K;0F
AGX,qB;MAQI,OAAO,iB;K;4FA+BX,qB;MAQI,OAAO,iB;K;8FAGX,qB;MAQI,OAAO,iB;K;8FAGX,yB;MAA
A,yC;MAAA,4B;QAQI,OAAO,oBAAW,SAAX,C;O;KARX,C;4FAWA,yB;MAAA,uC;MAAA,4B;QAQI,OAAO,
mBAAU,SAAV,C;O;KARX,C;8FAWA,yB;MAAA,yC;MAAA,4B;QAQI,OAAO,oBAAW,SAAX,C;O;KARX,C;g
GAWA,yB;MAAA,2C;MAAA,4B;QAQI,OAAO,qBAAY,SAAZ,C;O;KARX,C;IAWA,2C;MASI,OAAy,gBAAL,
SAAK,EAAC,KAAd,C;K;IAGhB,2C;MASI,OAAy,gBAAL,SAAK,EAAC,KAAd,C;K;IAGhB,2C;MASI,OAAy,g
BAAL,SAAK,EAAC,KAAd,C;K;IAGhB,2C;MASI,OAAy,gBAAL,SAAK,EAAC,KAAd,C;K;IAGhB,2C;MAOI,O
AAqB,cAAAd,4CAAc,EAAC,oCAAd,C;K;IAGzB,2C;MAOI,OAAqB,cAAAd,4CAAc,EAAC,oCAAd,C;K;IAGzB,2C;
MAOI,OAAqB,cAAAd,4CAAc,EAAC,oCAAd,C;K;IAGzB,2C;MAOI,OAAqB,cAAAd,4CAAc,EAAC,oCAAd,C;K;IA
GzB,sC;MAQI,OAAy,kBAAL,SAAK,C;K;IAGhB,sC;MAQI,OAAy,kBAAL,SAAK,C;K;IAGhB,sC;MAQI,OAA
Y,kBAAL,SAAK,C;K;IAGhB,sC;MAQI,OAAy,kBAAL,SAAK,C;K;IAGhB,sC;MAMI,OAAqB,gBAAd,4CAAc,
C;K;IAGzB,sC;MAMI,OAAqB,gBAAd,4CAAc,C;K;IAGzB,sC;MAMI,OAAqB,gBAAd,4CAAc,C;K;IAGzB,sC;
MAMI,OAAqB,gBAAd,4CAAc,C;K;IAGzB,sC;MAUI,OAAy,kBAAL,SAAK,C;K;IAGhB,sC;MAUI,OAAy,kBA
AL,SAAK,C;K;IAGhB,sC;MAUI,OAAy,kBAAL,SAAK,C;K;IAGhB,sC;MAUI,OAAy,kBAAL,SAAK,C;K;IAGh
B,sC;MAQW,Q;MAAP,OAAO,sDAAmB,IAAnB,EAAyB,GAAzB,EAA8B,GAA9B,2BAAsC,M;K;IAGjD,sC;MA
QW,Q;MAAP,OAAO,sDAAmB,IAAnB,EAAyB,GAAzB,EAA8B,GAA9B,2BAAsC,M;K;IAGjD,sC;MAQW,Q;M
AAP,OAAO,sDAAmB,IAAnB,EAAyB,GAAzB,EAA8B,GAA9B,2BAAsC,M;K;IAGjD,sC;MAQW,Q;MAAP,OA
AO,sDAAmB,IAAnB,EAAyB,GAAzB,EAA8B,GAA9B,2BAAsC,M;K;SFAgjD,yB;MvBxhFA,8C;MuBwhFA,kF;
QAmB6D,iC;UAAA,oBAAYB,C;QAAG,0B;UAAA,aAAkB,C;QAAG,wB;UAAA,WAAGB,c;QvBvhF1H,UuBwhF
A,iBvBxhFA,EuBwhFiB,WAAY,QvBxhF7B,EuBwhFsC,iBvBxhFiC,EuBwhFyD,UvBxhFzD,EuBwhFqE,QvBxhF
rE,C;QuByhFA,OAAO,W;O;KArBX,C;wFAwBA,yB;MvBxhFA,8C;MuBwhFA,kF;QAmB+D,iC;UAAA,oBAAY
B,C;QAAG,0B;UAAA,aAAkB,C;QAAG,wB;UAAA,WAAGB,c;QvBvhF5H,UuBwhFA,iBvBxhFA,EuBwhFiB,W
AAY,QvBxhF7B,EuBwhFsC,iBvBxhFiC,EuBwhFyD,UvBxhFzD,EuBwhFqE,QvBxhFrE,C;QuByhFA,OAAO,W;
O;KArBX,C;wFAwBA,yB;MvBxnFA,8C;MuBwnFA,kF;QAmB+D,iC;UAAA,oBAAYB,C;QAAG,0B;UAAA,aAA
kB,C;QAAG,wB;UAAA,WAAGB,c;QvBvnF5H,UuBwnFA,iBvBxnFA,EuBwnFiB,WAAY,QvBxnF7B,EuBwnFsC
,iBvBxnFiC,EuBwnFyD,UvBxnFzD,EuBwnFqE,QvBxnFrE,C;QuBynFA,OAAO,W;O;KArBX,C;wFAwBA,yB;M
vBxnFA,8C;MuBwnFA,kF;QAmBiE,iC;UAAA,oBAAYB,C;QAAG,0B;UAAA,aAAkB,C;QAAG,wB;UAAA,WA
AGB,c;QvBvnF9H,UuBwnFA,iBvBxnFA,EuBwnFiB,WAAY,QvBxnF7B,EuBwnFsC,iBvBxnFiC,EuBwnFyD,UvB
xnFzD,EuBwnFqE,QvBxnFrE,C;QuBynFA,OAAO,W;O;KArBX,C;kFAwBA,yB;MAAA,uC;MAAA,4B;QASI,OA
AO,mBAAU,iBvB58EO,QuB48EjB,C;O;KATX,C;oFAYA,yB;MAAA,gD;MAAA,yC;MAAA,4B;QASI,OAAO,o
BAAmB,OAAR,iBAAQ,CAAnB,C;O;KATX,C;oFAYA,yB;MAAA,yC;MAAA,4B;QASI,OAAO,oBAAW,iBvBx/
EM,QuBw/EjB,C;O;KATX,C;oFAYA,yB;MAAA,2C;MAAA,4B;QASI,OAAO,qBAAY,iBvB1/EK,QuB0/EjB,C;O

;KATX,C;oFAYA,yB;MAAA,gD;MAAA,uC;MAAA,qC;QAWI,OAAO,mBAAkB,OAAR,iBAAQ,EAAO,OAAP,CAAIB,C;O;KAXX,C;oFAcA,yB;MAAA,gD;MAAA,yC;MAAA,qC;QAWI,OAAO,oBAAmB,OAAR,iBAAQ,EA AO,OAAP,CAAnB,C;O;KAXX,C;oFAcA,yB;MAAA,+C;MAAA,yC;MAAA,qC;QAWI,OAAO,oBAAmB,OAAR, iBAAQ,EAAO,OAAP,CAAnB,C;O;KAXX,C;oFAcA,yB;MAAA,gD;MAAA,2C;MAAA,qC;QAWI,OAAO,qBAA oB,OAAR,iBAAQ,EAAO,OAAP,CAApB,C;O;KAXX,C;4FAcA,yB;MAAA,0D;MAAA,uC;MAAA,gD;QAaI,OA AO,mBAAkB,YAAR,iBAAQ,EAAy,SAAZ,EAAuB,OAAvB,CAAIB,C;O;KAbX,C;8FAgBA,yB;MAAA,0D;MA AAA,yC;MAAA,gD;QAaI,OAAO,oBAAmB,YAAR,iBAAQ,EAAy,SAAZ,EAAuB,OAAvB,CAAnB,C;O;KAbX,C; 8FAgBA,yB;MAAA,0D;MAAA,yC;MAAA,gD;QAaI,OAAO,oBAAmB,YAAR,iBAAQ,EAAy,SAAZ,EAAuB,O AA vB,CAAnB,C;O;KAbX,C;6FAgBA,yB;MAAA,0D;MAAA,2C;MAAA,gD;QAaI,OAAO,qBAAoB,YAAR,iBA AQ,EAAy,SAAZ,EAAuB,OAAvB,CAApB,C;O;KAbX,C;IAGBA,sD;MAWYc,yB;QAAA,YAAiB,C;MAAG,uB; QAAA,UAAe,c;MACHe,OAAR,iBAAQ,EAAK,OnCv8GoB,KmCu8GzB,EAAsB,SAAtB,EAAiC,OAAjC,C;K;IA GZ,wD;MAW2C,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,c;MACIE,OAAR,iBAAQ,EAAK,OnB38GsB,K mB28G3B,EAAuB,SAAvB,EAAkC,OAAIC,C;K;IAGZ,wD;MAW2C,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,U AAe,c;MACIE,OAAR,iBAAQ,EAAK,OpC7gHsB,KoC6gH3B,EAAuB,SAAvB,EAAkC,OAAIC,C;K;IAGZ,wD;M AW6C,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,c;MACpE,OAAR,iBAAQ,EAAK,OICjhHwB,KkCihH7B,E AAwB,SAAxB,EAAmC,OAAnc,C;K;8FASR,yB;MAAA,0D;MAAA,4B;QAAQ,OAAQ,YAAR,iBAAQ,C;O;KA AhB,C;8FAQA,yB;MAAA,0D;MAAA,4B;QAAQ,OAAQ,YAAR,iBAAQ,C;O;KAAhB,C;+FAQA,yB;MAAA,0D; MAAA,4B;QAAQ,OAAQ,YAAR,iBAAQ,C;O;KAAhB,C;+FAQA,yB;MAAA,0D;MAAA,4B;QAAQ,OAAQ,YA AR,iBAAQ,C;O;KAAhB,C;kGAQA,yB;MAAA,8D;MAAA,4B;QAAQ,OAAQ,cAAR,iBAAQ,C;O;KAAhB,C;kG AQA,yB;MAAA,8D;MAAA,4B;QAAQ,OAAQ,cAAR,iBAAQ,C;O;KAAhB,C;mGAQA,yB;MAAA,8D;MAAA,4 B;QAAQ,OAAQ,cAAR,iBAAQ,C;O;KAAhB,C;mGAQA,yB;MAAA,8D;MAAA,4B;QAAQ,OAAQ,cAAR,iBAA Q,C;O;KAAhB,C;iFAEJ,yB;MAAA,uC;MvBvoEA,iD;MuBuoEA,qC;QAOqB,4B;QAAA,gBAAU,OnC9jHM,K;Q mC8jHjC,OAAO,mBvBzoEA,2BAxIK,gBAAW,SAAX,EAwIL,CuByoEA,C;O;KAPX,C;iFAUA,yB;MAAA,yC;M vBzoEA,iD;MuByoEA,qC;QAOI,OAAO,oBvB3oEA,qBuB2oEW,iBvB3oEX,EAxIK,mBuBmxEgB,OnB7jHO,KJ0 yCvB,CAwIL,CuB2oEA,C;O;KAPX,C;iFAUA,yB;MAAA,yC;MvB3qEA,iD;MuB2qEA,qC;QAOsB,4B;QAAA,g BAAU,OpC1nHO,K;QoC0nHnC,OAAO,oBvB7qEA,2BAxIK,eAAY,SAAZ,EAwIL,CuB6qEA,C;O;KAPX,C;iFA UA,yB;MAAA,2C;MvB7qEA,iD;MuB6qEA,qC;QAOuB,4B;QAAA,gBAAU,OICznHQ,K;QkCynHrC,OAAO,qBv B/qEA,2BAxIK,gBAAa,SAAb,EAwIL,CuB+qEA,C;O;KAPX,C;IAUA,sC;MAQoB,UAAiB,M;MAFjC,YAAy,c; MACZ,aAAqB,UAAR,iBAAQ,EAAO,iBAAO,QAAS,KAAhB,IAAP,C;MACL,0B;MAAhB,OAAGB,cAAhB,C;Q AAAGB,yB;QAAU,OAAO,cAAP,EAAO,sBAAP,YAAkB,OnCvmHX,K;;MmCwmHjC,OAAO,cAAU,MAAV,C;K;I AGX,sC;MAQoB,UAAiB,M;MAFjC,YAAy,c;MACZ,aAAqB,UAAR,iBAAQ,EAAO,iBAAO,QAAS,KAAhB,IA AP,C;MACL,0B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAU,OAAO,cAAP,EAAO,sBAAP,YAAkB,OnBxmH T,K;;MmBymHnC,OAAO,eAAW,MAAX,C;K;IAGX,sC;MAQoB,UAAiB,M;MAFjC,YAAy,c;MACZ,aAAqB,U AAR,iBAAQ,EAAO,iBAAO,QAAS,KAAhB,IAAP,C;MACL,0B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAU, OAAO,cAAP,EAAO,sBAAP,YAAkB,OpCvqHT,K;;MoCwqHnC,OAAO,eAAW,MAAX,C;K;IAGX,sC;MAQoB, UAAiB,M;MAFjC,YAAy,c;MACZ,aAAqB,UAAR,iBAAQ,EAAO,iBAAO,QAAS,KAAhB,IAAP,C;MACL,0B;M AAhB,OAAGB,cAAhB,C;QAAGB,yB;QAAU,OAAO,cAAP,EAAO,sBAAP,YAAkB,OICxqHP,K;;MkCyqHrC,OA AO,gBAAy,MAAZ,C;K;iFAGX,yB;MAAA,uC;MvB/tEA,iD;MuB+tEA,sC;QAOI,OAAO,mBvBjuEA,qBuBiuEU, iBvBjuEV,EUbiuEoB,QAAS,QvBjuE7B,CuBiuEA,C;O;KAPX,C;iFAUA,yB;MAAA,yC;MvBjuEA,iD;MuBiuEA,s C;QAOI,OAAO,oBvBnuEA,qBuBmuEW,iBvBnuEX,EUbmuEqB,QAAS,QvBnuE9B,CuBmuEA,C;O;KAPX,C;iF AUA,yB;MAAA,yC;MvBnwEA,iD;MuBmwEA,sC;QAOI,OAAO,oBvBrwEA,qBuBqwEW,iBvBrwEX,EUbqwEq B,QAAS,QvBrwE9B,CuBqwEA,C;O;KAPX,C;iFAUA,yB;MAAA,2C;MvBrwEA,iD;MuBqwEA,sC;QAOI,OAAO ,qBvBvwEA,qBuBwEY,iBvBvwEZ,EUbuwEsB,QAAS,QvBvwE/B,CuBwEA,C;O;KAPX,C;IAUA,2B;MAQI,IA AI,iBAAO,CAAX,C;QAAC,YAAU,SAAV,EAAGB,CAAhB,EAAMB,cAAnB,C;K;IAGIB,2B;MAQI,IAAI,iBAA O,CAAX,C;QAAC,YAAU,SAAV,EAAGB,CAAhB,EAAMB,cAAnB,C;K;IAGIB,2B;MAQI,IAAI,iBAAO,CAAX,C ;QAAC,YAAU,SAAV,EAAGB,CAAhB,EAAMB,cAAnB,C;K;IAGIB,+C;MAA0B,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,U AAe,c;MACzD,oCAAA,2BAAkB,SAAlB,EAA6B,OAA7B,EAAc,cAAtC,C;MACb,YAAU,SAAV,EAAGB,SAAh

B,EAA2B,OAA3B,C;K;IAGJ,+C;MAa2B,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,c;MAC1D,oCAAa,2BA
AkB,SAAlB,EAA6B,OAA7B,EAAc,cAAtC,C;MACb,YAAU,SAAV,EAAgB,SAAhB,EAA2B,OAA3B,C;K;IAG
J,+C;MAa2B,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,c;MAC1D,oCAAa,2BAAkB,SAAlB,EAA6B,OAA7
B,EAAc,cAAtC,C;MACb,YAAU,SAAV,EAAgB,SAAhB,EAA2B,OAA3B,C;K;IAGJ,+C;MAa4B,yB;QAAA,YA
AiB,C;MAAG,uB;QAAA,UAAe,c;MAC3D,oCAAa,2BAAkB,SAAlB,EAA6B,OAA7B,EAAc,cAAtC,C;MACb,
YAAU,SAAV,EAAgB,SAAhB,EAA2B,OAA3B,C;K;IAGJ,0D;MAaI,kBAAK,SAAL,EAAgB,OAAhB,C;MAh8C
Q,WAAR,iBAAQ,EAI8CA,SAj8CA,EAI8CW,OAJ8CX,C;K;IAo8CZ,0D;MAaI,kBAAK,SAAL,EAAgB,OAAhB,C
;MAj8CQ,WAAR,iBAAQ,EAK8CA,SAI8CA,EAK8CW,OAI8CX,C;K;IAq8CZ,0D;MAaI,kBAAK,SAAL,EAAgB,
OAAhB,C;MAI8CQ,UAAAR,iBAAQ,EAm8CA,SAn8CA,EAm8CW,OAn8CX,C;K;IAS8CZ,0D;MAaI,kBAAK,SA
AL,EAAgB,OAAhB,C;MAN8CQ,WAAR,iBAAQ,EAo8CA,SAP8CA,EAo8CW,OAP8CX,C;K;8FAu8CZ,qB;MAQ
I,OAAO,iBvB3jGiB,Q;K;4FuB8jG5B,qB;MAQI,OAAO,iBvBljGiB,Q;K;8FuBqjG5B,yB;MAAA,gD;MAAA,4B;Q
AQI,OAAe,OAAR,iBAAQ,C;O;KARnB,C;gGAWA,qB;MAQI,OAAO,iBvBlIgiB,Q;K;IuB2lIGL,gD;MAAA,wB;Q
AAW,qCAAK,KAAL,C;O;K;IANIC,iC;MAMI,OAAO,iBAAM,cAAN,EAAY,8BAAZ,C;K;IASY,kD;MAAA,wB;
QAAW,qCAAK,KAAL,C;O;K;IANIC,mC;MAMI,OAAO,iBAAM,cAAN,EAAY,gCAAZ,C;K;IASY,kD;MAAA,w
B;QAAW,qCAAK,KAAL,C;O;K;IANIC,mC;MAMI,OAAO,iBAAM,cAAN,EAAY,gCAAZ,C;K;IASY,kD;MAAA
,wB;QAAW,qCAAK,KAAL,C;O;K;IANIC,mC;MAMI,OAAO,iBAAM,cAAN,EAAY,gCAAZ,C;K;IASiB,gD;MA
AA,wB;QAAW,yBAAK,KAAL,C;O;K;IANvC,iC;MAMI,OJnqIO,eAAW,+BImqIA,gBJnqIA,GAAgB,kBImqIV,8
BJnqIU,CAAhB,CAAX,C;K;gGIsqIX,yB;MAAA,yC;MAAA,4B;QAQI,OAAO,oBAAW,SvBppGM,QuBopGjB,C;
O;KARX,C;IAiB2B,8C;MAAA,wB;QAAW,wBAAK,KAAL,C;O;K;IANtC,gC;MAMI,OHvrIO,cAAU,gCGurIA,g
BHvrIA,GAAe,iBGurIT,6BHvrIS,CAAf,CAAV,C;K;8FG0rIX,yB;MAAA,uC;MAAA,4B;QAQI,OAAO,mBAAU,
SvBppGO,QuBopGjB,C;O;KARX,C;IAiB4B,gD;MAAA,wB;QAAW,yBAAK,KAAL,C;O;K;IANvC,iC;MAMI,OF
3sIO,eAAW,kBE2sIA,gBF3sIA,EAAgB,kBE2sIV,8BF3sIU,CAAhB,CAAX,C;K;gGE8sIX,yB;MAAA,gD;MAAA,
yC;MAAA,4B;QAQI,OAAO,oBAAgB,OAAL,SAAK,CAAhB,C;O;KARX,C;IAiB6B,kD;MAAA,wB;QAAW,0BA
AK,KAAL,C;O;K;IANxC,kC;MAMI,OD/tIO,gBAAY,gCC+tIA,gBD/tIA,GAAiB,mBC+tIX,+BD/tIW,CAAjB,CA
AZ,C;K;kGCKuIX,yB;MAAA,2C;MAAA,4B;QAQI,OAAO,qBAAY,SvBtsGK,QuBssGjB,C;O;KARX,C;mGAWA
,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,2C;QAcI,aAAa,mBAAyC,cAAIB,YAAY,cAAZ,CAAKB,EAAc,EA
Ad,CAAzC,C;QAsEG,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UArEO,MAsEP,aAAI,OAAJ,EAtEe,a
AsEF,CAAc,OAAd,CAAb,C;;QAtEhB,OAAuB,M;O;Kaf3B,C;mGakBA,yB;MAAA,0D;MAAA,yD;MAAA,uE;
MAAA,2C;QAcI,aAAa,mBAA0C,cAAIB,YAAY,cAAZ,CAAKB,EAAc,EAAAd,CAA1C,C;QAsEG,Q;QAAA,2B;Q
AAhB,OAAGB,cAAhB,C;UAAgB,yB;UArEO,MAsEP,aAAI,OAAJ,EAtEe,aAsEF,CAAc,OAAd,CAAb,C;;QAtEh
B,OAAuB,M;O;Kaf3B,C;kGakBA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,2C;QAcI,aAAa,mBAA0C,cAA
IB,YAAY,cAAZ,CAAKB,EAAc,EAAAd,CAA1C,C;QAsEG,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;
UArEO,MAsEP,aAAI,OAAJ,EAtEe,aAsEF,CAAc,OAAd,CAAb,C;;QAtEhB,OAAuB,M;O;Kaf3B,C;mGakBA,y
B;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,2C;QAcI,aAAa,mBAA2C,cAAIB,YAAY,cAAZ,CAAKB,EAAc,EAA
d,CAA3C,C;QAsEG,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UArEO,MAsEP,aAAI,OAAJ,EAtEe,a
AsEF,CAAc,OAAd,CAAb,C;;QAtEhB,OAAuB,M;O;Kaf3B,C;uGakBA,iD;MAYoB,Q;MAAA,2B;MAAhB,OA
gB,cAAhB,C;QAAGB,yB;QACZ,WAAY,aAAI,OAAJ,EAAa,cAAc,OAAd,CAAb,C;;MAEhB,OAAO,W;K;uGAG
X,iD;MAYoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,WAAY,aAAI,OAAJ,EAAa,cAAc,OA
Ad,CAAb,C;;MAEhB,OAAO,W;K;uGAGX,iD;MAYoB,Q;MAAA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;Q
ACZ,WAAY,aAAI,OAAJ,EAAa,cAAc,OAAd,CAAb,C;;MAEhB,OAAO,W;K;uGAGX,iD;MAYoB,Q;MAAA,2B;
MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,WAAY,aAAI,OAAJ,EAAa,cAAc,OAAd,CAAb,C;;MAEhB,OAA
O,W;K;uFAGX,yB;MAAA,+D;MAoLA,gD;MAPLA,uC;QASW,kBAAU,gB;QAKLD,Q;QAAA,2B;QAAhB,OA
gB,cAAhB,C;UAAgB,yB;UACZ,WAnL6B,SAmLIB,CAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QA
pLhB,OAsLO,W;O;KA/LX,C;uFAYA,yB;MAAA,+D;MAsLA,gD;MatLA,uC;QASW,kBAAU,gB;QAoLD,Q;QA
AA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,WArL6B,SAqLIB,CAAU,OAAV,C;UACC,OAAZ,WAAY,
EAAO,IAAP,C;;QAtLhB,OAwLO,W;O;KAjMX,C;uFAYA,yB;MAAA,+D;MAwLA,gD;MAxLA,uC;QASW,kBA
AU,gB;QAsLD,Q;QAAA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,WAvL6B,SAuLIB,CAAU,OAAV,C;
UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAxLhB,OAoLO,W;O;KAnMX,C;uFAYA,yB;MAAA,+D;MAoLA,gD;M

A1LA,uC;QASW,kBAAU,gB;QAwLD,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WaZL6B,SA
yLIB,CAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QA1LhB,OA4LO,W;O;KArMX,C;qGAYA,yB;MA
AA,+D;MA4DA,gD;MA5DA,uC;QAYW,kBAAiB,gB;QA2DR,gB;QADhB,YAAY,C;QACI,2B;QAAhB,OAAgB,
cAAhB,C;UAAgB,yB;UACZ,WA5DoC,SA4DzB,EAAU,cAAV,EAAU,sBAAV,WAAMb,OAAAnB,C;UACC,OAA
Z,WAAY,EAAO,IAAP,C;;QA7DhB,OA+DO,W;O;KA3EX,C;qGAeA,yB;MAAA,+D;MA+DA,gD;MA/DA,uC;Q
AYW,kBAAiB,gB;QA8DR,gB;QADhB,YAAY,C;QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WA/
DoC,SA+DzB,EAAU,cAAV,EAAU,sBAAV,WAAMb,OAAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAhEh
B,OAkEO,W;O;KA9EX,C;qGAeA,yB;MAAA,+D;MAkEA,gD;MAIEA,uC;QAYW,kBAAiB,gB;QAIeR,gB;QAD
hB,YAAY,C;QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAIeOC,SAkEzB,EAAU,cAAV,EAAU,sB
AAV,WAAMb,OAAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAnEhB,OAqEO,W;O;KAjFX,C;qGAeA,yB;
MAAA,+D;MAqEA,gD;MArEA,uC;QAYW,kBAAiB,gB;QAoER,gB;QADhB,YAAY,C;QACI,2B;QAAhB,OAAg
B,cAAhB,C;UAAgB,yB;UACZ,WArEOC,SAqEzB,EAAU,cAAV,EAAU,sBAAV,WAAMb,OAAAnB,C;UACC,OA
AZ,WAAY,EAAO,IAAP,C;;QAtEhB,OAwEO,W;O;KApFX,C;yGAeA,yB;MAAA,gD;MAAA,oD;QAWoB,UAC
S,M;QAFzB,YAAY,C;QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAAW,WAAU,cAAV,EAAU,s
BAAV,WAAMb,OAAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KAfX,C;yGAKBA,yB;
MAAA,gD;MAAA,oD;QAWoB,UACS,M;QAFzB,YAAY,C;QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;U
ACZ,WAAW,WAAU,cAAV,EAAU,sBAAV,WAAMb,OAAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB
,OAAO,W;O;KAfX,C;yGAKBA,yB;MAAA,gD;MAAA,oD;QAWoB,UACS,M;QAFzB,YAAY,C;QACI,2B;QAAh
B,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAAW,WAAU,cAAV,EAAU,sBAAV,WAAMb,OAAAnB,C;UACC,OA
AZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KAfX,C;yGAKBA,yB;MAAA,gD;MAAA,oD;QAWoB,UACS,
M;QAFzB,YAAY,C;QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAAW,WAAU,cAAV,EAAU,sB
AAV,WAAMb,OAAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KAfX,C;2FAkBA,yB;MAA
A,gD;MAAA,oD;QAOoB,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAAW,UAAU,OAAV,C;
UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KAXX,C;2FAcA,yB;MAAA,gD;MAAA,oD;QAOoB
,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAY,E
AAO,IAAP,C;;QAEhB,OAAO,W;O;KAXX,C;2FAcA,yB;MAAA,gD;MAAA,oD;QAOoB,Q;QAAA,2B;QAAhB,
OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,
OAAO,W;O;KAXX,C;2FAcA,yB;MAAA,gD;MAAA,oD;QAOoB,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAA
gB,yB;UACZ,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KAXX,C;u
FAcA,yB;MAAA,wE;MA4HA,+D;MA5HA,yC;QAYW,kBAAU,oB;QA4HD,Q;QAAA,2B;QAAhB,OAAgB,cAA
hB,C;UAAgB,yB;UACZ,UA7HoD,WA6H1C,CAAY,OAAZ,C;U/B59IP,U;UADP,Y+B89Ie,W/B99IH,W+B89IwB
,G/B99IxB,C;UACL,IAAI,aAAJ,C;YACH,a+B49IuC,gB;YAA5B,W/B39IX,a+B29IgC,G/B39IhC,EAAS,MAAT,C
;YACA,e;;YAEA,c;;U+Bw9IA,iB;UACA,IAAK,WAAI,OAAJ,C;;QA/HT,OAIiO,W;O;KA7IX,C;uFAeA,yB;MAA
A,wE;MAiIA,+D;MAjIA,yC;QAYW,kBAAU,oB;QAIID,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;U
ACZ,UAIqD,WakI3C,CAAY,OAAZ,C;U/Bh/IP,U;UADP,Y+Bk/Ie,W/BI/IH,W+Bk/IwB,G/BI/IxB,C;UACL,IAAI
,aAAJ,C;YACH,a+Bg/IuC,gB;YAA5B,W/B+/IX,a+B++IgC,G/B+/IhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;U+B4
+IA,iB;UACA,IAAK,WAAI,OAAJ,C;;QApIT,OAsIO,W;O;KAIJX,C;sFAeA,yB;MAAA,wE;MAsIA,+D;MAtIA,y
C;QAYW,kBAAU,oB;QAsID,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UAvIqD,WauI3C,CA
AY,OAAZ,C;U/BpgJP,U;UADP,Y+BsgJe,W/BtgJH,W+BsgJwB,G/BtgJxB,C;UACL,IAAI,aAAJ,C;YACH,a+BogJ
uC,gB;YAA5B,W/BngJX,a+BmgJgC,G/BngJhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;U+BggJA,iB;UACA,IAAK,
WAAI,OAAJ,C;;QAzIT,OA2IO,W;O;KAvJX,C;uFAeA,yB;MAAA,wE;MA2IA,+D;MA3IA,yC;QAYW,kBAAU,o
B;QA2ID,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UA5IsD,WA4I5C,CAAY,OAAZ,C;U/Bxh
JP,U;UADP,Y+B0hJe,W/B1hJH,W+B0hJwB,G/B1hJxB,C;UACL,IAAI,aAAJ,C;YACH,a+BwhJuC,gB;YAA5B,W
/BvhJX,a+BuhJgC,G/BvhJhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;U+B0hJA,iB;UACA,IAAK,WAAI,OAAJ,C;;Q
A9IT,OAgJO,W;O;KA5JX,C;uFAeA,yB;MAAA,wE;MAgJA,+D;MAhJA,yD;QAaW,kBAAU,oB;QAgJD,Q;QAA
A,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UAjJiD,WaiJvC,CAAY,OAAZ,C;U/B7iJP,U;UADP,Y+B+iJ
e,W/B/iJH,W+B+iJwB,G/B/iJxB,C;UACL,IAAI,aAAJ,C;YACH,a+B6iJuC,gB;YAA5B,W/B5iJX,a+B4iJgC,G/B5i
JhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;U+ByiJA,iB;UACA,IAAK,WAnJyD,cAmJrD,CAAE,OAAf,CAAJ,C;;QA

nJT,OAqJO,W;O;KAIKX,C;uFAGBA,yB;MAAA,wE;MAqJA,+D;MArJA,yD;QAaW,kBAAU,oB;QAqJD,Q;QAA
A,2B;QAaHb,OAAGB,cAAhB,C;UAAgB,yB;UACZ,UAtJiD,WAsJvC,CAAY,OAAZ,C;U/BlkJP,U;UADP,Y+Bok
Je,W/BpkJH,W+BokJwB,G/BpkJxB,C;UACL,IAAI,aAAJ,C;YACH,a+BkkJuC,gB;YAA5B,W/BjkJX,a+BikJgC,G/
BjkJhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;U+B8jJA,iB;UACA,IAAK,WaxJyD,cAwJrD,CAAe,OAAf,CAAJ,C;;
QAxJT,OA0JO,W;O;KAvKX,C;uFAGBA,yB;MAAA,wE;MA0JA,+D;MA1JA,yD;QAaW,kBAAU,oB;QA0JD,Q;Q
AAA,2B;QAaHb,OAAGB,cAAhB,C;UAAgB,yB;UACZ,UA3JiD,WA2JvC,CAAY,OAAZ,C;U/BvIJP,U;UADP,Y+
BylJe,W/BzljH,W+BylJwB,G/BzljxB,C;UACL,IAAI,aAAJ,C;YACH,a+BulJuC,gB;YAA5B,W/BtlJX,a+BslJgC,G/
BtlJhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;U+BmlJA,iB;UACA,IAAK,WA7JyD,cA6JrD,CAAe,OAAf,CAAJ,C;;
QA7JT,OA+JO,W;O;KA5KX,C;uFAGBA,yB;MAAA,wE;MA+JA,+D;MA/JA,yD;QAaW,kBAAU,oB;QA+JD,Q;Q
AAA,2B;QAaHb,OAAGB,cAAhB,C;UAAgB,yB;UACZ,UAhKiD,WAgKvC,CAAY,OAAZ,C;U/B5mJP,U;UADP,
Y+B8mJe,W/B9mJH,W+B8mJwB,G/B9mJxB,C;UACL,IAAI,aAAJ,C;YACH,a+B4mJuC,gB;YAA5B,W/B3mJX,a
+B2mJgC,G/B3mJhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;U+BwmJA,iB;UACA,IAAK,WAlKyD,cAkKrD,CAAe,
OAAf,CAAJ,C;;QAIKT,OAoKO,W;O;KAjLX,C;2FAGBA,yB;MAAA,+D;MAAA,sD;QAYoB,Q;QAAA,2B;QAA
hB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,UAAU,YAAY,OAAZ,C;U/B59IP,U;UADP,Y+B89Ie,W/B99IH,W+B89
IwB,G/B99Ix,C;UACL,IAAI,aAAJ,C;YACH,a+B49IuC,gB;YAA5B,W/B39IX,a+B29IgC,G/B39IhC,EAAS,MA
AT,C;YACA,e;;YAEA,c;;U+Bw9IA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KAjBX,C;2FAoBA,y
B;MAAA,+D;MAAA,sD;QAYoB,Q;QAAA,2B;QAaHb,OAAGB,cAAhB,C;UAAgB,yB;UACZ,UAAU,YAAY,O
AAZ,C;U/Bh/IP,U;UADP,Y+Bk/Ie,W/BI/IH,W+Bk/IwB,G/BI/IxB,C;UACL,IAAI,aAAJ,C;YACH,a+Bg/IuC,gB;Y
AA5B,W/B/+IX,a+B++IgC,G/B/+IhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;U+B4+IA,iB;UACA,IAAK,WAAI,OA
AJ,C;;QAET,OAAO,W;O;KAjBX,C;2FAoBA,yB;MAAA,+D;MAAA,sD;QAYoB,Q;QAAA,2B;QAaHb,OAAGB,
cAAhB,C;UAAgB,yB;UACZ,UAAU,YAAY,OAAZ,C;U/BpgJP,U;UADP,Y+BsgJe,W/BtgJH,W+BsgJwB,G/BtgJx
B,C;UACL,IAAI,aAAJ,C;YACH,a+BogJuC,gB;YAA5B,W/BngJX,a+BmgJgC,G/BngJhC,EAAS,MAAT,C;YACA
e;;YAEA,c;;U+BggJA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KAjBX,C;2FAoBA,yB;MAAA,+D;
MAAA,sD;QAYoB,Q;QAAA,2B;QAaHb,OAAGB,cAAhB,C;UAAgB,yB;UACZ,UAAU,YAAY,OAAZ,C;U/BxhJ
P,U;UADP,Y+B0hJe,W/B1hJH,W+B0hJwB,G/B1hJxB,C;UACL,IAAI,aAAJ,C;YACH,a+BwhJuC,gB;YAA5B,W/
BvhJX,a+BuhJgC,G/BvhJhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;U+BohJA,iB;UACA,IAAK,WAAI,OAAJ,C;;Q
AET,OAAO,W;O;KAjBX,C;2FAoBA,yB;MAAA,+D;MAAA,sE;QAaoB,Q;QAAA,2B;QAaHb,OAAGB,cAAhB,C
;UAAgB,yB;UACZ,UAAU,YAAY,OAAZ,C;U/B7iJP,U;UADP,Y+B+iJe,W/B/iJH,W+B+iJwB,G/B/iJxB,C;UACL
IAAI,aAAJ,C;YACH,a+B6iJuC,gB;YAA5B,W/B5iJX,a+B4iJgC,G/B5iJhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;
U+ByiJA,iB;UACA,IAAK,WAAI,eAAe,OAAf,CAAJ,C;;QAET,OAAO,W;O;KAIBX,C;2FAqBA,yB;MAAA,+D;
MAAA,sE;QAaoB,Q;QAAA,2B;QAaHb,OAAGB,cAAhB,C;UAAgB,yB;UACZ,UAAU,YAAY,OAAZ,C;U/BlkJP,
U;UADP,Y+BokJe,W/BpkJH,W+BokJwB,G/BpkJxB,C;UACL,IAAI,aAAJ,C;YACH,a+BkkJuC,gB;YAA5B,W/Bj
kJX,a+BikJgC,G/BjkJhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;U+B8jJA,iB;UACA,IAAK,WAAI,eAAe,OAAf,CA
AJ,C;;QAET,OAAO,W;O;KAIBX,C;2FAqBA,yB;MAAA,+D;MAAA,sE;QAaoB,Q;QAAA,2B;QAaHb,OAAGB,c
AAhB,C;UAAgB,yB;UACZ,UAAU,YAAY,OAAZ,C;U/BvIJP,U;UADP,Y+BylJe,W/BzljH,W+BylJwB,G/BzljxB,
C;UACL,IAAI,aAAJ,C;YACH,a+BulJuC,gB;YAA5B,W/BtlJX,a+BslJgC,G/BtlJhC,EAAS,MAAT,C;YACA,e;;YA
EA,c;;U+BmlJA,iB;UACA,IAAK,WAAI,eAAe,OAAf,CAAJ,C;;QAET,OAAO,W;O;KAIBX,C;2FAqBA,yB;MAA
A,+D;MAAA,sE;QAaoB,Q;QAAA,2B;QAaHb,OAAGB,cAAhB,C;UAAgB,yB;UACZ,UAAU,YAAY,OAAZ,C;U/
B5mJP,U;UADP,Y+B8mJe,W/B9mJH,W+B8mJwB,G/B9mJxB,C;UACL,IAAI,aAAJ,C;YACH,a+B4mJuC,gB;YA
A5B,W/B3mJX,a+B2mJgC,G/B3mJhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;U+BwmJA,iB;UACA,IAAK,WAAI,e
AAe,OAAf,CAAJ,C;;QAET,OAAO,W;O;KAIBX,C;+EAqBA,yB;MAAA,gE;MAAA,uC;QAUW,kBAAM,eAAa,c
AAb,C;QAsKA,Q;QAAA,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,WAAy,WAvKiB,SAuKb,CAAU,IAAV,CA
AJ,C;;QAvKhB,OAwKO,W;O;KAILX,C;+EAaA,yB;MAAA,gE;MAAA,uC;QAUW,kBAAM,eAAa,cAAb,C;QAs
KA,Q;QAAA,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,WAAy,WAvKiB,SAuKb,CAAU,IAAV,CAAJ,C;;QAv
KhB,OAwKO,W;O;KAILX,C;8EAaA,yB;MAAA,gE;MAAA,uC;QAUW,kBAAM,eAAa,cAAb,C;QAsKA,Q;QAA
A,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,WAAy,WAvKiB,SAuKb,CAAU,IAAV,CAAJ,C;;QAvKhB,OAwK
O,W;O;KAILX,C;+EAaA,yB;MAAA,gE;MAAA,uC;QAUW,kBAAM,eAAa,cAAb,C;QAsKA,Q;QAAA,2B;QAAb
,OAAa,cAAb,C;UAAa,sB;UACT,WAAy,WAvKiB,SAuKb,CAAU,IAAV,CAAJ,C;;QAvKhB,OAwKO,W;O;KAI

gB,yB;QAAM,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;;MACpC,OAAO,W;K;iFAGX,yC;MAaoB,Q;MADhB,kB
AAkB,O;MACF,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QAAM,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;;M
ACpC,OAAO,W;K;+FAGX,yC;MAeoB,UAA8B,M;MAF9C,YAAY,C;MACZ,kBAaKB,O;MACF,2B;MAAhB,O
AAgB,cAAhB,C;QAAGB,yB;QAAM,cAAc,WAAU,cAAV,EAAU,sBAAV,WAAmB,WAAAnB,EAAgC,OAAhC,C;
;MACpC,OAAO,W;K;+FAGX,yC;MAeoB,UAA8B,M;MAF9C,YAAY,C;MACZ,kBAaKB,O;MACF,2B;MAAhB,
OAAgB,cAAhB,C;QAAGB,yB;QAAM,cAAc,WAAU,cAAV,EAAU,sBAAV,WAAmB,WAAAnB,EAAgC,OAAhC,
C;;MACpC,OAAO,W;K;+FAGX,yC;MAeoB,UAA8B,M;MAF9C,YAAY,C;MACZ,kBAaKB,O;MACF,2B;MAAhB
B,OAAgB,cAAhB,C;QAAGB,yB;QAAM,cAAc,WAAU,cAAV,EAAU,sBAAV,WAAmB,WAAAnB,EAAgC,OAAh
C,C;;MACpC,OAAO,W;K;+FAGX,yC;MAeoB,UAA8B,M;MAF9C,YAAY,C;MACZ,kBAaKB,O;MACF,2B;MA
AhB,OAAgB,cAAhB,C;QAAGB,yB;QAAM,cAAc,WAAU,cAAV,EAAU,sBAAV,WAAmB,WAAAnB,EAAgC,OA
AhC,C;;MACpC,OAAO,W;K;0FAGX,yB;MA1uDI,8D;MA0uDj,gD;QAeoC,Q;QAHhC,YAtvDgB,cAAR,iBAAQ,
C;QAuvDhB,kBAaKB,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,uBAAI,YAAJ,EAAI,oBAAJ,QAA
V,EAAwB,WAAxB,C;;QAEIB,OAAO,W;O;KAjBX,C;2FAoBA,yB;MATvDI,8D;MASvDJ,gD;QAeoC,Q;QAHhC,
YAlwDgB,cAAR,iBAAQ,C;QAmwDhB,kBAaKB,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,uBAA
I,YAAJ,EAAI,oBAAJ,QAAV,EAAwB,WAAxB,C;;QAEIB,OAAO,W;O;KAjBX,C;2FAoBA,yB;MAIwDI,8D;MA
kwDJ,gD;QAeoC,Q;QAHhC,YA9wDgB,cAAR,iBAAQ,C;QA+wDhB,kBAaKB,O;QACIB,OAAO,SAAS,CAAhB,
C;UACI,cAAc,UAAU,uBAAI,YAAJ,EAAI,oBAAJ,QAAV,EAAwB,WAAxB,C;;QAEIB,OAAO,W;O;KAjBX,C;2
FAoBA,yB;MA9wDI,8D;MA8wDJ,gD;QAeoC,Q;QAHhC,YA1xDgB,cAAR,iBAAQ,C;QA2xDhB,kBAaKB,O;Q
ACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,uBAAI,YAAJ,EAAI,oBAAJ,QAAV,EAAwB,WAAxB,C;;QA
EIB,OAAO,W;O;KAjBX,C;yGAoBA,yB;MA1zDI,8D;MA0zDJ,gD;QAaI,YAv0DgB,cAAR,iBAAQ,C;QAw0DhB,
kBAaKB,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,sBAAI,KAJ,CAAjB,EAA6B,
WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KANBX,C;yGAsBA,yB;MAx0DI,8D;MAw0DJ,gD;QAaI,YAr1DgB,cA
AR,iBAAQ,C;QAs1DhB,kBAaKB,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,sBAA
I,KAJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KANBX,C;yGAsBA,yB;MAT1DI,8D;MAS1DJ,
gD;QAaI,YAn2DgB,cAAR,iBAAQ,C;QAo2DhB,kBAaKB,O;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAA
U,KAAV,EAAiB,sBAAI,KAJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KANBX,C;yGAsBA,y
B;MAp2DI,8D;MAo2DJ,gD;QAaI,YAj3DgB,cAAR,iBAAQ,C;Qak3DhB,kBAaKB,O;QACIB,OAAO,SAAS,CAA
hB,C;UACI,cAAc,UAAU,KAAV,EAAiB,sBAAI,KAJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;
O;KANBX,C;uFAsBA,6B;MAOoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QAAM,OAAO,OAAP,
C;;K;uFAG1B,6B;MAOoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QAAM,OAAO,OAAP,C;;K;uF
AG1B,6B;MAOoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QAAM,OAAO,OAAP,C;;K;uFAG1B,6
B;MAOoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QAAM,OAAO,OAAP,C;;K;qGAG1B,6B;MAU
iB,UAAa,M;MAD1B,YAAY,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QAAM,QAAO,cAAP,EAAO,sBAAP
,WAAgB,IAAhB,C;;K;qGAGvB,6B;MAUiB,UAAa,M;MAD1B,YAAY,C;MACC,2B;MAAb,OAAa,cAAb,C;QAA
a,sB;QAAM,QAAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;K;qGAGvB,6B;MAUiB,UAAa,M;MAD1B,YAAY,
C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QAAM,QAAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;K;qGAG
vB,6B;MAUiB,UAAa,M;MAD1B,YAAY,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QAAM,QAAO,cAAP,E
AAO,sBAAP,WAAgB,IAAhB,C;;K;IAGvB,2B;MAKI,OAAO,uB;K;IAGX,2B;MAKI,OAAO,uB;K;IAGX,2B;MA
KI,OAAO,uB;K;IAGX,2B;MAKI,OAAO,uB;K;mFAGX,yB;MA9gEI,8D;MA8gEJ,sC;QAMW,sB;;UAuCP,IAAI,
mBAAJ,C;YAAe,qBAAO,I;YAAP,uB;;UACf,cAAc,sBAAK,CAAL,C;UACd,gBA7jEgB,cAAR,iBAAQ,C;UA8jE
hB,IAAI,cAAa,CAAjB,C;YAAoB,qBAAO,O;YAAP,uB;;UACpB,eA3CmB,QA2CJ,CAAS,OAAT,C;UACf,aAAU,
CAAV,OAAa,SAAb,M;YACI,QAAQ,sBAAK,CAAL,C;YACR,QA9Ce,QA8CP,CAAS,CAAT,C;YACR,IAAI,2B
AAW,CAAX,KAJ,C;cACI,UAAU,C;cACV,WAAW,C;;UAGnB,qBAAO,O;;QApDP,yB;O;KANJ,C;mFASA,y
B;MA/gEI,8D;MA+gEJ,sC;QAMW,sB;;UAuDP,IAAI,mBAAJ,C;YAAe,qBAAO,I;YAAP,uB;;UACf,cAAc,sBAA
K,CAAL,C;UACd,gBA9kEgB,cAAR,iBAAQ,C;UA+kEhB,IAAI,cAAa,CAAjB,C;YAAoB,qBAAO,O;YAAP,uB;;
UACpB,eA3DmB,QA2DJ,CAAS,OAAT,C;UACf,aAAU,CAAV,OAAa,SAAb,M;YACI,QAAQ,sBAAK,CAAL,C;
YACR,QA9De,QA8DP,CAAS,CAAT,C;YACR,IAAI,2BAAW,CAAX,KAJ,C;cACI,UAAU,C;cACV,WAAW,C;
;;UAGnB,qBAAO,O;;QApEP,yB;O;KANJ,C;mFASA,yB;MAhhEI,8D;MAghEJ,sC;QAMW,sB;;UAuEP,IAAI,mB

AAJ,C;YAAe,qBAAO,I;YAAP,uB;;UACf,cAAc,sBAAK,CAAL,C;UACd,gBA/IEgB,cAAR,iBAAQ,C;UAgmEhB,IAAI,cAAa,CAAjB,C;YAAoB,qBAAO,O;YAAP,uB;;UACpB,eA3EmB,QA2EJ,CAAS,OAAT,C;UACf,aAAU,CAAV,OAAa,SAAb,M;YACI,QAAQ,sBAAK,CAAL,C;YACR,QA9Ee,QA8EP,CAAS,CAAT,C;YACR,IAAI,2BAAW,CAAX,KAAJ,C;cACI,UAAU,C;cACV,WAAW,C;;;UAGnB,qBAAO,O;;;QApFP,yB;O;KANJ,C;mFASA,yB;MAjhEI,8D;MAihEJ,sC;QAMW,sB;;UAuFP,IAAI,mBAAJ,C;YAAe,qBAAO,I;YAAP,uB;;UACf,cAAc,sBAAK,CAAL,C;UACd,gBAhnEgB,cAAR,iBAAQ,C;UainEhB,IAAI,cAAa,CAAjB,C;YAAoB,qBAAO,O;YAAP,uB;;UACpB,eA3FmB,QA2FJ,CAAS,OAAT,C;UACf,aAAU,CAAV,OAAa,SAAb,M;YACI,QAAQ,sBAAK,CAAL,C;YACR,QA9Fe,QA8FP,CAAS,CAAT,C;YACR,IAAI,2BAAW,CAAX,KAAJ,C;cACI,UAAU,C;cACV,WAAW,C;;;UAGnB,qBAAO,O;;;QApGP,yB;O;KANJ,C;+FASA,yB;MAIjEI,8D;MAkjEJ,sC;QASI,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,cAAc,sBAAK,CAAL,C;QACd,gBA7jEgB,cA6jEA,SA7jER,QAAQ,C;QA8jEhB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAtBX,C;+FAyBA,yB;MAnkEI,8D;MAMkEJ,sC;QASI,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,cAAc,sBAAK,CAAL,C;QACd,gBA9kEgB,cA8kEA,SA9kER,QAAQ,C;QA+kEhB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAtBX,C;+FAyBA,yB;MAplEI,8D;MAolEJ,sC;QASI,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,cAAc,sBAAK,CAAL,C;QACd,gBA/IEgB,cA+lEA,SA/IER,QAAQ,C;QAgmEhB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAtBX,C;+FAyBA,yB;MApEI,8D;MApEI,sC;QASI,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,cAAc,sBAAK,CAAL,C;QACd,gBAhnEgB,cAgnEA,SAhnER,QAAQ,C;QainEhB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAtBX,C;kFAyBA,yB;MAAA,sE;MATpEI,8D;MpBnwHJ,iB;MoBy5LA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAtqEG,cAAR,iBAAQ,C;QAsqEhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBn6LG,MAAO,KoBm6LO,QpBn6LP,EoBm6LiB,CpBn6LjB,C;;;QoBq6Ld,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MARqEI,8D;MpB3wHJ,iB;MoBg7LA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OArrEG,cAAR,iBAAQ,C;QAqrEhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB17LG,MAAO,KoB07LO,QpB17LP,EoB07LiB,CpB17LjB,C;;;QoB47Ld,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAprEI,8D;MpBnxHJ,iB;MoBu8LA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAsEG,cAAR,iBAAQ,C;QAosEhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBj9LG,MAAO,KoBi9LO,QpBj9LP,EoBi9LiB,CpBj9LjB,C;;;QoBm9Ld,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAnsEI,8D;MpB3xHJ,iB;MoB89LA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAntEG,cAAR,iBAAQ,C;QAmtEhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBx+LG,MAAO,KoBw+LO,QpBx+LP,EoBw+LiB,CpBx+LjB,C;;;QoB0+Ld,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAIvEI,8D;MpB9wHJ,iB;MoBggMA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAlwEG,cAAR,iBAAQ,C;QakwEhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB1gMG,MAAO,KoB0gMO,QpB1gMP,EoB0gMiB,CpB1gMjB,C;;;QoB4gMd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAjwEI,8D;MpBtxHJ,iB;MoBuhMA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAhEG,cAAR,iBAAQ,C;QAgyEhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBxjMG,MAAO,KoBwjMO,QpBxjMP,EoBwjMiB,CpBxjMjB,C;;;QoB0jMd,OAAO,Q;O;KApBX,C;mFAuBA,yB;

MAAA,sE;MA/xEI,8D;MpBtyHJ,iB;MoBqkMA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB ,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA/yEG,cAAR,iBAAQ,C;QA+yEhB,aAAU,CAAV,iB;UACI,QAA Q,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB/kMG,MAAO,KoB+kMO,QpB/kMP,EoB+kMiB,CpB/kMjB,C;;Qo BilMd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MA90EI,8D;MA80EJ,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C ;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA51EG,cAAR,iBAAQ,C;QA41EhB,a AAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WA AAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MA71EI,8D;MA61EJ,sC;QAcIB,Q;QAFb,IAAI, mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA32EG,cAAR,iBAAQ,C;Q A22EhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;Y ACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MA52EI,8D;MA42EJ,sC;QAcIB,Q;QA Fb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA13EG,cAAR,iBA AQ,C;QA03EhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,K AAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MA33EI,8D;MA23EJ,sC;QAc iB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAz4EG,cA AR,iBAAQ,C;QAY4EhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,C AAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;8FAuBA,yB;MA16EI,8D;MpBnwHJ,iB;MoB6q MA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QAcTb,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,O Ax7EG,cAAR,iBAAQ,C;QAw7EhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBr rMG,MAAO,KoBqrMO,QpBrrMP,EoBqrMiB,CpBrrMjB,C;;QoBurMd,OAAO,Q;O;KAIBX,C;+FAqBA,yB;MAv7 EI,8D;MpB3wHJ,iB;MoBksMA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QAcTb,eAAe,SAAS,sBAA K,CAAL,CAAT,C;QACF,OA8EG,cAAR,iBAAQ,C;QAq8EhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,C AAL,CAAT,C;UACR,WpB1sMG,MAAO,KoB0sMO,QpB1sMP,EoB0sMiB,CpB1sMjB,C;;QoB4sMd,OAAO,Q;O ;KAIBX,C;+FAqBA,yB;MAp8EI,8D;MpBnxHJ,iB;MoButMA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO, I;QAcTb,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA19EG,cAAR,iBAAQ,C;Qak9EhB,aAAU,CAAV,iB;UA CI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB/tMG,MAAO,KoB+tMO,QpB/tMP,EoB+tMiB,CpB/tMjB, C;;QoBiuMd,OAAO,Q;O;KAIBX,C;+FAqBA,yB;MAj9EI,8D;MpB3xHJ,iB;MoB4uMA,sC;QAcIB,Q;QAFb,IAAI, mBAAJ,C;UAAe,OAAO,I;QAcTb,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA/9EG,cAAR,iBAAQ,C;QA+9 EhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBpvMG,MAAO,KoBovMO,QpBp vMP,EoBovMiB,CpBpvMjB,C;;QoBsvMd,OAAO,Q;O;KAIBX,C;+FAqBA,yB;MA9/EI,8D;MpB9wHJ,iB;MoB4w MA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QAcTb,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,O A5gFG,cAAR,iBAAQ,C;QA4gFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBp xMG,MAAO,KoBoxMO,QpBpxMP,EoBoxMiB,CpBpxMjB,C;;QoBsxMd,OAAO,Q;O;KAIBX,C;+FAqBA,yB;M A3gFI,8D;MpBtxHJ,iB;MoBiyMA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QAcTb,eAAe,SAAS,sBA AK,CAAL,CAAT,C;QACF,OAzhFG,cAAR,iBAAQ,C;QAYhFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK, CAAL,CAAT,C;UACR,WpBzyMG,MAAO,KoByyMO,QpBzyMP,EoByyMiB,CpBzyMjB,C;;QoB2yMd,OAAO,Q ;O;KAIBX,C;+FAqBA,yB;MAxhFI,8D;MpB9xHJ,iB;MoBszMA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAA O,I;QAcTb,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAtiFG,cAAR,iBAAQ,C;QAsiFhB,aAAU,CAAV,iB;U ACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB9zMG,MAAO,KoB8zMO,QpB9zMP,EoB8zMiB,CpB9z MjB,C;;QoBg0Md,OAAO,Q;O;KAIBX,C;+FAqBA,yB;MAriFI,8D;MpBtyHJ,iB;MoB20MA,sC;QAcIB,Q;QAFb,I AAI,mBAAJ,C;UAAe,OAAO,I;QAcTb,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAanjFG,cAAR,iBAAQ,C;Q AmjFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBn1MG,MAAO,KoBm1MO, QpBn1MP,EoBm1MiB,CpBn1MjB,C;;QoBq1Md,OAAO,Q;O;KAIBX,C;+FAqBA,yB;MAlFI,8D;MAklFI,sC;QA YiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QAcTb,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA9IFG,cAA R,iBAAQ,C;QA8IFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CA AX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAIBX,C;+FAqBA,yB;MA/IFI,8D;MA+IFJ,sC;QAYiB,Q;Q AFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QAcTb,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA3mFG,cAAR,iBA AQ,C;QA2mFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,K AAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAIBX,C;+FAqBA,yB;MA5mFI,8D;MA4mFJ,sC;QAYiB,Q;QAF

b,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAxnFG,cAAR,iBAAQ,C;QAwnFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAJ ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAIBX,C;+FAqBA,yB;MAznFI,8D;MAynFJ,sC;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAroFG,cAAR,iBAAQ,C;QAq oFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAJ,C;YAC I,WAAW,C;;;QAGnB,OAAO,Q;O;KAIBX,C;2FAqBA,yB;MAAA,sE;MATqFI,8D;MASqFJ,kD;QACiB,Q;QAFb,IA AI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAprFG,cAAR,iBAAQ,C; QAorFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EA AkB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;0FAuBA,yB;MAAA,s E;MArrFI,8D;MAqrFJ,kD;QACiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAA L,CAAT,C;QACF,OAnsFG,cAAR,iBAAQ,C;QAmsFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,C AAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGn B,OAAO,Q;O;KApBX,C;2FAuBA,yB;MAAA,sE;MApsFI,8D;MAosFJ,kD;QACiB,Q;QAFb,IAAI,mBAAJ,C;UAA e,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAltFG,cAAR,iBAAQ,C;QaktFhB,aAAU,CA AV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX, GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;2FAuBA,yB;MAAA,sE;MAntFI,8D;Mamt FJ,kD;QACiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF, OajuFG,cAAR,iBAAQ,C;QAiuFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI, UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApB X,C;uGAuBA,yB;MALwFI,8D;MAkwFJ,kD;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SA A S,sBAAK,CAAL,CAAT,C;QACF,OA9wFG,cAAR,iBAAQ,C;QA8wFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,s BAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,W AAW,C;;;QAGnB,OAAO,Q;O;KAIBX,C;sGAqBA,yB;MA/wFI,8D;MA+wFJ,kD;QAYiB,Q;QAFb,IAAI,mBAAJ, C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA3xFG,cAAR,iBAAQ,C;QA2xFhB,aA AU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB, CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAIBX,C;uGAqBA,yB;MA5xFI,8D;MA4xFJ, kD;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,Oaxy FG,cAAR,iBAAQ,C;QAwyFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UA AW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAIBX,C; uGAqBA,yB;MAzyFI,8D;MAyyFJ,kD;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sB AAK,CAAL,CAAT,C;QACF,OArzFG,cAAR,iBAAQ,C;QAqzFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK, CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C; ;QAGnB,OAAO,Q;O;KAIBX,C;IAqBA,iC;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAe,OAAO,I;MActB,UAAU,sB AAK,CAAL,C;MACG,OA91FG,gBAAR,iBAAQ,C;MA81FhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C; QACR,InC5mN8D,YmC4mN1D,GnC5mN2E,KAAjB,EmC4mNpD,CnC5mNiF,KAA7B,CmC4mN1D,IAAJ,C;UA Aa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,iC;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAe,OAAO,I;MActB,UAAU,s BAAK,CAAL,C;MACG,OAr2FG,gBAAR,iBAAQ,C;MAq2FhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C ;QACR,InBnnN+D,amBmnN3D,GnBnnN6E,KAAIB,EmBmnNrD,CnBnnNmF,KAA9B,CmBmnN3D,IAAJ,C;UA Aa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,iC;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAe,OAAO,I;MActB,UAAU,s BAAK,CAAL,C;MACG,OA52FG,gBAAR,iBAAQ,C;MA42FhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C ;QACR,IpC1pN4E,0BoC0pNx,E,GpC/6M8B,KAAL,GAAiB,GA3O8B,EoC0pNIE,CpC/6MwB,KAAL,GAAiB,GA 3O8B,CoC0pNx,E,IAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,iC;MAQiB,Q;MAFb,IAAI,mBAAJ,C;Q AAe,OAAO,I;MActB,UAAU,sBAAK,CAAL,C;MACG,OAn3FG,gBAAR,iBAAQ,C;MAm3FhB,aAAU,CAAV,iB ;QACI,QAAQ,sBAAK,CAAL,C;QACR,IICjqN6E,0BkCiqNzE,GIC77M8B,KAAL,GAAiB,KApO+B,EkCiqNnE,C IC77MwB,KAAL,GAAiB,KApO+B,CkCiqNzE,IAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,2C;MAKI, OAAO,4BAAc,UAAAd,C;K;IAGX,2C;MAKI,OAAO,4BAAc,UAAAd,C;K;IAGX,2C;MAKI,OAAO,4BAAc,UAAAd, C;K;IAGX,2C;MAKI,OAAO,4BAAc,UAAAd,C;K;IAGX,iD;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAe,OAAO,I;M ActB,UAAU,sBAAK,CAAL,C;MACG,OA17FG,gBAAR,iBAAQ,C;MA07FhB,aAAU,CAAV,iB;QACI,QAAQ,s

BAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;
MAE9C,OAAO,G;K;IAGX,iD;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MAcTb,UAAU,sBAAK,CAAL,
C;MACG,OAj8FG,gBAAR,iBAAQ,C;MAi8FhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IAAI,U
AAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,iD;
MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MAcTb,UAAU,sBAAK,CAAL,C;MACG,OAx8FG,gBAAR,iB
AAQ,C;MAw8FhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,C
AAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,iD;MAQiB,Q;MAFb,IAAI,mB
AAJ,C;QAAe,OAAO,I;MAcTb,UAAU,sBAAK,CAAL,C;MACG,OA/8FG,gBAAR,iBAAQ,C;MA+8FhB,aAAU,CA
AV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,
C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,2B;MAKI,OAAO,uB;K;IAGX,2B;MAKI,OAAO,uB;K;IAGX,2
B;MAKI,OAAO,uB;K;IAGX,2B;MAKI,OAAO,uB;K;mFAGX,yB;MA9gGI,8D;MA8gGJ,sC;QAMW,sB;;UAuCP,
IAAI,mBAAJ,C;YAAe,qBAAO,I;YAAP,uB;;UACf,cAAc,sBAAK,CAAL,C;UACd,gBA7jGgB,cAAR,iBAAQ,C;U
A8jGhB,IAAI,cAAa,CAAjB,C;YAAoB,qBAAO,O;YAAP,uB;;UACpB,eA3CmB,QA2CJ,CAAS,OAAT,C;UACf,a
AAU,CAAV,OAAa,SAAb,M;YACI,QAAQ,sBAAK,CAAL,C;YACR,QA9Ce,QA8CP,CAAS,CAAT,C;YACR,IA
AI,2BAAW,CAAX,KAAJ,C;cACI,UAAU,C;cACV,WAAW,C;;UAGnB,qBAAO,O;;QApDP,yB;O;KANJ,C;mFA
SA,yB;MA/gGI,8D;MA+gGJ,sC;QAMW,sB;;UAuDP,IAAI,mBAAJ,C;YAAe,qBAAO,I;YAAP,uB;;UACf,cAAc,s
BAAK,CAAL,C;UACd,gBA9kGgB,cAAR,iBAAQ,C;UA+kGhB,IAAI,cAAa,CAAjB,C;YAAoB,qBAAO,O;YAAP
,uB;;UACpB,eA3DmB,QA2DJ,CAAS,OAAT,C;UACf,aAAU,CAAV,OAAa,SAAb,M;YACI,QAAQ,sBAAK,CAA
L,C;YACR,QA9De,QA8DP,CAAS,CAAT,C;YACR,IAAI,2BAAW,CAAX,KAAJ,C;cACI,UAAU,C;cACV,WAA
W,C;;UAGnB,qBAAO,O;;QApEP,yB;O;KANJ,C;mFASA,yB;MAhhGI,8D;MAghGJ,sC;QAMW,sB;;UAuEP,IA
AI,mBAAJ,C;YAAe,qBAAO,I;YAAP,uB;;UACf,cAAc,sBAAK,CAAL,C;UACd,gBA/IGgB,cAAR,iBAAQ,C;UAg
mGhB,IAAI,cAAa,CAAjB,C;YAAoB,qBAAO,O;YAAP,uB;;UACpB,eA3EmB,QA2EJ,CAAS,OAAT,C;UACf,aA
AU,CAAV,OAAa,SAAb,M;YACI,QAAQ,sBAAK,CAAL,C;YACR,QA9Ee,QA8EP,CAAS,CAAT,C;YACR,IAAI,
2BAAW,CAAX,KAAJ,C;cACI,UAAU,C;cACV,WAAW,C;;UAGnB,qBAAO,O;;QApFP,yB;O;KANJ,C;mFASA
,yB;MAjhGI,8D;MAihGJ,sC;QAMW,sB;;UAuFP,IAAI,mBAAJ,C;YAAe,qBAAO,I;YAAP,uB;;UACf,cAAc,sBAA
K,CAAL,C;UACd,gBAhnGgB,cAAR,iBAAQ,C;UAinGhB,IAAI,cAAa,CAAjB,C;YAAoB,qBAAO,O;YAAP,uB;;
UACpB,eA3FmB,QA2FJ,CAAS,OAAT,C;UACf,aAAU,CAAV,OAAa,SAAb,M;YACI,QAAQ,sBAAK,CAAL,C;
YACR,QA9Fe,QA8FP,CAAS,CAAT,C;YACR,IAAI,2BAAW,CAAX,KAAJ,C;cACI,UAAU,C;cACV,WAAW,C;;
UAGnB,qBAAO,O;;QApGP,yB;O;KANJ,C;+FASA,yB;MAIjGI,8D;MAkjGJ,sC;QASI,IAAI,mBAAJ,C;UAAe,O
AAO,I;QAcTb,cAAc,sBAAK,CAAL,C;QACd,gBA7jGgB,cA6jGA,SA7jGR,QAAQ,C;QA8jGhB,IAAI,cAAa,CA
AjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBA
AK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAA
W,C;;QAGnB,OAAO,O;O;KAtBX,C;+FAyBA,yB;MAnkGI,8D;MAmkGJ,sC;QASI,IAAI,mBAAJ,C;UAAe,OAA
O,I;QAcTb,cAAc,sBAAK,CAAL,C;QACd,gBA9kGgB,cA8kGA,SA9kGR,QAAQ,C;QA+kGhB,IAAI,cAAa,CAAj
B,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAA
K,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAA
W,C;;QAGnB,OAAO,O;O;KAtBX,C;+FAyBA,yB;MAplGI,8D;MAolGJ,sC;QASI,IAAI,mBAAJ,C;UAAe,OAAO
,I;QAcTb,cAAc,sBAAK,CAAL,C;QACd,gBA/IGgB,cA+IGA,SA/IGR,QAAQ,C;QAgmGhB,IAAI,cAAa,CAAjB,C
;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAAK,C
AAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;
;;QAGnB,OAAO,O;O;KAtBX,C;+FAyBA,yB;MArmGI,8D;MAqmGJ,sC;QASI,IAAI,mBAAJ,C;UAAe,OAAO,I;
QAcTb,cAAc,sBAAK,CAAL,C;QACd,gBAhnGgB,cAgnGA,SAhnGR,QAAQ,C;QAinGhB,IAAI,cAAa,CAAjB,C;
UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAAK,CA
AL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;
QAGnB,OAAO,O;O;KAtBX,C;kFAyBA,yB;MAAA,sE;MAtpGI,8D;MpB/iHJ,iB;MoBqsNA,sC;QAgBiB,Q;QAFb
,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA tqGG,cAAR,iBAAQ
,C;QAsqGhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB/sNG,MAAO,KoB+sN
O,QpB/sNP,EoB+sNiB,CpB/sNjB,C;;QoBitNd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MARqGI,8D;MpB

vjHJ,iB;MoB4tNA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OArrGG,cAAR,iBAAQ,C;QAqrGhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBtuNG,MAAO,KoBsuNO,QpBtuNP,EoBsuNiB,CpBtuNjB,C;;QoBwuNd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAprGI,8D;MpB/jHJ,iB;MoBmvNA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OapsGG,cAAR,iBAAQ,C;QAosGhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB7vNG,MAAO,KoB6vNO,QpB7vNP,EoB6vNiB,CpB7vNjB,C;;QoB+vNd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAAnsGI,8D;MpBvkHJ,iB;MoB0wNA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAntGG,cAAR,iBAAQ,C;QAmtGhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBpxNG,MAAO,KoBoxNO,QpBpxNP,EoBoxNiB,CpBpxNjB,C;;QoBsxNd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAIvGI,8D;MpB1jHJ,iB;MoB4yNA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAlwGG,cAAR,iBAAQ,C;QAKwGhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBtzNG,MAAO,KoBszNO,QpBtzNP,EoBszNiB,CpBtzNjB,C;;QoBwzNd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAjwGI,8D;MpBlkHJ,iB;MoBm0NA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAJxGG,cAAR,iBAAQ,C;QAixGhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB70NG,MAAO,KoB60NO,QpB70NP,EoB60NiB,CpB70NjB,C;;QoB+0Nd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAhxGI,8D;MpB1kHJ,iB;MoB01NA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAhYGG,cAAR,iBAAQ,C;QAgYghB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBp2NG,MAAO,KoBo2NO,QpBp2NP,EoBo2NiB,CpBp2NjB,C;;QoBs2Nd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MA/xGI,8D;MpBlIHJ,iB;MoBi3NA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA/yGG,cAAR,iBAAQ,C;QA+yGhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB33NG,MAAO,KoB23NO,QpB33NP,EoB23NiB,CpB33NjB,C;;QoB63Nd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MA90GI,8D;MA80GJ,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA51GG,cAAR,iBAAQ,C;QA41GhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MA71GI,8D;MA61GJ,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA32GG,cAAR,iBAAQ,C;QA22GhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MA52GI,8D;MA42GJ,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA13GG,cAAR,iBAAQ,C;QA03GhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MA33GI,8D;MA23GJ,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAz4GG,cAAR,iBAAQ,C;QAY4GhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;8FAuBA,yB;MA16GI,8D;MpB/iHJ,iB;MoBy9NA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACTB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAx7GG,cAAR,iBAAQ,C;QAw7GhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBj+NG,MAAO,KoBi+NO,QpBj+NP,EoBi+NiB,CpBj+NjB,C;;QoBm+Nd,OAAO,Q;O;KAIBX,C;+FAqBA,yB;MAv7GI,8D;MpBvjHJ,iB;MoB8+NA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACTB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAr8GG,cAAR,iBAAQ,C;QAq8GhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBt/NG,MAAO,KoBs/NO,QpBt/NP,EoBs/NiB,CpBt/NjB,C;;QoBw/Nd,OAAO,Q;O;KAIBX,C;+FAqBA,yB;MAp8GI,8D;MpB/jHJ,iB;MoBmgOA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACTB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAI9GG,cAAR,iBAAQ,C;QAK9GhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB3gOG,MAAO,KoB2gOO,QpB3gOP,EoB2gOiB,CpB3gOjB,C;;QoB6gOd,OAAO,Q;O;KAIBX,C;+FAqBA,yB;MAj9GI,8D;MpBvkHJ,iB;MoBwhOA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACTB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA/9GG,cAAR,iBAAQ,C;QA+9GhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBhiOG,MAAO,KoBgiO

O,QpBhiOP,EoBgiOiB,CpBhiOjB,C;;QoBkiOd,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MA9/GI,8D;MpB1jHJ,iB;MoBwjOA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA5gHG,cAAR,iBAAQ,C;QA4gHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBhkOG,MAAO,KoBgkOO,QpBhkOP,EoBgkOiB,CpBhkOjB,C;;QoBkkOd,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MA3gHI,8D;MpBlkHJ,iB;MoB6kOA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAzhHG,cAAR,iBAAQ,C;QAYhHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBrlOG,MAAO,KoBqlOO,QpBrlOP,EoBqlOiB,CpBrlOjB,C;;QoBulOd,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MAxhHI,8D;MpB1kHJ,iB;MoBkmOA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAtiHG,cAAR,iBAAQ,C;QAsiHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB1mOG,MAAO,KoB0mOO,QpB1mOP,EoB0mOiB,CpB1mOjB,C;;QoB4mOd,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MAriHI,8D;MpBlIHJ,iB;MoBunOA,sC;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA9IHG,cAAR,iBAAQ,C;QAmjHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB/nOG,MAAO,KoB+nOO,QpB/nOP,EoB+nOiB,CpB/nOjB,C;;QoBioOd,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MAIHI,8D;MAkiHJ,sC;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA9IHG,cAAR,iBAAQ,C;QA8lHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MA/IHI,8D;MA+IHJ,sC;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA3mHG,cAAR,iBAAQ,C;QA2mHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MA5mHI,8D;MA4mHJ,sC;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAxnHG,cAAR,iBAAQ,C;QAWNhHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MAznHI,8D;MAynHJ,sC;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAroHG,cAAR,iBAAQ,C;QAqoHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;2FAqBA,yB;MAAA,sE;MAAtqHI,8D;MASqHJ,kD;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAprHG,cAAR,iBAAQ,C;QAorHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;0FAuBA,yB;MAAA,sE;MArrHI,8D;MAqrHJ,kD;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAAnsHG,cAAR,iBAAQ,C;QAmsHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;2FAuBA,yB;MAAA,sE;MApsHI,8D;MAosHJ,kD;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAItHG,cAAR,iBAAQ,C;QAktHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;2FAuBA,yB;MAAA,sE;MAntHI,8D;MAmtHJ,kD;QAcIB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAjuHG,cAAR,iBAAQ,C;QAiuHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;uGAuBA,yB;MALwHI,8D;MAkwHJ,kD;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA9wHG,cAAR,iBAAQ,C;QA8wHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;sGAqBA,yB;MA/wHI,8D;MA+wHJ,kD;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA3xHG,cAAR,iBAAQ,C;QA2xHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAakB,CAAIB,CAAX,GAakC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;uGAqBA,yB;MA5xHI,8D;MA4xHJ,kD;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAxyHG,cAAR,iBAAQ,C;QAwyHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,

IAAI,UAAW,SAAQ,QAAR,EAAkB,CAAIB,CAAX,GAaKc,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;K
AlBX,C;uGAqBA,yB;MAzyHI,8D;MAyyHJ,kD;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,
SAAS,sBAAK,CAAL,CAAT,C;QACF,OArzHG,cAAR,iBAAQ,C;QAqzHhB,aAAU,CAAV,iB;UACI,QAAQ,SAA
S,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAAkB,CAAIB,CAAX,GAaKc,CAAtC,C;YACI,
WAAW,C;;;QAGnB,OAAO,Q;O;KAIBX,C;IAqBA,iC;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MACTB
,UAAU,sBAAK,CAAL,C;MACG,OA9IHG,gBAAR,iBAAQ,C;MA8IHhB,aAAU,CAAV,iB;QACI,QAAQ,sBAA
K,CAAL,C;QACR,InC5mP8D,YmC4mP1D,GnC5mP2E,KAAjB,EmC4mPpD,CnC5mPiF,KAA7B,CmC4mP1D,I
AAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,iC;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MACT
B,UAAU,sBAAK,CAAL,C;MACG,OAr2HG,gBAAR,iBAAQ,C;MAq2HhB,aAAU,CAAV,iB;QACI,QAAQ,sBAA
K,CAAL,C;QACR,InBnnP+D,amBnnP3D,GnBnnP6E,KAAIB,EmBnnPrD,CnBnnPmF,KAA9B,CmBnnP3D,IA
AJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,iC;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MACTB
,UAAU,sBAAK,CAAL,C;MACG,OA52HG,gBAAR,iBAAQ,C;MA42HhB,aAAU,CAAV,iB;QACI,QAAQ,sBAA
K,CAAL,C;QACR,IpC1pP4E,0BoC0pPxE,GpC/6O8B,KAAL,GAAiB,GA3O8B,EoC0pPIE,CpC/6OwB,KAAL,GA
AiB,GA3O8B,CoC0pPxE,IAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,iC;MAQiB,Q;MAFb,IAAI,mBA
AJ,C;QAAe,OAAO,I;MACTB,UAAU,sBAAK,CAAL,C;MACG,OAn3HG,gBAAR,iBAAQ,C;MAM3HhB,aAAU,C
AAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IICjqP6E,0BkCiqPzE,GlC77O8B,KAAL,GAAiB,KApO+B,EkCiq
PnE,CIC77OwB,KAAL,GAAiB,KApO+B,CkCiqPzE,IAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,2C;M
AKI,OAAO,4BAAc,UAAAd,C;K;IAGX,2C;MAKI,OAAO,4BAAc,UAAAd,C;K;IAGX,2C;MAKI,OAAO,4BAAc,UA
Ad,C;K;IAGX,2C;MAKI,OAAO,4BAAc,UAAAd,C;K;IAGX,iD;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;
MACTB,UAAU,sBAAK,CAAL,C;MACG,OA17HG,gBAAR,iBAAQ,C;MA07HhB,aAAU,CAAV,iB;QACI,QAAQ
,sBAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C
;;MAE9C,OAAO,G;K;IAGX,iD;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MACTB,UAAU,sBAAK,CAA
L,C;MACG,OAj8HG,gBAAR,iBAAQ,C;MAi8HhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IAA
I,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,i
D;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MACTB,UAAU,sBAAK,CAAL,C;MACG,OAx8HG,gBAAR
,iBAAQ,C;MAw8HhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAA
a,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,iD;MAQiB,Q;MAFb,IAAI,m
BAAJ,C;QAAe,OAAO,I;MACTB,UAAU,sBAAK,CAAL,C;MACG,OA/8HG,gBAAR,iBAAQ,C;MA+8HhB,aAAU
,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CA
AjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;iFAGX,qB;MASI,OAAO,mB;K;iFAGX,qB;MASI,OAAO,mB;K;
iFAGX,qB;MASI,OAAO,mB;K;iFAGX,qB;MASI,OAAO,mB;K;iFAGX,gC;MASoB,Q;MAAA,2B;MAAhB,OAA
gB,cAAhB,C;QAAGB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,K;;MACrD,OAAO,I;K;iFAGX,g
C;MASoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,
OAAO,K;;MACrD,OAAO,I;K;iFAGX,gC;MASoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QAAM,
IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,K;;MACrD,OAAO,I;K;iFAGX,gC;MASoB,Q;MAAA,2B;MAAhB,
OAAgB,cAAhB,C;QAAGB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,K;;MACrD,OAAO,I;K;qF
AGX,6B;MAOmC,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QAAM,OAAO,OAAP,C;;MAArC,gB;K
;qFAGJ,6B;MAOmC,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QAAM,OAAO,OAAP,C;;MAArC,gB
;K;qFAGJ,6B;MAOmC,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QAAM,OAAO,OAAP,C;;MAArC,
gB;K;qFAGJ,6B;MAOmC,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAGB,yB;QAAM,OAAO,OAAP,C;;MAAr
C,gB;K;mGAGJ,6B;MAtrEiB,gB;MADb,YAAY,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QAAM,QAAO,c
AAP,EAAO,sBAAP,WAAgB,IAAhB,C;;MAGsEnB,gB;K;mGAGJ,6B;MAtrEiB,gB;MADb,YAAY,C;MACC,2B;
MAAb,OAAa,cAAb,C;QAAa,sB;QAAM,QAAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;MAGsEnB,gB;K;mGA
GJ,6B;MAtrEiB,gB;MADb,YAAY,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QAAM,QAAO,cAAP,EAAO,s
BAAP,WAAgB,IAAhB,C;;MAGsEnB,gB;K;mGAGJ,6B;MAtrEiB,gB;MADb,YAAY,C;MACC,2B;MAAb,OAAa,
cAAb,C;QAAa,sB;QAAM,QAAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;MAGsEnB,gB;K;qFAGJ,yB;MAAA,
4F;MA9qII,8D;MA8qIJ,uC;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kB
AAkB,sBAAK,CAAL,C;QACD,OAjsID,cAAR,iBAAQ,C;QAisIhB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV

,EAAuB,sBAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAtBX,C;qFAyBA,yB;MAAA,4F;MA+rII,8D;MA+rIJ,u C;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAaKB,sBAAK,CAAL,C; QACD,OAltID,cAAR,iBAAQ,C;QAKtIhB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,sBAAK,KAAL, CAA9B,C;;QAEIB,OAAO,W;O;KAtBX,C;qFAyBA,yB;MAAA,4F;MAhtII,8D;MAgtIJ,uC;QAmBqB,Q;QAHjB,I AAI,mBAAJ,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAaKB,sBAAK,CAAL,C;QACD,OAnuID,cAAR,i BAAQ,C;QAmulhB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,sBAAK,KAAL,CAA9B,C;;QAEIB,O AAO,W;O;KAtBX,C;qFAyBA,yB;MAAA,4F;MAjuII,8D;MAiuIJ,uC;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI, MAAM,mCAA8B,+BAA9B,C;QACV,kBAaKB,sBAAK,CAAL,C;QACD,OApvID,cAAR,iBAAQ,C;QAovIhB,iB AAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,sBAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAtBX,C;m GAYBA,yB;MAAA,4F;MALxII,8D;MAKxIJ,uC;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,MAAM,mCAA8B,+B AA9B,C;QACV,kBAaKB,sBAAK,CAAL,C;QACD,OArYID,cAAR,iBAAQ,C;QAqyIhB,iBAAc,CAAd,yB;UACI, cAAc,UAAU,KAaV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAtBX,C;mGAYB A,yB;MAAA,4F;MAnyII,8D;MAmyIJ,uC;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,MAAM,mCAA8B,+BAA9 B,C;QACV,kBAaKB,sBAAK,CAAL,C;QACD,OAtzID,cAAR,iBAAQ,C;QAszIhB,iBAAc,CAAd,yB;UACI,cAAc, UAAU,KAaV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAtBX,C;mGAYBA,yB; MAAA,4F;MApzII,8D;MAozIJ,uC;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,MAAM,mCAA8B,+BAA9B,C;Q ACV,kBAaKB,sBAAK,CAAL,C;QACD,OAv0ID,cAAR,iBAAQ,C;QAu0IhB,iBAAc,CAAd,yB;UACI,cAAc,UAA U,KAaV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAtBX,C;mGAYBA,yB;MAA A,4F;MAr0II,8D;MAq0IJ,uC;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,k BAaKB,sBAAK,CAAL,C;QACD,OAx1ID,cAAR,iBAAQ,C;QAw1IhB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KA AV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAtBX,C;+GAYBA,yB;MAT3II,8D; MAS3IJ,uC;QAKbqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,OAAO,I;QACX,kBAaKB,sBAAK,CAAL,C;QACD,OAx4 ID,cAAR,iBAAQ,C;QAw4IhB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAaV,EAAiB,WAAjB,EAA8B,sBAAK,K AAL,CAA9B,C;;QAEIB,OAAO,W;O;KArBX,C;+GAwBA,yB;MAT4II,8D;MAS4IJ,uC;QAKbqB,Q;QAHjB,IAAI, mBAAJ,C;UACI,OAAO,I;QACX,kBAaKB,sBAAK,CAAL,C;QACD,OAx5ID,cAAR,iBAAQ,C;QAw5IhB,iBAAc ,CAAd,yB;UACI,cAAc,UAAU,KAaV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O; KArBX,C;+GAwBA,yB;MAT5II,8D;MAS5IJ,uC;QAKbqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,OAAO,I;QACX,kBA AkB,sBAAK,CAAL,C;QACD,OAx6ID,cAAR,iBAAQ,C;QAw6IhB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAaV ,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KArBX,C;+GAwBA,yB;MAT6II,8D;M As6IJ,uC;QAKbqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,OAAO,I;QACX,kBAaKB,sBAAK,CAAL,C;QACD,OAx7ID ,cAAR,iBAAQ,C;QAw7IhB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAaV,EAAiB,WAAjB,EAA8B,sBAAK,KAA L,CAA9B,C;;QAEIB,OAAO,W;O;KArBX,C;iGAwBA,yB;MAT9II,8D;MAS9IJ,uC;QAmBqB,Q;QAHjB,IAAI,mB AAJ,C;UACI,OAAO,I;QACX,kBAaKB,sBAAK,CAAL,C;QACD,OAz+ID,cAAR,iBAAQ,C;QAY+IhB,iBAAc,CA Ad,yB;UACI,cAAc,UAAU,WAAV,EAAuB,sBAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAtBX,C;iGAYBA,y B;MAv+II,8D;MAu+IJ,uC;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,OAAO,I;QACX,kBAaKB,sBAAK,CAAL, C;QACD,OAI/ID,cAAR,iBAAQ,C;QA0/IhB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,sBAAK,KA AL,CAA9B,C;;QAEIB,OAAO,W;O;KAtBX,C;iGAYBA,yB;MAx/II,8D;MAw/IJ,uC;QAmBqB,Q;QAHjB,IAAI,m BAAJ,C;UACI,OAAO,I;QACX,kBAaKB,sBAAK,CAAL,C;QACD,OA3gJD,cAAR,iBAAQ,C;QA2gJhB,iBAAc,C AAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,sBAAK,KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAtBX,C;iGAYBA ,yB;MAzgJI,8D;MAygJJ,uC;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,OAAO,I;QACX,kBAaKB,sBAAK,CAA L,C;QACD,OA5hJD,cAAR,iBAAQ,C;QA4hJhB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,sBAAK, KAAL,CAA9B,C;;QAEIB,OAAO,W;O;KAtBX,C;+FAyBA,yB;MAAA,4F;MA1jJI,8D;MA0jJJ,uC;QAKB0B,UAE U,M;QAJhC,YA1kJgB,cAAR,iBAAQ,C;QA2kJhB,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C; QACrB,kBAaKB,uBAAI,YAAJ,EAAI,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,uBAAI,cA AJ,EAAI,sBAAJ,UAAV,EAAwB,WAAxB,C;;QAEIB,OAAO,W;O;KAtBX,C;+FAyBA,yB;MAAA,4F;MA3kJI,8D ;MA2kJJ,uC;QAKB0B,UAEU,M;QAJhC,YA3lJgB,cAAR,iBAAQ,C;QA4lJhB,IAAI,QAAQ,CAAZ,C;UAAe,MAA M,mCAA8B,+BAA9B,C;QACrB,kBAaKB,uBAAI,YAAJ,EAAI,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UAC I,cAAc,UAAU,uBAAI,cAAJ,EAAI,sBAAJ,UAAV,EAAwB,WAAxB,C;;QAEIB,OAAO,W;O;KAtBX,C;+FAyBA,

yB;MAAA,4F;MA51JI,8D;MA4IJJ,uC;QAKB0B,UAEU,M;QAJhC,YA5mJgB,cAAR,iBAAQ,C;QA6mJhB,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAaKB,uBAAI,YAAJ,EAAL,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,uBAAL,cAAJ,EAAL,sBAAJ,UAAV,EAawB,WAAxB,C;;QAEIB,OA AO,W;O;KAtBX,C;+FAyBA,yB;MAAA,4F;MA7mJI,8D;MA6mJJ,uC;QAKB0B,UAEU,M;QAJhC,YA7nJgB,cAAR,iBAAQ,C;QA8nJhB,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAaKB,uBAAI,YAAJ,EAAL,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,uBAAL,cAAJ,EAAL,sBAAJ,UAAV,EA AwB,WAAxB,C;;QAEIB,OAAO,W;O;KAtBX,C;6GAyBA,yB;MAAA,4F;MA9pJI,8D;MA8pJJ,uC;QAKB0B,Q;QAFtB,YA9qJgB,cAAR,iBAAQ,C;QA+qJhB,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAaKB,uBAAI,YAAJ,EAAL,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAL,sBAAI,KAAJ,CAAjB,EAAL6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAvBX,C;6GA0BA,yB;MAAA,4F;MAhrJ I,8D;MAgrJJ,uC;QAKB0B,Q;QAFtB,YAhsJgB,cAAR,iBAAQ,C;QAisJhB,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAaKB,uBAAI,YAAJ,EAAL,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAL,sBAAI,KAAJ,CAAjB,EAAL6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAvBX,C;6 GA0BA,yB;MAAA,4F;MAIsJI,8D;MAksJJ,uC;QAKB0B,Q;QAFtB,YAltJgB,cAAR,iBAAQ,C;QamtJhB,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAaKB,uBAAI,YAAJ,EAAL,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAL,sBAAI,KAAJ,CAAjB,EAAL6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAvBX,C;6 GA0BA,yB;MAAA,4F;MAptJI,8D;MAotJJ,uC;QAKB0B,Q;QAFtB,YApuJgB,cAAR,iBAAQ,C;QAquJhB,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAaKB,uBAAI,YAAJ,EAAL,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAL,sBAAI,KAAJ,CAAjB,EAAL6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAvBX,C;yHA0BA,yB;MATwJI,8D;MASwJJ,uC;QAIb0B,Q;QAF tB,YArxJgB,cAAR,iBAAQ,C;QAsxJhB,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAaKB,uBAAI,YAAJ,EAAL,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAL,sBAAI,KAAJ,CAAjB,EAAL6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAtBX,C;yHAyBA,yB;MAvxJI,8D;MAuxJJ,uC;QAIb0B,Q;QAFtB,YA tyJgB,cAAR,iBAAQ,C;QAuyJhB,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAaKB,uBAAI,YAAJ,EAAL,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAL,sBAAI,KAAJ,CAAjB,EAAL6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAtBX,C;yHAyBA,yB;MAzJI,8D;MAyzJJ,uC;QAIb0B,Q;QAFtB,YAx0 JgB,cAAR,iBAAQ,C;QAY0JhB,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAaKB,uBAAI,YAAJ,EAAL,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAL,sBAAI,KAAJ,CAAjB,EAAL6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAtBX,C;2GAyBA,yB;MA12JI,8D;MA02JJ,uC;QAKB0B,UAEU,M;QAJhC,YA13JgB,cAAR,iBAAQ,C;QA23JhB,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAaKB,uBAAI,YAAJ,EAAL,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,uBAAL,cAAJ,EAAL,sBAAJ,UAAV,EAawB,WAAxB,C;;QAEIB,OAAO,W;O;KAtBX,C;2GAyBA,yB;MA33JI,8D;MA23JJ,uC;QAKB0B,UAEU,M;QAJhC,YA34JgB,cAAR,iBAAQ,C;QA44JhB,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAaKB,uBAAI,YAAJ,EAAL,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,uBAAL,cAAJ,EAAL,sBAAJ,UAAV,EAawB,WAAxB,C;;QAEIB,OAAO,W;O;KAtBX,C;2GAyBA,yB;MA54JI,8D;MA44JJ,uC;QAKB0B,UAEU,M;QAJhC,YA55JgB,cAAR,iBAAQ,C;QA65JhB,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAaKB,uBAAI,YAAJ,EAAL,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,uBAAL,cAAJ,EAAL,sBAAJ,UAAV,EAawB,WAAxB,C;;QAEIB,OAAO,W;O;KAtBX,C;2GAyBA,yB;MA75JI,8D;MA65JJ,uC;QAKB0B,UAEU,M;QAJhC,YA76JgB,cAAR,iBAAQ,C;QA86JhB,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAaKB,uBAAI,YAAJ,EAAL,oBAAJ,Q;QACIB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,uBAAL,cAAJ,EAAL,sBAAJ,UAAV,EAawB,WAAxB,C;;QAEIB,OAAO,W;O;KAtBX,C;+FAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAKBoB,Q;QAHhB,IAAI,mBAAJ,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,arBrRO,W;Q qBstRP,kBAaKB,O;QACF,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,cAAc,UAAU,WAAV,EAauB,OA AvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;+FAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAKBoB,Q;QAHhB,IAAI,mBAAJ,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,iBAAO,CAAP,

IAAb,C;QAA+B,8B;QAA5C,arB9uRO,W;QqB+uRP,kBAAkB,O;QACF,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;+FAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAKBoB,Q;QAHhB,IAAI,mBAAJ,C;UAAe,OAAO,OAAO,OAA P,C;QACc,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,arBvwRO,W;QqBwwRP,kBAAkB,O;QAC F,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;UACd,MAAO,W AAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;+FAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAKBoB,Q;QAHh B,IAAI,mBAAJ,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;QAA+B,8B;QA A5C,arBhyRO,W;QqBiyRP,kBAAkB,O;QACF,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,cAAc,UAAU, WAAV,EAAuB,OAAvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;6GAyBA,yB;MAAA,g D;MAAA,gE;MAlIKI,0D;MAkIKJ,gD;QAmBkB,gC;QAHd,IAAI,mBAAJ,C;UAAe,OAAO,OAAO,OAAP,C;QAC c,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,arB1zRO,W;QqB2zRP,kBAAkB,O;QACJ,OArmK E,YAAR,iBAAQ,C;QAqmKF,mB;QAAA,kB;QAAA,kB;QAAAd,0D;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB, EAA8B,sBAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAvBX,C;6GA0BA,yB; MAAA,gD;MAAA,gE;MApmKI,0D;MAomKJ,gD;QAmBkB,gC;QAHd,IAAI,mBAAJ,C;UAAe,OAAO,OAAO,O AAP,C;QACc,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,arBp1RO,W;QqBq1RP,kBAAkB,O;Q ACJ,OAvnKE,YAAR,iBAAQ,C;QAunKF,mB;QAAA,kB;QAAA,kB;QAAAd,0D;UACI,cAAc,UAAU,KAAV,EAAi B,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAvBX,C;6G A0BA,yB;MAAA,gD;MAAA,gE;MATnKI,0D;MASnKJ,gD;QAmBkB,gC;QAHd,IAAI,mBAAJ,C;UAAe,OAAO,O AAO,OAAP,C;QACc,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,arB92RO,W;QqB+2RP,kBAA kB,O;QACJ,OAZoKE,YAAR,iBAAQ,C;QAyoKF,mB;QAAA,kB;QAAA,kB;QAAAd,0D;UACI,cAAc,UAAU,KAA V,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAvB X,C;6GA0BA,yB;MAAA,gD;MAAA,gE;MAxoKI,0D;MAwoKJ,gD;QAmBkB,gC;QAHd,IAAI,mBAAJ,C;UAAe, OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,arBx4RO,W;QqBy4 RP,kBAAkB,O;QACJ,OA3pKE,YAAR,iBAAQ,C;QA2pKF,mB;QAAA,kB;QAAA,kB;QAAAd,0D;UACI,cAAc,UA AU,KAAV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M ;O;KAvBX,C;mGA0BA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAKB0B,Q;QAHtB,IAAI,mBAAJ,C;UAAe,OAA O,W;QACTb,sBAAkB,sBAAK,CAAL,CAAIB,C;QACmC,kBAAtB,eAAgB,cAAhB,C;QAA8B,sBAAI,aAAJ,C;Q AA3C,arB16RO,W;QqBm6Re,qB;QAAtB,iBAAc,CAAd,wB;UACI,gBAAc,UAAU,aAAV,EAAuB,sBAAK,KAAL ,CAAvB,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;mGAyBA,yB;MAAA,qD;MAAA,gE;M AAA,uC;QAKB0B,Q;QAHtB,IAAI,mBAAJ,C;UAAe,OAAO,W;QACTb,sBAAkB,sBAAK,CAAL,CAAIB,C;QAC oC,kBAAvB,eAAiB,cAAjB,C;QAA+B,sBAAI,aAAJ,C;QAA5C,arB37RO,W;QqB47Re,qB;QAAtB,iBAAc,CAAd, wB;UACI,gBAAc,UAAU,aAAV,EAAuB,sBAAK,KAAL,CAAvB,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAA O,M;O;KAtBX,C;mGAyBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAKB0B,Q;QAHtB,IAAI,mBAAJ,C;UAAe,O AAO,W;QACTb,sBAAkB,sBAAK,CAAL,CAAIB,C;QACoC,kBAAvB,eAAiB,cAAjB,C;QAA+B,sBAAI,aAAJ,C; QAA5C,arBp9RO,W;QqBq9Re,qB;QAAtB,iBAAc,CAAd,wB;UACI,gBAAc,UAAU,aAAV,EAAuB,sBAAK,KAA L,CAAvB,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;mGAyBA,yB;MAAA,qD;MAAA,gE; MAAA,uC;QAKB0B,Q;QAHtB,IAAI,mBAAJ,C;UAAe,OAAO,W;QACTb,sBAAkB,sBAAK,CAAL,CAAIB,C;QA CqC,kBAAxB,eAAkB,cAAIB,C;QAAgC,sBAAI,aAAJ,C;QAA7C,arB7+RO,W;QqB8+Re,qB;QAAtB,iBAAc,CA Ad,wB;UACI,gBAAc,UAAU,aAAV,EAAuB,sBAAK,KAAL,CAAvB,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,O AAO,M;O;KAtBX,C;iHAyBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAmB0B,Q;QAHtB,IAAI,mBAAJ,C;UAAe ,OAAO,W;QACTb,sBAAkB,sBAAK,CAAL,CAAIB,C;QACmC,kBAAtB,eAAgB,cAAhB,C;QAA8B,sBAAI,aAAJ ,C;QAA3C,arBvgSO,W;QqBwgSe,qB;QAAtB,iBAAc,CAAd,wB;UACI,gBAAc,UAAU,KAAV,EAAiB,aAAjB,EA A8B,sBAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAvBX,C;iHA0BA,yB;MAA A,qD;MAAA,gE;MAAA,uC;QAmB0B,Q;QAHtB,IAAI,mBAAJ,C;UAAe,OAAO,W;QACTb,sBAAkB,sBAAK,CA AL,CAAIB,C;QACoC,kBAAvB,eAAiB,cAAjB,C;QAA+B,sBAAI,aAAJ,C;QAA5C,arBjiSO,W;QqBkiSe,qB;QAA tB,iBAAc,CAAd,wB;UACI,gBAAc,UAAU,KAAV,EAAiB,aAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;UACd,MA AO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAvBX,C;iHA0BA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAmB0B,Q; QAHtB,IAAI,mBAAJ,C;UAAe,OAAO,W;QACTb,sBAAkB,sBAAK,CAAL,CAAIB,C;QACoC,kBAAvB,eAAiB,c

AAjB,C;QAA+B,sBAAl,aAAJ,C;QAA5C,arB3jSO,W;QqB4jSe,qB;QAAtB,iBAAC,CAAd,wB;UACI,gBAAC,UA
AU,KAAV,EAAiB,aAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O
;KAvBX,C;iHA0BA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAmB0B,Q;QAHtB,IAAI,mBAAJ,C;UAAe,OAAO,
W;QACtB,sBAAkB,sBAAK,CAAL,CAAIb,C;QACqC,kBAAXB,eAAkB,cAAIB,C;QAAgC,sBAAl,aAAJ,C;QAA
7C,arBrlSO,W;QqBslSe,qB;QAAtB,iBAAC,CAAd,wB;UACI,gBAAC,UAAU,KAAV,EAAiB,aAAjB,EAA8B,sBA
AK,KAAL,CAA9B,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAvBX,C;iFA0BA,yB;MAxZA,gD;M
AAA,gE;MAwZA,gD;QAgBW,sB;;UAtZS,Q;UAHhB,IAAI,mBAAJ,C;YAAe,qBAAO,OAYZH,OAZG,C;YAAP,
uB;;UACqB,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;UAA+B,sBAwZzB,OAXzYB,C;UAA5C,arBrRO,W;UqBstR
P,kBAuZmB,O;UAtZH,2B;UAAhB,OAAGB,cAAhB,C;YAAgB,yB;YACZ,cAqZwB,SArZV,CAAU,WAAV,EAAu
B,OAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QAKZP,yB;O;KAhBJ,C;iFamBA,yB;MAIza,g
D;MAAA,gE;MAKZA,gD;QAgBW,sB;;UAhZS,Q;UAHhB,IAAI,mBAAJ,C;YAAe,qBAAO,OAmZH,OAnZG,C;Y
AAP,uB;;UACqB,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;UAA+B,sBAkZzB,OAlZYB,C;UAA5C,arB9uRO,W;Uq
B+uRP,kBAiZmB,O;UAhZH,2B;UAAhB,OAAGB,cAAhB,C;YAAgB,yB;YACZ,cA+YwB,SA/YV,CAAU,WAAV
,EAAuB,OAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QA4YP,yB;O;KAhBJ,C;iFamBA,yB;M
A5YA,gD;MAAA,gE;MA4YA,gD;QAgBW,sB;;UA1YS,Q;UAHhB,IAAI,mBAAJ,C;YAAe,qBAAO,OA6YH,OA7
YG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;UAA+B,sBA4YzB,OA5YyB,C;UAA5C,arBvw
RO,W;UqBwwRP,kBA2YmB,O;UA1YH,2B;UAAhB,OAAGB,cAAhB,C;YAAgB,yB;YACZ,cAyYwB,SAzYV,C
AAU,WAAV,EAAuB,OAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QAsYP,yB;O;KAhBJ,C;iF
AmBA,yB;MAtYA,gD;MAAA,gE;MA5YA,gD;QAgBW,sB;;UApYS,Q;UAHhB,IAAI,mBAAJ,C;YAAe,qBAAO,
OAuYH,OAvYG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;UAA+B,sBA5YzB,OAtYyB,C;U
AA5C,arBhyRO,W;UqBiyRP,kBAqYmB,O;UApYH,2B;UAAhB,OAAGB,cAAhB,C;YAAgB,yB;YACZ,cAmYwB
,SAnYV,CAAU,WAAV,EAAuB,OAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QAgYP,yB;O;K
AhBJ,C;+FamBA,yB;MAhYA,gD;MAAA,gE;MAIIKI,0D;MAK9KJ,gD;QAIbW,6B;;UA9XO,gC;UAHd,IAAI,mB
AAJ,C;YAAe,4BAAO,OAIYI,OAJYJ,C;YAAP,8B;;UACqB,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;UAA+B,sBA
gYIB,OAhYkB,C;UAA5C,arB1zRO,W;UqB2zRP,kBA+X0B,O;UA9XZ,OArmKE,YAAR,iBAAQ,C;UAqmKF,m
B;UAAA,kB;UAAA,kB;UAAAd,0D;YACI,cA6X+B,SA7XjB,CAAU,KAAV,EAAiB,WAAjB,EAA8B,sBAAK,KA
AL,CAA9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;;QA0XP,gC;O;KAjBJ,C;+FAoBA,yB;MA1XA
,gD;MAAA,gE;MApMKI,0D;MA89KJ,gD;QAIbW,6B;;UAxXO,gC;UAHd,IAAI,mBAAJ,C;YAAe,4BAAO,OA2
XI,OA3XJ,C;YAAP,8B;;UACqB,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;UAA+B,sBA0XIB,OA1XkB,C;UAA5C,
arBp1RO,W;UqBq1RP,kBAyX0B,O;UAxXZ,OAvnKE,YAAR,iBAAQ,C;UAunKF,mB;UAAA,kB;UAAA,kB;UA
Ad,0D;YACI,cAuX+B,SAvXjB,CAAU,KAAV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;YACd,MAAO
,WAAI,WAAJ,C;;UAEX,4BAAO,M;;;QAoXP,gC;O;KAjBJ,C;+FAoBA,yB;MApXA,gD;MAAA,gE;MAtnKI,0D;
MA0+KJ,gD;QAIbW,6B;;UAIXO,gC;UAHd,IAAI,mBAAJ,C;YAAe,4BAAO,OAqXI,OArXJ,C;YAAP,8B;;UACq
B,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;UAA+B,sBAoXIB,OApXkB,C;UAA5C,arB92RO,W;UqB+2RP,kBAm
X0B,O;UAIXZ,OAzoKE,YAAR,iBAAQ,C;UAyoKF,mB;UAAA,kB;UAAA,kB;UAAAd,0D;YACI,cAiX+B,SAjXjB
,CAAU,KAAV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAA
O,M;;;QA8WP,gC;O;KAjBJ,C;+FAoBA,yB;MA9WA,gD;MAAA,gE;MAxoKI,0D;MA5/KJ,gD;QAIbW,6B;;UA5
WO,gC;UAHd,IAAI,mBAAJ,C;YAAe,4BAAO,OA+Wl,OA/WJ,C;YAAP,8B;;UACqB,kBAAvB,eAAa,iBAAO,C
AAP,IAAb,C;UAA+B,sBA8WIB,OA9WkB,C;UAA5C,arBx4RO,W;UqBy4RP,kBA6W0B,O;UA5WZ,OA3pKE,Y
AAR,iBAAQ,C;UA2pKF,mB;UAAA,kB;UAAA,kB;UAAAd,0D;YACI,cA2W+B,SA3WjB,CAAU,KAAV,EAAiB,
WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;;QAwWP,gC;O;
KAjBJ,C;mFAoBA,yB;MAAA,wB;MAAA,sC;QAuOB,Q;QADhB,UAAgB,W;QACA,2B;QAAhB,OAAGB,cAAh
B,C;UAAgB,yB;UACZ,MnCvWsiD,SmCuWsjD,GnCvW2D,KAAK,GmCuWszD,SAAS,OAAT,CnCvWsoE,KAA
X,IAAf,C;;QmCywSrD,OAAO,G;O;KAbX,C;mFAGBA,yB;MAAA,wB;MAAA,sC;QAuOB,Q;QADhB,UAAgB,W
;QACA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,MnCvxSiD,SmCuxSjD,GnCvxS2D,KAAK,GmCuxSz
D,SAAS,OAAT,CnCvxSoE,KAAx,IAAf,C;;QmCyxSrD,OAAO,G;O;KAbX,C;mFAGBA,yB;MAAA,wB;MAAA,s
C;QAuOB,Q;QADhB,UAAgB,W;QACA,2B;QAAhB,OAAGB,cAAhB,C;UAAgB,yB;UACZ,MnCvySiD,SmCuySj
D,GnCvyS2D,KAAK,GmCuySzD,SAAS,OAAT,CnCvySoE,KAAx,IAAf,C;;QmCyySrD,OAAO,G;O;KAbX,C;m

FaGBA,yB;MAAA,wB;MAAA,sC;QAUoB,Q;QADhB,UAAgB,W;QACA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,MnCvzSiD,SmCuzSjD,GnCvzS2D,KAAK,GmCuzSzD,SAAS,OAAT,CnCvzSoE,KAAX,IAAf,C;;QmCyzSrD,OAAO,G;O;KAhBX,C;8FAgBA,+B;MAUoB,Q;MADhB,UAAkB,G;MACF,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;+FAGX,+B;MAUoB,Q;MADhB,UAAkB,G;MA CF,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;+FAGX,+B; MAUoB,Q;MADhB,UAAkB,G;MACF,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,SAAS,OAAT, C;;MAEX,OAAO,G;K;+FAGX,+B;MAUoB,Q;MADhB,UAAkB,G;MACF,2B;MAAhB,OAAgB,cAAhB,C;QAAg B,yB;QACZ,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;kFAGX,+B;MAYoB,Q;MADhB,UAAoB,C;MACJ,2B; MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;mFAGX,+B;MAYo B,Q;MADhB,UAAoB,C;MACJ,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,SAAS,OAAT,C;;MAE X,OAAO,G;K;mFAGX,+B;MAYoB,Q;MADhB,UAAoB,C;MACJ,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;Q ACZ,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;mFAGX,+B;MAYoB,Q;MADhB,UAAoB,C;MACJ,2B;MAAh B,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;mFAGX,+B;MAYoB,Q;M ADhB,UAAe,C;MACC,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,YAAO,SAAS,OAAT,CAAP,I;;MAEJ, OAAO,G;K;mFAGX,+B;MAYoB,Q;MADhB,UAAe,C;MACC,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ ,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;mFAGX,+B;MAYoB,Q;MADhB,UAAe,C;MACC,2B;MAAhB ,OAAgB,cAAhB,C;QAAgB,yB;QACZ,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;mFAGX,+B;MAYoB,Q ;MADhB,UAAe,C;MACC,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,YAAO,SAAS,OAAT,CAAP,I;;MA EJ,OAAO,G;K;mFAGX,yB;MAAA,SAWoB,gB;MAXpB,sC;QAYoB,Q;QADhB,Y;QACgB,2B;QAAhB,OAAgB, cAAhB,C;UAAgB,yB;UACZ,cAAO,SAAS,OAAT,CAAP,C;;QAEJ,OAAO,G;O;KAFx,C;mFAkBA,yB;MAAA,S AWoB,gB;MAXpB,sC;QAYoB,Q;QADhB,Y;QACgB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,cAAO,S AAS,OAAT,CAAP,C;;QAEJ,OAAO,G;O;KAFx,C;mFAkBA,yB;MAAA,SAWoB,gB;MAXpB,sC;QAYoB,Q;QA DhB,Y;QACgB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,cAAO,SAAS,OAAT,CAAP,C;;QAEJ,OAAO, G;O;KAFx,C;mFAkBA,yB;MAAA,SAWoB,gB;MAXpB,sC;QAYoB,Q;QADhB,Y;QACgB,2B;QAAhB,OAAgB,c AAhB,C;UAAgB,yB;UACZ,cAAO,SAAS,OAAT,CAAP,C;;QAEJ,OAAO,G;O;KAFx,C;mFAkBA,yB;MnC5xSA, 6B;MmC4xSA,sC;QAaoB,Q;QADhB,UnC9xSmC,cmC8xSnB,CnC9xSmB,C;QmC+xSnB,2B;QAAhB,OAAgB,cA AhB,C;UAAgB,yB;UACZ,MnCImTiD,cmCkmTjD,GnCImT2D,KAAK,GmCkmTzD,SAAS,OAAT,CnCImToE,K AAX,IAAf,C;;QmComTrD,OAAO,G;O;KAhBX,C;mFAmBA,yB;MnC/ySA,6B;MmC+ySA,sC;QAaoB,Q;QADhB ,UnCjzSmC,cmCizSnB,CnCjzSmB,C;QmCkzSnB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,MnCrmTiD,c mCqnTjD,GnCrmT2D,KAAK,GmCqnTzD,SAAS,OAAT,CnCrmToE,KAAX,IAAf,C;;QmCunTrD,OAAO,G;O;KA hBX,C;mFAmBA,yB;MnCl0SA,6B;MmCk0SA,sC;QAaoB,Q;QADhB,UnCp0SmC,cmCo0SnB,CnCp0SmB,C;Qm Cq0SnB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,MnCxoTiD,cmCwoTjD,GnCxoT2D,KAAK,GmCwoT zD,SAAS,OAAT,CnCxoToE,KAAX,IAAf,C;;QmC0oTrD,OAAO,G;O;KAhBX,C;mFAmBA,yB;MnCr1SA,6B;M mCq1SA,sC;QAaoB,Q;QADhB,UnCv1SmC,cmCu1SnB,CnCv1SmB,C;QmCw1SnB,2B;QAAhB,OAAgB,cAAhB, C;UAAgB,yB;UACZ,MnC3pTiD,cmC2pTjD,GnC3pT2D,KAAK,GmC2pTzD,SAAS,OAAT,CnC3pToE,KAAX,IA Af,C;;QmC6pTrD,OAAO,G;O;KAhBX,C;mFAmBA,yB;MnBr2SA,+B;MmBq2SA,sC;QAaoB,Q;QADhB,UnBt2S qC,eAAW,oBmBs2S/B,CnBt2S+B,CAAX,C;QmBu2SrB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,MnB 3qTmD,emB2qTnD,GnB3qT8D,KAAK,KmB2qT5D,SAAS,OAAT,CnB3qTuE,KAAX,CAAhB,C;;QmB6qTvD,O AAO,G;O;KAhBX,C;mFAmBA,yB;MnBx3SA,+B;MmBw3SA,sC;QAaoB,Q;QADhB,UnBz3SqC,eAAW,oBmBy3 S/B,CnBz3S+B,CAAX,C;QmB03SrB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,MnB9rTmD,emB8rTnD, GnB9rT8D,KAAK,KmB8rT5D,SAAS,OAAT,CnB9rTuE,KAAX,CAAhB,C;;QmBgsTvD,OAAO,G;O;KAhBX,C; mFAmBA,yB;MnB34SA,+B;MmB24SA,sC;QAaoB,Q;QADhB,UnB54SqC,eAAW,oBmB44S/B,CnB54S+B,CAA X,C;QmB64SrB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,MnBjtTmD,emBitTnD,GnBjtT8D,KAAK,Km BitT5D,SAAS,OAAT,CnBjtTuE,KAAX,CAAhB,C;;QmBmtTvD,OAAO,G;O;KAhBX,C;mFAmBA,yB;MnB95SA ,+B;MmB85SA,sC;QAaoB,Q;QADhB,UnB/5SqC,eAAW,oBmB+5S/B,CnB/5S+B,CAAX,C;QmBg6SrB,2B;QAAh B,OAAgB,cAAhB,C;UAAgB,yB;UACZ,MnBpuTmD,emBouTnD,GnBpuT8D,KAAK,KmBouT5D,SAAS,OAAT, CnBpuTuE,KAAX,CAAhB,C;;QmBsuTvD,OAAO,G;O;KAhBX,C;IAMBA,kC;MA2DI,WpBnnTO,MAAO,KoBm nTG,cpBnnTH,EoBikTH,KAkDkB,OpBnnTf,C;MoBonTd,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB

,IAAIB,M;QACI,IAAK,WArDqB,GAqDP,sBAAK,CAAL,CArDO,EAAnB,KAqDqB,CAAM,CAAN,CArDF,CAqDrB,C;;MArDT,OAUdO,I;K;IApDX,kC;MAkEI,WpBtoTO,MAAO,KoBsoTG,cpBtoTH,EoB6kTH,KAyDkB,OpBtoTf,C;MoBuoTd,WAAW,iBAaA,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WA5DqB,GA4DP,sBAAK,CAAL,CA5DO,EAAnB,KA4DqB,CAAM,CAAN,CA5DF,CA4DrB,C;;MA5DT,OA8DO,I;K;IA3DX,kC;MAyEI,WpBzpTO,MAAO,KoBypTG,cpBzpTH,EoBylTH,KAgEkB,OpBzpTf,C;MoB0pTd,WAAW,iBAaA,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WAnEqB,GAmEP,sBAAK,CAAL,CAnEO,EAAnB,KAmEqB,CAAM,CAAN,CAnEF,CAmErB,C;;MAnET,OAqEO,I;K;IAIEX,kC;MAGFI,WpB5qTO,MAAO,KoB4qTG,cpB5qTH,EoBqmTH,KAuEkB,OpB5qTf,C;MoB6qTd,WAAW,iBAaA,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WA1EqB,GA0EP,sBAAK,CAAL,CA1EO,EAAnB,KA0EqB,CAAM,CAAN,CA1EF,CA0ErB,C;;MAIET,OA4EO,I;K;+EAzEX,yB;MAAA,gE;MpB9mTA,iB;MoB8mTA,8C;QAWI,WpBnnTO,MAAO,KoBm nTG,cpBnnTH,EoBmnTS,KAAM,OpBnnTf,C;QoBonTd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,sBAAK,CAAL,CAAV,EAAmB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAhBX,C;+EAmBA,yB;MAAA,gE;MpBjoTA,iB;MoBioTA,8C;QAWI,WpBtoTO,MAAO,KoBsoTG,cpBtoTH,EoBsoTS,KAAM,OpBtoTf,C;QoBuoTd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,sBAAK,CAAL,CAAV,EAAmB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAhBX,C;+EAmBA,yB;MAAA,gE;MpBppTA,iB;MoBopTA,8C;QAWI,WpBzpTO,MAAO,KoBypTG,cpBzpTH,EoBypTS,KAAM,OpBzpTf,C;QoB0pTd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,sBAAK,CAAL,CAAV,EAAmB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAhBX,C;+EAmBA,yB;MAAA,gE;MpBvqTA,iB;MoBuqTA,8C;QAWI,WpB5qTO,MAAO,KoB4qTG,cpB5qTH,EoB4qTS,KAAM,OpB5qTf,C;QoB6qTd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,sBAAK,CAAL,CAAV,EAAmB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAhBX,C;IAmBA,kC;MA8DoB,gB;MAHhB,gBAAgB,c;MACHB,WAAW,iBpBhvTJ,MAAO,KoBgvTsB,wBAnDzB,KAmDyB,EAawB,EAaxB,CpBhvTtB,EoBgvTmD,SpBhvTnD,CoBgvTH,C;MACX,QAAQ,C;MACQ,OArDL,KAqDK,W;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,IAAI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK,WAvDqB,GAuDP,uBAAK,UAAAL,EAak,kBAAL,UAvDO,EAuDI,OAxDJ,CAuDrB,C;;MAvDT,OAYDO,I;K;IAtdX,kC;MAuEoB,gB;MAHhB,gBAAGB,c;MACHB,WAAW,iBpBrwTJ,MAAO,KoBqwTsB,wBA5DzB,KA4DyB,EAawB,EAaxB,CpBrwTtB,EoBqwTmD,SpBrwTnD,CoBqwTH,C;MACX,QAAQ,C;MACQ,OA9DL,KA8DK,W;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,IAAI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK,WAhEqB,GAGEP,uBAAK,UAAAL,EAak,kBAAL,UAhEO,EAgeI,OAheJ,CAGErB,C;;MAhET,OAkeO,I;K;IA/DX,kC;MAGFoB,gB;MAHhB,gBAAGB,c;MACHB,WAAW,iBpB1xTJ,MAAO,KoB0xTsB,wBArEzB,KAqEyB,EAawB,EAaxB,CpB1xTtB,EoB0xTmD,SpB1xTnD,CoB0xTH,C;MACX,QAAQ,C;MACQ,OAveL,KAuEK,W;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,IAAI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK,WAZEqB,GAYEP,uBAAK,UAAAL,EAak,kBAAL,UAZEO,EAyEI,OAzeJ,CAYErB,C;;MAzET,OA2EO,I;K;IAxEX,kC;MAyFoB,gB;MAHhB,gBAAGB,c;MACHB,WAAW,iBpB/yTJ,MAAO,KoB+yTsB,wBA9EzB,KA8EyB,EAawB,EAaxB,CpB/yTtB,EoB+yTmD,SpB/yTnD,CoB+yTH,C;MACX,QAAQ,C;MACQ,OAhFL,KAGFK,W;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,IAAI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK,WAlFqB,GAKFP,uBAAK,UAAAL,EAak,kBAAL,UAlFO,EAKFI,OAIFJ,CakFrB,C;;MAIFT,OAoFO,I;K;+EAjFX,yB;MAAA,kF;MAAA,gE;MpB1uTA,iB;MoB0uTA,8C;QAcOB,UAEY,M;QAL5B,gBAAGB,c;QACHB,WAAW,epBhvTJ,MAAO,KoBgvTsB,wBAAN,KAAM,EAawB,EAaxB,CpBhvTtB,EoBgvTmD,SpBhvTnD,CoBgvTH,C;QACX,QAAQ,C;QACQ,uB;QAAhB,OAAGB,cAAhB,C;UAGB,yB;UACZ,IAAI,KAAK,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,uBAAK,UAAAL,EAak,kBAAL,UAAV,EAaqB,OAARb,CAAJ,C;;QAET,OAAO,I;O;KAIBX,C;+EAqBA,yB;MAAA,kF;MAAA,gE;MpB/vTA,iB;MoB+vTA,8C;QAcOB,UAEY,M;QAL5B,gBAAGB,c;QACHB,WAAW,epBrwTJ,MAAO,KoBqwTsB,wBAAN,KAAM,EAawB,EAaxB,CpBrwTtB,EoBqwTmD,SpBrwTnD,CoBqwTH,C;QACX,QAAQ,C;QACQ,uB;QAAhB,OAAGB,cAAhB,C;UAGB,yB;UACZ,IAAI,KAAK,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,uBAAK,UAAAL,EAak,kBAAL,UAAV,EAaqB,OAARb,CAAJ,C;;QAET,OAAO,I;O;KAIBX,C;+EAqBA,yB;MAAA,kF;MAAA,gE;MpBpxTA,iB;MoBoxTA,8C;QAcOB,UAEY,M;QAL5B,gBAAGB,c;QACHB,WAAW,epB1xTJ,MAAO,KoB0xTsB,wBAAN,KAAM,EAawB,EAaxB,CpB1xTtB,EoB0xTmD,SpB1xTnD,CoB0xTH,C;QACX,QAAQ,C;QACQ,uB;QAAhB,OAAGB,cAAhB,C;UAGB,yB;UACZ,IAAI,KAAK,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,uBAAK,UAAAL,EAak,k

BAAL,UAAV,EAAqB,OAARb,CAAJ,C;;QAET,OAAO,I;O;KAIBX,C;8EAqBA,yB;MAAA,kF;MAAA,gE;MpBzy
TA,iB;MoByyTA,8C;QAcOB,UAEY,M;QAL5B,gBAAgB,c;QACHb,WAAW,epB/yTJ,MAAO,KoB+yTsB,wBAA
N,KAAM,EAawB,EAaxB,CpB/yTtB,EoB+yTmD,SpB/yTnD,CoB+yTH,C;QACX,QAAQ,C;QACQ,uB;QAAhB,
OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAAI,KAAK,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,uBAAK,U
AAL,EAAK,kBAAL,UAAV,EAAqB,OAARb,CAAJ,C;;QAET,OAAO,I;O;KAIBX,C;IAqBA,kC;MA2DI,WpBn3T
O,MAAO,KoBm3TG,cpBn3TH,EoBi0TH,KakDkB,KpBn3Tf,C;MoBo3Td,WAAW,iBAAa,IAAb,C;MACX,aAA
U,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WArDqB,GAqDP,sBAAK,CAAL,CARDO,EAAAnB,KAqDqB,aAAM,C
AAN,CARDF,CAqDrB,C;;MARDT,OAuDO,I;K;IAPDX,kC;MAkEI,WpBt4TO,MAAO,KoBs4TG,cpBt4TH,EoB60
TH,KAYdkB,KpBt4Tf,C;MoBu4Td,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAA
K,WA5DqB,GA4DP,sBAAK,CAAL,CA5DO,EAAAnB,KA4DqB,aAAM,CAAN,CA5DF,CA4DrB,C;;MA5DT,OA8
DO,I;K;IA3DX,kC;MAyEI,WpBz5TO,MAAO,KoBy5TG,cpBz5TH,EoBy1TH,KAgEkB,KpBz5Tf,C;MoB05Td,W
AAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAIB,M;QACI,IAAK,WAnEqB,GAmEP,sBAAK,CAAL,C
AnEO,EAAAnB,KAmEqB,aAAM,CAAN,CAnEF,CAMerB,C;;MANET,OAqEO,I;K;IAIEX,kC;MAgFI,WpB56TO,
MAAO,KoB46TG,cpB56TH,EoBq2TH,KAuEkB,KpB56Tf,C;MoB66Td,WAAW,iBAAa,IAAb,C;MACX,aAAU,C
AAV,MAAkB,IAAIB,M;QACI,IAAK,WA1EqB,GA0EP,sBAAK,CAAL,CA1EO,EAAAnB,KA0EqB,aAAM,CAAN,
CA1EF,CA0ErB,C;;MA1ET,OA4EO,I;K;+EAzEX,yB;MAAA,gE;MpB92TA,iB;MoB82TA,8C;QAWI,WpBn3TO,
MAAO,KoBm3TG,cpBn3TH,EoBm3TS,KAAM,KpBn3Tf,C;QoBo3Td,WAAW,eAAa,IAAb,C;QACX,aAAU,CA
AV,MAAkB,IAAIB,M;UACI,IAAK,WAAI,UAAU,sBAAK,CAAL,CAAV,EAAMb,kBAAM,CAAN,CAAnB,CA
AJ,C;;QAET,OAAO,I;O;KAhBX,C;+EAmBA,yB;MAAA,gE;MpBj4TA,iB;MoBi4TA,8C;QAWI,WpBt4TO,MAA
O,KoBs4TG,cpBt4TH,EoBs4TS,KAAM,KpBt4Tf,C;QoBu4Td,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAA
kB,IAAIB,M;UACI,IAAK,WAAI,UAAU,sBAAK,CAAL,CAAV,EAAMb,kBAAM,CAAN,CAAnB,CAAJ,C;;QAE
T,OAAO,I;O;KAhBX,C;+EAmBA,yB;MAAA,gE;MpBp5TA,iB;MoBo5TA,8C;QAWI,WpBz5TO,MAAO,KoBy5
TG,cpBz5TH,EoBy5TS,KAAM,KpBz5Tf,C;QoB05Td,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAA
IB,M;UACI,IAAK,WAAI,UAAU,sBAAK,CAAL,CAAV,EAAMb,kBAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAA
O,I;O;KAhBX,C;+EAmBA,yB;MAAA,gE;MpBv6TA,iB;MoBu6TA,8C;QAWI,WpB56TO,MAAO,KoB46TG,cpB
56TH,EoB46TS,KAAM,KpB56Tf,C;QoB66Td,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAIB,M;U
ACI,IAAK,WAAI,UAAU,sBAAK,CAAL,CAAV,EAAMb,kBAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;
KAhBX,C;IAmBA,2B;MAQoB,Q;MADhB,UAAgB,W;MACHb,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;
QACI,MnCjnUiD,SmCinUjD,GnCjnU2D,KAAK,GmCinUzD,OnCjnUoE,KAAx,IAAf,C;;MmCmnUrD,OAAO,G;
K;IAGX,2B;MAQoB,Q;MADhB,UAAiB,2B;MACjB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,Mn
B5nUmD,UmB4nUnD,GnB5nU8D,KAAK,KmB4nU5D,OnB5nUuE,KAAx,CAAhB,C;;MmB8nUvD,OAAO,G;K;
IAGX,2B;MAQoB,Q;MADhB,UAAgB,W;MACHb,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,MnC7
oUiD,SmC6oUjD,GnC7oU2D,KAAK,GAAW,CD2O5C,SoCk6TxB,OpCl6TkC,KAAL,GAAiB,GAAtB,CC3O4C,
MAAX,IAAf,C;;MmC+oUrD,OAAO,G;K;IAGX,2B;MAQoB,Q;MADhB,UAAgB,W;MACHb,wBAAgB,SAAhB,
gB;QAAgB,cAAA,SAAhB,M;QACI,MnC3pUiD,SmC2pUjD,GnC3pU2D,KAAK,GAAW,CC4O5C,SkC+6TxB,Ol
C/6TkC,KAAL,GAAiB,KAAtB,CD5O4C,MAAX,IAAf,C;;MmC6pUrD,OAAO,G;K;+EAGX,yB;MAAA,0C;MnC
x2TA,6B;MmCw2TA,4B;QAOI,OnCr2TmC,cmCq2TpB,IAAR,iBAAQ,CnCr2ToB,C;O;KmC81TvC,C;+EAUA,y
B;MAAA,0C;MnBn2TA,+B;MmBm2TA,4B;QAOI,OnBh2TsC,emBg2TvB,IAAR,iBAAQ,CnBh2TuB,C;O;KmBy
1T1C,C;+EAUA,yB;MAAA,sC;MnC53TA,6B;MmC43TA,iBAOiB,yB;QpCz9Tb,6B;eoCy9Ta,c;UAAE,OpCh9To
B,coCg9TpB,EpCh9T8B,KAAL,GAAiB,GAAtB,C;S;OoCg9TtB,C;MAPjB,4B;QA7iBoB,Q;QADhB,UnCp0SmC,
cmCo0SnB,CnCP0SmB,C;QmCq0SnB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,MnCxoTiD,cmCwoTjD,
GnCxoT2D,KAAK,GAAW,CD2O5C,coC65Sf,OpC75SyB,KAAL,GAAiB,GAAtB,CC3O4C,MAAX,IAAf,C;;Qm
C2rUrD,OAjjoB,G;O;KA0iBX,C;+EAUA,yB;MAAA,sC;MnCt4TA,6B;MmCs4TA,iBAOiB,yB;QlCl+Tb,6B;ekCk
+Ta,c;UAAE,OICz9ToB,ckCy9TpB,EICz9T8B,KAAL,GAAiB,KAAtB,C;S;OkCy9TtB,C;MAPjB,4B;QApiBoB,Q;
QADhB,UnCv1SmC,cmCu1SnB,CnCV1SmB,C;QmCw1SnB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,M
nC3pTiD,cmC2pTjD,GnC3pT2D,KAAK,GAAW,CC4O5C,ckC+6Sf,OIC/6SyB,KAAL,GAAiB,KAAtB,CD5O4C,
MAAX,IAAf,C;;QmCqsUrD,OAXiBO,G;O;KAiiBX,C;IC3vUA,mC;MAQoB,UACL,M;MAHX,aAAa,gBAAW,cA
AX,C;MACb,YAAY,C;MACI,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,oBAAO,cAAP,EAAO,sBAAP,

WAAkB,OAAIB,C;;MACJ,OAAO,M;K;IAGX,kC;MAQoB,UACL,M;MAHX,aAAa,eAAU,cAAV,C;MACb,YAA Y,C;MACI,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,oBAAO,cAAP,EAAO,sBAAP,WAAkB,OAAIB,C; ;MACJ,OAAO,M;K;IAGX,mC;MAQoB,UACL,M;MAHX,aAAa,gBAAW,cAAx,C;MACb,YAAy,C;MACI,2B;M AAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,oBAAO,cAAP,EAAO,sBAAP,WAAkB,OAAIB,C;;MACJ,OAAO,M ;K;IAGX,oC;MAQoB,UACL,M;MAHX,aAAa,iBAAy,cAAZ,C;MACb,YAAy,C;MACI,2B;MAAhB,OAAGB,cA AhB,C;QAAGB,yB;QACZ,oBAAO,cAAP,EAAO,sBAAP,WAAkB,OAAIB,C;;MACJ,OAAO,M;K;IAGX,2B;MA QoB,Q;MADhB,UAGB,W;MACA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,MpCAiD,SoCAjD,GpCA 2D,KAAK,GoCAzD,OpCAoE,KAAx,IAAf,C;;MoCErD,OAAO,G;K;IAGX,2B;MAQoB,Q;MADhB,UAAiB,2B; MACD,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,MpBXmD,UoBWnD,GpBX8D,KAAK,KoBW5D,OpB XuE,KAAx,CAAhB,C;;MoBavD,OAAO,G;K;IAGX,2B;MAQoB,Q;MADhB,UAGB,W;MACA,2B;MAAhB,OA AgB,cAAhB,C;QAAGB,yB;QACZ,MpC5BiD,SoC4BjD,GpC5B2D,KAAK,GAAW,CD2O5C,SqC/MxB,OrC+MkC ,KAAL,GAAiB,GAAtB,CC3O4C,MAAX,IAAf,C;;MoC8BrD,OAAO,G;K;IAGX,2B;MAQoB,Q;MADhB,UAGB ,W;MACA,2B;MAAhB,OAAGB,cAAhB,C;QAAGB,yB;QACZ,MpC1CiD,SoC0CjD,GpC1C2D,KAAK,GAAW,CC 4O5C,SmClMxB,OnCkMkC,KAAL,GAAiB,KAAtB,CD5O4C,MAAX,IAAf,C;;MoC4CrD,OAAO,G;K;IC3GX,w B;MAMI,OrCuCkE,YqCvCvD,CrCuCwE,KAAjB,EqCvCiD,CrCuC+E,KAA7B,CqCvCvD,KAAJ,GAAY,CAAZ,G AAmB,C;K;IAG9B,wB;MAMI,OrBsCmE,aqBtCxD,CrBsC0E,KAAiB,EqBtCnD,CrBsCiF,KAA9B,CqBtCxD,KA AJ,GAAY,CAAZ,GAAM,B,C;K;IAG9B,wB;MAMI,OtCKgF,0BsCLrE,CtCgP2B,KAAL,GAAiB,GA3O8B,EsCLh E,CtCgPsB,KAAL,GAAiB,GA3O8B,CsCLrE,KAAJ,GAAY,CAAZ,GAAM,B,C;K;IAG9B,wB;MAMI,OpClIF,0Bo CjTE,CpCwO2B,KAAL,GAAiB,KApO+B,EoCjJE,CpCwOsB,KAAL,GAAiB,KApO+B,CoCjTE,KAAJ,GAAY,CA AZ,GAAM,B,C;K;mFAG9B,yB;MAAA,8C;MAAA,0B;QAOI,OAAO,MAAM,CAAN,EAAS,MAAM,CAAN,EAA S,CAAT,CAAT,C;O;KAPX,C;mFAUA,yB;MAAA,8C;MAAA,0B;QAOI,OAAO,MAAM,CAAN,EAAS,MAAM,C AAN,EAAS,CAAT,CAAT,C;O;KAPX,C;mFAUA,yB;MAAA,8C;MAAA,0B;QAOI,OAAO,MAAM,CAAN,EAA S,MAAM,CAAN,EAAS,CAAT,CAAT,C;O;KAPX,C;mFAUA,yB;MAAA,8C;MAAA,0B;QAOI,OAAO,MAAM,C AAN,EAAS,MAAM,CAAN,EAAS,CAAT,CAAT,C;O;KAPX,C;IAUA,4B;MAOc,Q;MADV,UAAU,C;MACA,uB ;MAAV,OAAU,cAAV,C;QAAU,mB;QAAO,MAAM,SAAM,GAAN,EAAW,CAAX,C;;MACvB,OAAO,G;K;IAG X,4B;MAOc,Q;MADV,UAAU,C;MACA,uB;MAAV,OAAU,cAAV,C;QAAU,mB;QAAO,MAAM,SAAM,GAAN, EAAW,CAAX,C;;MACvB,OAAO,G;K;IAGX,4B;MAOc,Q;MADV,UAAU,C;MACA,uB;MAAV,OAAU,cAAV,C ;QAAU,mB;QAAO,MAAM,SAAM,GAAN,EAAW,CAAX,C;;MACvB,OAAO,G;K;IAGX,4B;MAOc,Q;MADV,U AAU,C;MACA,uB;MAAV,OAAU,cAAV,C;QAAU,mB;QAAO,MAAM,SAAM,GAAN,EAAW,CAAX,C;;MACv B,OAAO,G;K;IAGX,wB;MAMI,OrCjFkE,YqCiFvD,CrCjFwE,KAAjB,EqCiFiD,CrCjF+E,KAA7B,CqCiFvD,KAA J,GAAY,CAAZ,GAAM,B,C;K;IAG9B,wB;MAMI,OrBIFmE,aqBkFxD,CrBIF0E,KAAiB,EqBkFnD,CrBIFiF,KAA9 B,CqBkFxD,KAAJ,GAAY,CAAZ,GAAM,B,C;K;IAG9B,wB;MAMI,OtCnHgF,0BsCmHrE,CtCwH2B,KAAL,GAA iB,GA3O8B,EsCmHhE,CtCwHsB,KAAL,GAAiB,GA3O8B,CsCmHrE,KAAJ,GAAY,CAAZ,GAAM,B,C;K;IAG9B ,wB;MAMI,OpCpHiF,0BoCoHtE,CpCgH2B,KAAL,GAAiB,KApO+B,EoCoHjE,CpCgHsB,KAAL,GAAiB,KApO +B,CoCoHtE,KAAJ,GAAY,CAAZ,GAAM,B,C;K;mFAG9B,yB;MAAA,8C;MAAA,0B;QAOI,OAAO,MAAM,CA AN,EAAS,MAAM,CAAN,EAAS,CAAT,CAAT,C;O;KAPX,C;mFAUA,yB;MAAA,8C;MAAA,0B;QAOI,OAAO, MAAM,CAAN,EAAS,MAAM,CAAN,EAAS,CAAT,CAAT,C;O;KAPX,C;mFAUA,yB;MAAA,8C;MAAA,0B;QA OI,OAAO,MAAM,CAAN,EAAS,MAAM,CAAN,EAAS,CAAT,CAAT,C;O;KAPX,C;IAUA,4B;MAOc ,Q;MADV,UAAU,C;MACA,uB;MAAV,OAAU,cAAV,C;QAAU,mB;QAAO,MAAM,SAAM,GAAN,EAAW,CAA X,C;;MACvB,OAAO,G;K;IAGX,4B;MAOc,Q;MADV,UAAU,C;MACA,uB;MAAV,OAAU,cAAV,C;QAAU,mB; QAAO,MAAM,SAAM,GAAN,EAAW,CAAX,C;;MACvB,OAAO,G;K;IAGX,4B;MAOc,Q;MADV,UAAU,C;MA CA,uB;MAAV,OAAU,cAAV,C;QAAU,mB;QAAO,MAAM,SAAM,GAAN,EAAW,CAAX,C;;MACvB,OAAO,G; K;IAGX,4B;MAOc,Q;MADV,UAAU,C;MACA,uB;MAAV,OAAU,cAAV,C;QAAU,mB;QAAO,MAAM,SAAM, GAAN,EAAW,CAAX,C;;MACvB,OAAO,G;K;gFC7OX,yB;MAAA,mC;MAAA,2C;MAAA,4B;QASI,OAAO,kB AAO,cAAP,C;O;KATX,C;gFAYA,yB;MAAA,mC;MAAA,2C;MAAA,4B;QASI,OAAO,kBAAO,cAAP,C;O;KAT X,C;IAYA,sC;;QASQ,OAAc,WAAP,MAAO,EAAS,SAAT,C;;QACHb,+C;UACE,MAAM,2BAAuB,CAAE,QAAz B,C;;UAHV,O;;K;IAOJ,sC;;QASQ,OAAc,YAAP,MAAO,EAAU,SAAV,C;;QACHb,+C;UACE,MAAM,2BAAuB,

CAAE,QAAzB,C;;UAHV,O;;K;4FAOJ,yB;MAAA,mC;MAAA,uD;MAAA,4B;QAOI,OAAO,wBAAa,cAAb,C;O; KAPX,C;4FAUA,yB;MAAA,mC;MAAA,uD;MAAA,4B;QAOI,OAAO,wBAAa,cAAb,C;O;KAPX,C;IAUA,4C;M AMI,IAAI,mBAAJ,C;QACI,OAAO,I;MACX,OAAc,WAAP,MAAO,EAAS,SAAT,C;K;IAGIB,4C;MAMI,IAAI,m BAAJ,C;QACI,OAAO,I;MACX,OAAc,YAAP,MAAO,EAAU,SAAV,C;K;oFAGIB,8B;MASI,OAAO,WAAW,IA AX,IAAmB,2BAAS,OAAT,C;K;oFAG9B,8B;MASI,OAAO,WAAW,IAAX,IAAmB,2BAAS,OAAT,C;K;IAG9B,u C;MAMI,OAAO,2BvC4K4B,SuC5KnB,KvC4K6B,KAAL,GAAiB,GAAtB,CuC5K5B,C;K;IAGX,uC;MAMI,OAA O,2BvC6K8B,UAAW,oBuC7KhC,KvC6K2B,KAAK,CAAL,UAAN,CuC7K9B,C;K;IAGX,uC;MAMI,OAAO,2Bt CwL8B,UAAW,oBsCxLhC,KtCwL2B,KAAK,CAAL,iBAAN,CsCxL9B,C;K;IAGX,uC;MAMY,Q;MAAD,cAAC, OtBqF4C,UsBrF5C,KtBqFkD,yBsBrFxC,EtBqFwC,CAAN,CsBrF7C,wBAA8B,2BAA9B,Q;MAAA,W;QAAqC,o CtCoPR,SsCpPiB,KtB6KIB,KhBuEW,QAAV,CsCpPQ,C;;MAA5C,a;K;IAGJ,uC;MAMI,OAAO,2BrCyI4B,SqCzI nB,KrCyI6B,KAAL,GAAiB,KAAtB,CqCzI5B,C;K;IAGX,uC;MAMI,OAAO,2BrC0I8B,UAAW,oBqC1IhC,KrC0I 2B,KAAK,CAAL,YAAN,CqC1I9B,C;K;IAGX,kC;MASI,OAAO,uCAAgB,yBvCmHY,SuCnHI,SvCmHM,KAAL, GAAiB,GAAtB,CuCnHZ,EvCmHY,SuCnHmB,EvCmHT,KAAL,GAAiB,GAAtB,CuCnHZ,EAA4C,EAA5C,C;K;I AG3B,kC;MASI,OAAO,uCAAgB,yBAAGB,SAAhB,EAAsB,EAAtB,EAA0B,EAA1B,C;K;IAG3B,kC;MASI,OAA O,wCAAiB,yBAAGB,SAAhB,EAAsB,EAAtB,M;K;IAG5B,kC;MASI,OAAO,uCAAgB,yBrCgFY,SqChFI,SrCgF M,KAAL,GAAiB,KAAtB,CqChFZ,ErCgFY,SqChFmB,ErCgFT,KAAL,GAAiB,KAAtB,CqChFZ,EAA4C,EAA5C, C;K;IAG3B,gC;MAMI,OAAO,uCAAgB,yBAAGB,cAAhB,EAAsB,eAAtB,EAA6B,CAAC,cAAD,IAA7B,C;K;IA G3B,gC;MAMI,OAAO,wCAAiB,yBAAGB,cAAhB,EAAsB,eAAtB,EAA8B,cAAD,aAA7B,C;K;IAG5B,iC;MAMI, oBAAoB,OAAO,CAA3B,EAA8B,IAA9B,C;MACA,OAAO,uCAAgB,yBAAGB,eAAhB,EAAuB,cAAvB,EAAiC,S AAK,KAAL,GAAy,CAAhB,GAAmB,IAAnB,GAA6B,CAAC,IAAD,IAA1D,C;K;IAG3B,iC;MAMI,oBAAoB,kB AAO,CAA3B,EAA8B,IAA9B,C;MACA,OAAO,wCAAiB,yBAAGB,eAAhB,EAAuB,cAAvB,EAAiC,SAAK,KAA L,cAAy,CAAhB,GAAmB,IAAnB,GAA8B,IAAD,aAA1D,C;K;IAG5B,iC;MAQI,IvC/OgF,0BuC+O5E,EvCjKc,K AAL,GAAiB,GA3O8B,EUc+Ote,6BAAM,UvCjSb,KAAL,GAAiB,GA3O8B,CuC+O5E,KAAJ,C;QAA2B,OAAO, iCAAU,M;MACHc,WvC6BuB,SuC7B5B,SvC6BsC,KAAL,GAAiB,GAAtB,C;MuC7BV,YAAK,W;MAA9B,OtCj D6D,oBAhJP,SAAU,CD8N7B,SuC7BV,EvC6BoB,KAAL,GAAiB,GAAtB,CC9N6B,MAAK,GDAK,KCAO,KAA Z,IAAf,CAgJO,C;K;IsCoDjE,iC;MAQI,ItC3OkE,YsC2O9D,EtC3O+E,KAAjB,EsC2OxD,4BAAK,UtC3OgF,KAA 7B,CsC2O9D,KAAJ,C;QAA0B,OAAO,iCAAU,M;MAC3C,OtC7D6D,csC6DtD,StC7DsD,EahJP,SsC6MtC,EtC7 MgD,KAAK,GAAy,CsC6M5D,WtC7M4D,MAAZ,IAAf,CAgJO,C;K;IsCgEjE,iC;MAQI,ItB/OmE,asB+O/D,EtB/ OiF,KAAiB,EsB+OzD,6BAAM,UtB/OiF,KAA9B,CsB+O/D,KAAJ,C;QAA2B,OAAO,kCAAW,M;MAC7C,OtBzE +D,iBsByExD,StBzEwD,EA7IP,UsBsNx,C,EtBtNmD,KAAK,UAAy,ChByP/C,UAAW,oBAAL,CsCnCb,WtCmC sB,MAAK,CAAL,iBAAN,CgBzP+C,MAAZ,CAAhB,CA6IO,C;K;IsB4EnE,iC;MAQI,IrC3QiF,0BqC2Q7E,ErCvC kC,KAAL,GAAiB,KApO+B,EqC2QvE,8BAAO,UrCvCqB,KAAL,GAAiB,KApO+B,CqC2Q7E,KAAJ,C;QAA4B, OAAO,iCAAU,M;MACjC,WrCnuB,SqCM5B,SrCnS,C,KAAL,GAAiB,KAAtB,C;MqCMV,YAAK,W;MAA9B,Ot CrF6D,oBAhJP,SAAU,CC+N7B,SqCMV,ErCNoB,KAAL,GAAiB,KAAtB,CD/N6B,MAAK,GCAK,KDAO,KAA Z,IAAf,CAgJO,C;K;IsCwFjE,kD;MAUI,OtCjRkE,YsCiRvD,StCjRwE,KAAjB,EsCiRhD,YtCjR6E,KAA7B,CsCiR vD,IAAJ,GAAyB,YAAzB,GAA2C,S;K;IAGtD,kD;MAUI,OtBtRmE,asBsRxD,StBtR0E,KAAiB,EsBsRjD,YtBtR+ E,KAA9B,CsBsRxD,IAAJ,GAAyB,YAAzB,GAA2C,S;K;IAGtD,kD;MAUI,OvC3TgF,0BuC2TrE,SvChF2B,KAA L,GAAiB,GA3O8B,EUc2T9D,YvChFoB,KAAL,GAAiB,GA3O8B,CuC2TrE,IAAJ,GAAyB,YAAzB,GAA2C,S;K; IAGtD,kD;MAUI,OrChUiF,0BqCgUtE,SrC5F2B,KAAL,GAAiB,KApO+B,EqCgU/D,YrC5FoB,KAAL,GAAiB,K ApO+B,CqCgUtE,IAAJ,GAAyB,YAAzB,GAA2C,S;K;IAGtD,iD;MAUI,OtCrUkE,YsCqUvD,StCrUwE,KAAjB,E sCqUhD,YtCrU6E,KAA7B,CsCqUvD,IAAJ,GAAyB,YAAzB,GAA2C,S;K;IAGtD,iD;MAUI,OtB1UmE,asB0UxD, StB1U0E,KAAiB,EsB0UjD,YtB1U+E,KAA9B,CsB0UxD,IAAJ,GAAyB,YAAzB,GAA2C,S;K;IAGtD,iD;MAUI,O vC/WgF,0BuC+WrE,SvCpI2B,KAAL,GAAiB,GA3O8B,EUc+W9D,YvCpIoB,KAAL,GAAiB,GA3O8B,CuC+Wr E,IAAJ,GAAyB,YAAzB,GAA2C,S;K;IAGtD,iD;MAUI,OrCpXiF,0BqCoXtE,SrChJ2B,KAAL,GAAiB,KApO+B,E qCoX/D,YrChJoB,KAAL,GAAiB,KApO+B,CqCoXtE,IAAJ,GAAyB,YAAzB,GAA2C,S;K;IAGtD,4D;MAUI,ItCz XkE,YsCyX9D,YtCzX+E,KAAjB,EsCyX/C,YtCzX4E,KAA7B,CsCyX9D,IAAJ,C;QAAiC,MAAM,gCAAyB,oDA AiD,YAAjD,8BAAoF,YAApF,MAAzB,C;MACvC,ItC1XkE,YsC0X9D,StC1X+E,KAAjB,EsC0XvD,YtC1XoF,K AA7B,CsC0X9D,IAAJ,C;QAAyB,OAAO,Y;MACHc,ItC3XkE,YsC2X9D,StC3X+E,KAAjB,EsC2XvD,YtC3XoF,

AAI,SAAQ,WAAZ,C;QAA2B,MAAa,gCAAyB,wEAAzB,C;MAG5C,aAGyB,K;MAEzB,YAGuF,OAA/D,0BAA0
B,KpBcR,IoBdlB,EAAc,YpBcpB,IoBdlB,EAAYd,IAAZD,CAA+D,C;MAEvF,YAGuB,I;K;yCAEvB,Y;MAAwC,
mCAAwB,UAAxB,EAA+B,SAA/B,EAAqC,SAArC,C;K;wCAExC,Y;MAMqC,OAAI,YAAO,CAAX,GAAc,AA
Q,SAAtB,GAAgC,AAQ,S;K;uCAE7E,iB;MACI,iDAA6B,kBAAa,KAAM,UAAAnB,KAC7B,eAAS,KAAM,MAAf
,IAAwB,cAAQ,KAAM,KAAtC,IAA8C,cAAQ,KAAM,KAD/B,CAA7B,C;K;yCAGJ,Y;MACI,OAAI,cAAJ,GAAe,
EAAf,GAAwB,OAAM,OAkK,UpBRG,IoBQR,UAAkB,SpBRV,IoBQR,KAAN,SAAqC,SAArC,I;K;yCAE5B,Y;
MAAkC,OAAI,YAAO,CAAX,GAAc,oBAAE,UAAF,+BAAU,SAAV,eAAqB,SAAnC,GAA8C,oBAAE,UAAF,qC
AAgB,SAAhB,gBAA4B,CAAC,SAAD,IAA5B,C;K;IAEHf,qC;MAAA,yC;K;KEACI,sC;MAQ2F,2BAAgB,UAAh
B,EAA4B,QAA5B,EAAc,IAAtC,C;K;;;IAT/F,iD;MAAA,gD;QAAA,+B;;MAAA,yC;K;;IAiBA,mD;MA6CA,sC;
MAtCI,IAAI,SAAQ,CAAZ,C;QAae,MAAa,gCAAyB,wBAAzB,C;MAC5B,IAAI,SAAQ,WAAZ,C;QAA2B,MAA
a,gCAAyB,wEAAzB,C;MAG5C,aAGwB,K;MAExB,YAGuB,0BAA0B,KAA1B,EAAiC,YAAjC,EAA+C,IAA/C,C
;MAEvB,YAGuB,I;K;wCAEvB,Y;MAAuC,kCAAUb,UAAvB,EAA8B,SAA9B,EAAoC,SAApC,C;K;uCAEvC,Y;
MAMqC,OAAI,YAAO,CAAX,GAAc,AAQ,SAAtB,GAAgC,AAQ,S;K;sCAE7E,iB;MACI,gDAA4B,kBAAa,KA
AM,UAAAnB,KAC5B,eAAS,KAAM,MAAf,IAAwB,cAAQ,KAAM,KAAtC,IAA8C,cAAQ,KAAM,KADhC,CAA5
B,C;K;wCAGJ,Y;MACI,OAAI,cAAJ,GAAe,EAAf,GAAwB,OAAM,MAAK,UAAAL,QAAa,SAAb,IAAN,SAA2B,S
AA3B,I;K;wCAE5B,Y;MAAkC,OAAI,YAAO,CAAX,GAAgB,UAAF,qBAAU,SAAV,cAAqB,SAAnC,GAAgD,U
AAF,2BAAgB,SAAhB,eAA4B,CAAC,SAAD,IAA5B,C;K;IAEHf,oC;MAAA,wC;K;IEACI,sC;MAQwF,0BAAe,U
AAf,EAA2B,QAA3B,EAAqC,IAArC,C;K;;;IAT5F,gD;MAAA,+C;QAAA,8B;;MAAA,wC;K;;IAiBA,oD;MA6CA,
uC;MAtCI,IAAI,gBAAJ,C;QAAGB,MAAa,gCAAyB,wBAAzB,C;MAC7B,IAAI,sCAAJ,C;QAA4B,MAAa,gCAAy
B,yEAAzB,C;MAG7C,aAGyB,K;MAEzB,YAGwB,4BAA0B,KAA1B,EAAiC,YAAjC,EAA+C,IAA/C,C;MAExB,
YAGwB,I;K;yCAExB,Y;MAAwC,mCAAwB,UAAxB,EAA+B,SAA/B,EAAqC,SAArC,C;K;wCAExC,Y;MAMqC
,OAAI,uBAAO,CAAX,GAAc,2BAAQ,SAAR,KAAd,GAAgC,2BAAQ,SAAR,K;K;uCAE7E,iB;MACI,iDAA6B,k
BAAa,KAAM,UAAAnB,KAC7B,mBAAS,KAAM,MAAf,KAAwB,kBAAQ,KAAM,KAAd,CAAxB,IAA8C,kBAA
Q,KAAM,KAAd,CADjB,CAA7B,C;K;yCAGJ,Y;MACI,OAAI,cAAJ,GAAe,EAAf,GAAwB,iCAAM,iCAAM,eAA
W,8BAAW,EAAX,CAAX,CAAN,MAAoC,cAAU,6BAAU,EAAV,CAAV,CAApC,CAAN,MAAuE,cAAU,6BAA
U,EAAV,CAAV,CAAvE,CAAiG,Q;K;yCAE7H,Y;MAAkC,OAAI,uBAAO,CAAX,GAAgB,UAAF,qBAAU,SAA
V,yBAAqB,SAArB,WAAAd,GAAgD,UAAF,2BAAgB,SAAhB,yBAA6B,SAAD,AA5B,W;K;IAEHf,qC;MAAA,y
C;K;KEACI,sC;MAQ4F,2BAAgB,UAAhB,EAA4B,QAA5B,EAAc,IAAtC,C;K;;;IAThG,iD;MAAA,gD;QAAA,+
B;;MAAA,yC;K;;;6CCIKa,iB;MAGkD,+BAAS,UAAAT,UAAkB,wBAAS,iBAAT,M;K;oCAEpE,Y;MAKgC,oCA
AQ,iBAAR,K;K;;I7CpBd,wC;MAkBIB,iC;MAtBsD,2BAAgB,KAAhB,EAAuB,YAAvB,EAAqC,CAArC,C;K;kFA
C7B,Y;MAAQ,8B;K;yFACD,Y;MAAQ,6B;K;2CAExC,iB;MAA8C,qBAAS,KAAT,IAAkB,SAAS,S;K;kCAEzE,Y
;MAKkC,oBAAQ,S;K;iCAE1C,iB;MACI,2CAAuB,kBAAa,KAAM,UAAAnB,KACvB,eAAS,KAAM,MAAf,IAAw
B,cAAQ,KAAM,KADf,CAAvB,C;K;mCAGJ,Y;MACI,OAAI,cAAJ,GAAe,EAAf,GAAwB,OAkK,UwBkBS,IxBI
Bd,UAAkB,SwBkJ,IxBIBd,K;K;mCAE5B,Y;MAAkC,2BAAE,UAAF,+BAAU,SAAV,C;K;IAEIC,+B;MAAA,m
C;MACI,aAC8B,cAAy,OAAF,CAAE,CAAZ,EAAwB,OAAF,CAAE,CAAxB,C;K;;;IAFIC,2C;MAAA,0C;QAAA,
yB;;MAAA,mC;K;;IASiB,uC;MAkBjB,gC;MAtBmD,0BAAE,KAAf,EAAcB,YAAAtB,EAAoC,CAApC,C;K;iFAC3
B,Y;MAAQ,iB;K;wFACD,Y;MAAQ,gB;K;0CAEvC,iB;MAA6C,qBAAS,KAAT,IAAkB,SAAS,S;K;iCAExE,Y;M
AKkC,oBAAQ,S;K;gCAE1C,iB;MACI,0CAAsB,kBAAa,KAAM,UAAAnB,KACtB,eAAS,KAAM,MAAf,IAAwB,c
AAQ,KAAM,KADhB,CAAtB,C;K;kCAGJ,Y;MACI,OAAI,cAAJ,GAAe,EAAf,GAAwB,MAAK,UAAAL,QAAa,SA
Ab,I;K;kCAE5B,Y;MAAkC,OAAE,UAAF,qBAAU,S;K;IAE5C,8B;MAAA,kC;MACI,aAC6B,aAAS,CAAT,EAA
Y,CAAZ,C;K;;;IAFjC,0C;MAAA,yC;QAAA,wB;;MAAA,kC;K;;IASkB,wC;MAkBIB,iC;MAtBsD,2BAAgB,KAA
hB,EAAuB,YAAvB,K;K;kFAC7B,Y;MAAQ,iB;K;yFACD,Y;MAAQ,gB;K;2CAExC,iB;MAA8C,kCAAS,KAAT,
UAAkB,sBAAS,SAAT,M;K;kCAEHf,Y;MAKkC,kCAAQ,SAAR,K;K;iCAEIC,iB;MACI,2CAAuB,kBAAa,KAA
M,UAAAnB,KACvB,mBAAS,KAAM,MAAf,KAAwB,kBAAQ,KAAM,KAAd,CADD,CAAvB,C;K;mCAGJ,Y;MA
CI,OAAI,cAAJ,GAAe,EAAf,GAAwB,iCAAM,eAAW,8BAAW,EAAX,CAAX,CAAN,MAAoC,cAAU,6BAAU,E
AAV,CAAV,CAApC,CAA8D,Q;K;mCAEIF,Y;MAAkC,OAAE,UAAF,qBAAU,SAAV,W;K;IAEIC,+B;MAAA,m
C;MACI,aAC8B,qB;K;;;IAFIC,2C;MAAA,0C;QAAA,yB;;MAAA,mC;K;;I8C9EJ,gB;MAAA,oB;K;8BAIL,Y;MA
A0B,oB;K;;;IAJ9B,4B;MAAA,2B;QAAA,U;;MAAA,oB;K;ICEA,yC;MAAA,e;MAAA,iB;MAAA,uB;K;IAAA,uC

;MAAA,0C;O;MAIL,kE;MAEA,wF;MAEA,oF;MAEA,wE;MAEA,kE;MAEA,oF;MAEA,sF;MAEA,8E;MAEA,wE
;MAEA,sF;MAEA,uF;MAEA,iE;MAEA,6E;MAEA,iE;MAEA,2E;K;;IA5BA,8C;MAAA,6B;MAAA,sC;K;;IAEA,
yD;MAAA,6B;MAAA,iD;K;;IAEA,uD;MAAA,6B;MAAA,+C;K;;IAEA,iD;MAAA,6B;MAAA,yC;K;;IAEA,8C;M
AAA,6B;MAAA,sC;K;;IAEA,uD;MAAA,6B;MAAA,+C;K;;IAEA,wD;MAAA,6B;MAAA,gD;K;;IAEA,oD;MAA
A,6B;MAAA,4C;K;;IAEA,iD;MAAA,6B;MAAA,yC;K;;IAEA,wD;MAAA,6B;MAAA,gD;K;;IAEA,wD;MAAA,6
B;MAAA,gD;K;;IAEA,6C;MAAA,6B;MAAA,qC;K;;IAEA,mD;MAAA,6B;MAAA,2C;K;;IAEA,6C;MAAA,6B;
MAAA,qC;K;;IAEA,kD;MAAA,6B;MAAA,0C;K;;IAhCJ,mC;MAAA,+oB;K;;IAAA,wC;MAAA,a;aAAA,O;UAA
A,2C;aAAA,kB;UAAA,sD;aAAA,gB;UAAA,oD;aAAA,U;UAAA,8C;aAAA,O;UAAA,2C;aAAA,gB;UAAA,oD;a
AAA,iB;UAAA,qD;aAAA,a;UAAA,iD;aAAA,U;UAAA,8C;aAAA,iB;UAAA,qD;aAAA,iB;UAAA,qD;aAAA,M;
UAAA,0C;aAAA,Y;UAAA,gD;aAAA,M;UAAA,0C;aAAA,W;UAAA,+C;;UAAA,uE;;K;;IAqCA,4C;MAAA,e;M
AAA,iB;MAAA,uB;K;IAAA,0C;MAAA,6C;O;MAMI,0E;MAEA,0E;MAEA,4E;K;;IAJA,kD;MAAA,gC;MAAA,0
C;K;;IAEA,kD;MAAA,gC;MAAA,0C;K;;IAEA,mD;MAAA,gC;MAAA,2C;K;;IAVJ,sC;MAAA,sI;K;;IAAA,2C;M
AAA,a;aAAA,Q;UAAA,+C;aAAA,Q;UAAA,+C;aAAA,S;UAAA,gD;;UAAA,0E;;K;;IAwB8B,gC;MAAC,oC;K;;I
AQE,0B;MAAC,qB;QAAA,iD;MAAA,kB;K;;IAEIC,sB;K;;IAMA,4B;K;;IC/EA,yB;K;;IAQA,6B;K;;ICnBA,mB;
MAEI,UAAU,IAAI,C;MACd,OAAW,OAAO,CAAX,GAAC,GAAd,GAAuB,MAAM,CAAN,I;K;IAGIC,qB;MACI,
UAAU,SAAI,CAAJ,C;MACV,OAAW,kBAAO,CAAX,GAAC,GAAd,GAAuB,QAAM,CAAN,C;K;IAGIC,mC;MA
EI,OAAO,IAAI,IAAI,CAAJ,EAAO,CAAP,IAAY,IAAI,CAAJ,EAAO,CAAP,CAAZ,IAAJ,EAA2B,CAA3B,C;K;I
AGX,qC;MACI,OAAO,MAAI,MAAI,CAAJ,EAAO,CAAP,WAAy,MAAI,CAAJ,EAAO,CAAP,CAAZ,CAAJ,EA
A2B,CAA3B,C;K;IAGX,qD;MAkBI,WAAO,CAAP,C;QAD2E,OAC3D,SAAS,GAAb,GAakB,GAAIB,GAA2B,M
AAM,iBAaIB,GAAjB,EAA6B,IAA7B,CAAN,I;WACvC,WAAO,CAAP,C;QAF2E,OAE3D,SAA
S,GAAb,GAakB,GAAIB,GAA2B,MAAM,iBAaIB,KAAjB,EAAwB,GAAxB,EAA6B,CAAC,IAAD,IAA7B,CAA
N,I;;QAC/B,MAAa,gCAAyB,eAAzB,C;K;IAGzB,uD;MAkBI,sBAAO,CAAP,C;QAD+E,OAC/D,sBAAS,GAAT,
MAAJ,GAAkB,GAAIB,GAA2B,aAAM,mBAaIB,GAAjB,EAA6B,IAA7B,CAAN,C;WACvC,sB
AAO,CAAP,C;QAF+E,OAE/D,sBAAS,GAAT,MAAJ,GAAkB,GAAIB,GAA2B,QAAM,mBAaIB,KAAjB,EAAw
B,GAAxB,EAA8B,IAAD,aAA7B,CAAN,C;;QAC/B,MAAa,gCAAyB,eAAzB,C;K;IC7DjB,kD;MAAA,8B;MACI,
aAAY,C;K;oDACZ,Y;MAAyB,oBAAQ,gBAaI,O;K;iDACrC,Y;MAAgD,Q;MAA1B,IAAI,aAAQ,gBAaI,OAAhB
,C;QAAA,OAA6B,iBAaI,iBAAJ,EAAI,yBAAJ,O;;QAAkB,MAAM,2BAAYB,UAAF,WAAvB,C;K;;IAPhF,oC;M
AEI,IAD8D,IAC9D,S;QACI,UAA0B,K;QAF0B,2C;;QAAA,QAAM,IAAN,C;eASxD,c;YATwD,OAStC,qBAaQb,
KAArB,C;eACIB,W;YAVwD,OAUzC,kBAaKb,KAAIB,C;eACf,Y;YAXwD,OAWxC,mBAaMB,KAArB,C;eAC
hB,W;YAZwD,OAYzC,kBAaKb,KAAIB,C;eACf,U;YAbwD,OAA1C,iBAaIB,KAAjB,C;eACd,W;YAdwD,OAcz
C,kBAaKb,KAAIB,C;eACf,Y;YAfWd,OAeXC,mBAaMB,KAArB,C;eACbB,a;YAhBwD,OAgBvC,oBAaOB,KA
ApB,C;;YACT,MAAM,6BAASB,2DAA+C,IAA/C,CAAtB,C;;K;IAIuC,2D;MAAA,kC;MAAS,0B;MAC9D,aAAY,
C;K;2DACZ,Y;MAAyB,oBAAQ,kBAAM,O;K;+DACvC,Y;MAA2D,Q;MAA9B,IAAI,aAAQ,kBAAM,OAAIB,C;
QAAA,OAAwB,mBAAM,iBAAN,EAAM,yBAAN,O;;QAAoB,MAAM,2BAAYB,UAAF,WAAvB,C;K;;IAJnF,qC;
MACyD,oD;K;IAON,wD;MAAA,kC;MAAS,uB;MACxD,aAAY,C;K;wDACZ,Y;MAAyB,oBAAQ,kBAAM,O;K;
yDACvC,Y;MAAwD,Q;MAA9B,IAAI,aAAQ,kBAAM,OAAIB,C;QAAA,OAAwB,mBAAM,iBAAN,EAAM,yBA
AN,O;;QAAoB,MAAM,2BAAYB,UAAF,WAAvB,C;K;;IAJhF,kC;MACmD,iD;K;IAOE,yD;MAAA,kC;MAAS,w
B;MAC1D,aAAY,C;K;yDACZ,Y;MAAyB,oBAAQ,kBAAM,O;K;2DACvC,Y;MAAyD,Q;MAA9B,IAAI,aAAQ,k
BAAM,OAAIB,C;QAAA,OAAwB,mBAAM,iBAAN,EAAM,yBAAN,O;;QAAoB,MAAM,2BAAYB,UAAF,WAA
vB,C;K;;IAJf,mC;MACqD,kD;K;IAOF,wD;MAAA,kC;MAAS,uB;MACxD,aAAY,C;K;wDACZ,Y;MAAyB,oBA
AQ,kBAAM,O;K;yDACvC,Y;MAAwD,Q;MAA9B,IAAI,aAAQ,kBAAM,OAAIB,C;QAAA,OAAwB,mBAAM,iB
AAN,EAAM,yBAAN,O;;QAAoB,MAAM,2BAAYB,UAAF,WAAvB,C;K;;IAJhF,kC;MACmD,iD;K;IAOF,uD;MA
AA,kC;MAAS,sB;MACiD,aAAY,C;K;uDACZ,Y;MAAyB,oBAAQ,kBAAM,O;K;uDACvC,Y;MAAuD,Q;MAA9
B,IAAI,aAAQ,kBAAM,OAAIB,C;QAAA,OAAwB,mBAAM,iBAAN,EAAM,yBAAN,O;;QAAoB,MAAM,2BAAY
B,UAAF,WAAvB,C;K;;IAJ/E,iC;MACiD,gD;K;IAOI,yD;MAAA,kC;MAAS,wB;MAC1D,aAAY,C;K;yDACZ,Y;
MAAyB,oBAAQ,kBAAM,O;K;2DACvC,Y;MAAyD,Q;MAA9B,IAAI,aAAQ,kBAAM,OAAIB,C;QAAA,OAAwB
,mBAAM,iBAAN,EAAM,yBAAN,O;;QAAoB,MAAM,2BAAYB,UAAF,WAAvB,C;K;;IAJf,mC;MACqD,kD;K;I
AOE,0D;MAAA,kC;MAAS,yB;MAC5D,aAAY,C;K;0DACZ,Y;MAAyB,oBAAQ,kBAAM,O;K;6DACvC,Y;MAA

0D,Q;MAA9B,IAAI,aAAQ,kBAAM,OAAIB,C;QAAA,OAAwB,mBAAM,iBAAN,EAAM,yBAAN,O;;QAAoB,M
AAM,2BAAYB,UAAF,WAAvB,C,K;;IAJIF,oC;MACuD,mD;K;IAOJ,wD;MAAA,kC;MAAS,uB;MACxD,aAAY,
C;K;wDACZ,Y;MAAYB,oBAAQ,kBAAM,O;K;yDACvC,Y;MAAwD,Q;MAA9B,IAAI,aAAQ,kBAAM,OAAIB,C
;QAAA,OAAwB,mBAAM,iBAAN,EAAM,yBAAN,O;;QAAoB,MAAM,2BAAYB,UAAF,WAAvB,C,K;;IAJhF,kC
;MACmD,iD;K;IAOpB,gC;MAAC,wB;K;;IAEhC,+B;MAC8C,MAAM,mC;K;IAEpD,8C;MAEI,IAAI,qBAAJ,C;Q
ACI,OAAO,C5ByIiF,W4BzIzE,U5ByIqE,E4BzIzD,Q5ByIyD,C;;Q4BvIxF,OAAS,CAAY,qBAAsB,UAAtB,EAAk
C,QAAIC,C;;K;IAI7B,2C;MAEI,IAAI,KAAy,kBAAhB,C;QAGI,KAAy,mBAAkB,QAAIB,C;;QAEH,QAAT,SA
A+C,CAAIB,IAAjC,KAAiC,EAAkB,O;;K;IAIvD,sC;MAGwB,Q;MADpB,gBAAgB,IAAhB,KAAgB,E;MACI,IA
AI,OCnGkB,ODmGT,OAAT,EAAqB,WAArB,CAAJ,C;QChB,OAAI,aAAJ,GAAMB,KAAM,WAAzB,GAAyC,I
;;QAEzC,c;;MAHJ,wB;MAKA,kBAAkB,K;MACIB,iBAAiB,W;MACjB,OAAO,S;K;IAIa,sB;MAAC,U;K;iCACrB
,iB;MACI,OAAO,mCAAsB,WAAK,KAAM,E;K;mCAG5C,Y;MACI,OAAO,M;K;mCAGX,Y;MACI,OAAuC,oB
AAAnB,UAA5B,IAAe,EAAa,CAAMb,C;K;OCAG3C,iB;MACI,OAAO,IAAI,EAAW,GAAN,K;K;kCAGL,Y;MAEI,
OAAO,M;K;;+DAIf,gB;MAEI,YAAy,MAAY,IAAK,OAAjB,C;MACZ,sBAAU,IAAV,a;QACI,UAAU,KAAK,CA
AL,C;QACV,IAAI,oBAAJ,C;UACI,MAAM,CAAN,IAAW,EAAS,MAAM,MAAK,GAAL,C;;UAE1B,MAAM,CA
AN,IAAW,G;;MAGnB,OAAO,EAAS,OAAO,OAAM,EAAN,EAAgB,KAAhB,C;K;IAG3B,2B;MAMW,WAAO,S
;MAIBd,YAAy,MAAY,IAAK,OAAjB,C;MACZ,sBAAU,IAAV,a;QACI,UAAU,KAAK,CAAL,C;QACV,IAAI,oB
AAJ,C;UACI,MAAM,CAAN,IAAW,EAAS,MAAM,MAAK,GAAL,C;;UAE1B,MAAM,CAAN,IAAW,G;;MAYn
B,OATO,EAAS,OAAO,OAAM,EAAN,EAAgB,KAAhB,C;K;IAY3B,oC;MAWI,WAAqB,S;MACrB,IAAI,qBAA
mB,CAAY,OAAoB,KAA2B,SAAhD,C;QAJCA,YAAy,MAkCM,IAICW,OAAjB,C;QACZ,sBAiCkB,IAjCIB,a;UAC
I,UAgCc,IAhCJ,CAAK,CAAL,C;UACV,IAAI,oBAAJ,C;YACI,MAAM,CAAN,IAAW,EAAS,MAAM,MAAK,GA
AL,C;;YAE1B,MAAM,CAAN,IAAW,G;;QA4Bf,OAZBG,EAAS,OAAO,OAAM,EAAN,EAAgB,KAAhB,C;;QA2
BnB,WAAW,C;QACX,OBAAU,IAAV,e;UACY,IAAoB,I;UAA5B,eAAQ,QAAoB,OAApB,IAAQ,CAAH,GAAG,
CAAY,OAApB,oCAAR,K;;QAEJ,aAAa,IAAjB,CAAC,YAAgB,CAAH,IAAG,C;QE3FjB,IF4FyB,CE5FhB,OAL
,KAAkB,SAAtB,C;UF4F4B,ME3FxB,UF2FqB,CE3FF,O;;QF4FnB,OAAO,C;QACP,OBAAU,IAAV,e;UAE0B,YA
CX,M;UAFX,YAAU,IAAQ,CAAH,GAAG,C;UACI,SAAJ,KAAI,O;UAAtB,aAAU,CAAV,kB;YACI,OAAO,aAA
P,EAAO,qBAAP,YAAiB,MAAI,CAAJ,C;;QAGzB,OAAO,M;;K;IAIf,0B;MACgC,WAAS,c;MAAT,YAAhC,EAA
E,MAAM,KAAiD,CAA3C,SAA2C,C;MAWrD,eAAiB,I;MAXW,OAYrB,K;K;IAVX,uB;MAC6B,WAAS,W;MAA
T,YAAsB,IAA/C,WAA+C,CAAnC,EAAE,MAAM,KAAK,CAAC,SAAD,CAAsB,C;MAQ/C,eAAiB,I;MARQ,OA
SIB,K;K;IAPX,uB;MAC6B,WAAS,W;MAAT,YAA7B,EAAE,MAAM,KAA2C,CAArC,SAAqC,C;MAK/C,eAAiB
,I;MALQ,OAMIB,K;K;2DAJX,uB;MAGI,eAAiB,I;MACjB,OAAO,K;K;KEG9MX,yB;MAAA,0B;MAAA,uB;QAS
I,OAAoB,OAAb,ItD0Q+B,KAAAL,GAaiB,KsD1Q9B,C;O;KATxB,C;ICiQ,C,2C;MAAC,8C;MACIC,eAAAsB,C;MA
CtB,wBAA+B,C;MAC/B,gBAA6B,I;MAC7B,mBAAsC,I;MACtC,qBAAyC,I;MAEzC,yBAAGD,yBAAMb,Q;MA
EnE,sBAAGD,I;K;wFAFhD,Y;MAAA,6B;K;0CAIA,Y;MAEY,kBADr,M;MAAA,U;MAAA,2C;QAAA,e;;QAES,
gBADD,2CAAQ,yCAAR,gDAAWD,IAAxD,6BAAiE,I;QACzD,sB1CwEd,S;Q0C1EF,S1C2EG,S;;M0C3EH,a;K;i
DAIJ,kB;MACI,kBAAC,IAAd,C;MACiC,oB;MCuBrB,Q;MADR,IdTBsB,MCsBtB,W;QADJ,mBACiB,I;;QADjB,
mBAEY,QDvBc,MCuBd,+D;;MDvBZ,yC;MACA,2BAAMC,MAAO,kBAA1C,C;MAGA,OAAO,IAAP,C;Q1CoC
Y,gB0CnCH,S;;QACD,iBAAiB,8B;QAGjB,IAAI,0BAAJ,C;UACI,qBAAC,e;;UAEd,oBAAQ,0B;UACR,wBAAy,k
B;;UAIZ,cAAc,oB;UACd,IAAI,YAAy,yBAAhB,C;YAAqC,M;UACrC,kBAAGB,O;UACHB,qBAAMb,I;;UAEnB
,kBAAGB,I;UACHB,qBAAMb,S;;QAGvB,gC;QAEA,IAAI,wCAAJ,C;UAEI,YAAU,U;;UAGV,U;UAAA,0C;YET
hB,8BDgDQ,WAAO,qBAAP,CChDR,C;YFSgB,a;;YAAA,a;UAAA,mB;YAEK,UEpBrB,oBDgDQ,WD5B+B,eC4
B/B,CChDR,C;;UFqBgB,M;;K;mDAMhB,Y;MACI,kBAAkB,mB;MACIB,IAAI,uBAAuB,gBAAGB,IAA3C,C;QA
CI,uCAAQ,yCAAR,EAAmC,wCAA+B,WAA/B,C;;MAEvC,sBAAoB,mC;K;;IAM5B,iC;MAAA,qC;K;gGAEQ,Y;
M7C0DyC,MAAM,6B6C1DjC,uC7C0D+D,WAA9B,C;K;yD6CxDnD,kB;M7CwD6C,MAAM,6B6CvDzC,uC7Cu
DuE,WAA9B,C;K;+C6CpDnD,Y;MAAkC,8C;K;;IARtC,6C;MAAA,4C;QAAA,2B;;MAAA,qC;K;IGyDA,mG;IA
AA,yH;IAAA,6F;MAKW,kC;MAAS,4C;K;IALpB,sEAMQ,Y;MACI,Q;MAAA,sC;QAAiB,U;;MACjB,OAAO,oB;
K;IARnB,6G;sJAJIA,iC;MAGBU,OAkK,SAAL,CAAiB,UAAjB,EAA6B,KAA7B,C;K;wJAEV,2C;MAiBU,OAkK
,SAAL,CAAiB,QAAjB,EAA2B,UAA3B,EAAuC,KAAvC,C;K;wJAEV,kD;MAKU,OAkK,SAAL,CAAiB,QAAjB,
EAA2B,KAA3B,EAAkC,UAAIC,EAA8C,KAA9C,C;K;IAGc6C,oG;MAAA,mB;QAC3C,OAkK,iCAAL,CAAiB,k

BAAjB,C;O;K;IA/BZ,6D;MA0BI,IAAS,SAAY,OAAjB,IAA2B,CAA/B,C;QAAA,OAES,SAAL,CAAiB,UAAjB,EAA6B,IAA7B,C;;QA8D0B,Q;QAhE9B,4DAImD,0DAJnD,EAgE8B,qBA5DS,UA4DT,qCAhE9B,C;;K;IAwCmD,wH;MAAA,mB;QAC3C,OAAC,iCAAL,CAAiB,gBAAjB,EAA2B,kBAA3B,C;O;K;IAhCZ,yE;MA2BI,IAAS,SAAY,OAAjB,IAA2B,CAA/B,C;QAAA,OAES,SAAL,CAAiB,QAAjB,EAA2B,UAA3B,EAAuC,IAAvC,C;;QA0B0B,Q;QA5B9B,4DAImD,sEAJnD,EA4B8B,qBAxBs,UAWBT,qCA5B9B,C;;K;IASJ,gC;MAWK,kBAAD,M;MAAA,kBAAC,qEAAD,4DAA2C,S;K;6CAG/C,yB;MAAA,mG;MAAA,yH;MAAA,6F;QAKW,kC;QAAS,4C;O;MALpB,sEAMQ,Y;QACI,Q;QAAA,sC;UAAiB,U;;QACjB,OAAO,oB;O;MARnB,6G;MAAA,oC;QAKkC,Q;QAA9B,mEA8B,oEAA9B,C;O;KALJ,C;iFC7HA,a;MAC6C,OAAA,MAAa,YAAW,CAAX,C;K;ICM3B,iC;;MAA6E,Q;MAAA,+BAAS,I;sCAAIB,O,2DAAA,O;;;K;;;IAC/F,2B;MAAA,iD;MAAuB,oBAAK,IAAL,EAAW,IAAX,C;MAAvB,Y;K;IACA,sC;MAAA,iD;MAAuC,oBAAK,OAAL,EAAC,IAAd,C;MAAvC,Y;K;IACA,oC;MAAA,iD;MAAwC,oBAAK,SAAL,EAAGB,KAAhB,C;MAAxC,Y;K;IAI+B,mC;;MAA6E,Q;MAAA,+BAAS,I;sCAAIB,O,2DAAA,O;;;K;;;IACnG,+B;MAAA,mD;MAAuB,sBAAK,IAAL,EAAW,IAAX,C;MAAvB,Y;K;IACA,0C;MAAA,mD;MAAuC,sBAAK,OAAL,EAAC,IAAd,C;MAAvC,Y;K;IACA,wC;MAAA,mD;MAAwC,sBAAK,SAAL,EAAGB,KAAhB,C;MAAxC,Y;K;IAGsC,0C;MAA0D,qBAAU,OAAV,EAAMB,KAAhB,C;;K;;IACgS,C;MAAA,0D;MAAuB,6BAAK,IAAL,EAAW,IAAX,C;MAAvB,Y;K;IACA,iD;MAAA,0D;MAAuC,6BAAK,OAAL,EAAC,IAAd,C;MAAvC,Y;K;IACA,+C;MAAA,0D;MAAwC,6BAAK,SAAL,EAAGB,KAAhB,C;MAAxC,Y;K;IAG8C,kD;MAA0D,4BAAiB,OAAjB,EAA0B,KAA1B,C;;K;;IACxG,8C;MAAA,kE;MAAuB,qCAAK,IAAL,EAAW,IAAX,C;MAAvB,Y;K;IACA,yD;MAAA,kE;MAAuC,qCAAK,OAAL,EAAC,IAAd,C;MAAvC,Y;K;IACA,uD;MAAA,kE;MAAwC,qCAAK,SAAL,EAAGB,KAAhB,C;MAAxC,Y;K;IAG2C,+C;MAA0D,4BAAiB,OAAjB,EAA0B,KAA1B,C;;K;;IACrG,2C;MAAA,+D;MAAuB,kCAAK,IAAL,EAAW,IAAX,C;MAAvB,Y;K;IACA,sD;MAAA,+D;MAAuC,kCAAK,OAAL,EAAC,IAAd,C;MAAvC,Y;K;IACA,oD;MAAA,+D;MAAwC,kCAAK,SAAL,EAAGB,KAAhB,C;MAAxC,Y;K;IAG+C,4C;8BAAwD,O;;K;;IACvG,+C;MAAA,mE;MAAuB,sCAAK,IAAL,C;MAAvB,Y;K;IAGqD,yD;MAA0D,4BAAiB,OAAjB,EAA0B,KAA1B,C;;K;;IAC/G,qD;MAAA,yE;MAAuB,4CAAK,IAAL,EAAW,IAAX,C;MAAvB,Y;K;IACA,gE;MAAA,yE;MAAuC,4CAAK,OAAL,EAAC,IAAd,C;MAAvC,Y;K;IACA,8D;MAAA,yE;MAAwC,4CAAK,SAAL,EAAGB,KAAhB,C;MAAxC,Y;K;IAGmD,uD;MAA0D,4BAAiB,OAAjB,EAA0B,KAA1B,C;;K;;IAC7G,mD;MAAA,uE;MAAuB,0CAAK,IAAL,EAAW,IAAX,C;MAAvB,Y;K;IACA,8D;MAAA,uE;MAAuC,0CAAK,OAAL,EAAC,IAAd,C;MAAvC,Y;K;IACA,4D;MAAA,uE;MAAwC,0CAAK,SAAL,EAAGB,KAAhB,C;MAAxC,Y;K;IAI2C,wC;sCAAG,E,O;;K;;IAC3G,2C;MAAA,+D;MAAuB,kCAAK,IAAL,C;MAAvB,Y;K;IAI0C,uC;8BAAwD,O;;K;;IACIG,0C;MAAA,8D;MAAuB,iCAAK,IAAL,C;MAAvB,Y;K;IAGwC,qC;8BAAwD,O;;K;;IACg,wC;MAAA,4D;MAAuB,+BAAK,IAAL,C;MAAvB,Y;K;IAIJ,wC;MACmD,mBAAM,OAAN,EAae,KAaf,C;;K;;IAC/C,oC;MAAA,wD;MAAuB,sBAAK,IAAL,Q;MAAvB,Y;K;IACA,+C;MAAA,wD;MAAgC,2BAAK,OAAL,EAAC,IAAd,C;MAAhC,Y;K;IACA,+C;MAAA,wD;MAAiD,IAAY,I;MAAZB,2BAAa,SAAR,OAAQ,CAAb,EAAyB,sDAzB,C;MAApC,Y;K;IAG4C,yC;8BAAwD,O;;K;;IACpG,4C;MAAA,gE;MAAuB,mCAAK,IAAL,C;MAAvB,Y;K;IAIyC,sC;8BAAwD,O;;K;;IACjG,yC;MAAA,6D;MAAuB,gCAAK,IAAL,C;MAAvB,Y;K;IAGkD,sD;MAA0D,4BAAiB,OAAjB,EAA0B,KAA1B,C;;K;;IAC5G,kD;MAAA,sE;MAAuB,yCAAK,IAAL,EAAW,IAAX,C;MAAvB,Y;K;IACA,6D;MAAA,sE;MAAuC,yCAAK,OAAL,EAAC,IAAd,C;MAAvC,Y;K;IACA,2D;MAAA,sE;MAAwC,yCAAK,SAAL,EAAGB,KAAhB,C;MAAxC,Y;K;IAG0D,8D;MAA0D,4BAAiB,OAAjB,EAA0B,KAA1B,C;;K;;IACpH,0D;MAAA,8E;MAAuB,iDAAK,IAAL,EAAW,IAAX,C;MAAvB,Y;K;IACA,qE;MAAA,8E;MAAuC,iDAAK,OAAL,EAAC,IAAd,C;MAAvC,Y;K;IACA,mE;MAAA,8E;MAAwC,iDAAK,SAAL,EAAGB,KAAhB,C;MAAxC,Y;K;6FCIGJ,yB;MAEI,OAAG,GAAG,CAAC,QAAD,C;K;mFAGV,oB;MAEI,OAAJ,GAAL,GAAG,G;K;6ETVN,a;MAK8C,cAAvC,C;K;6ECHP,Y;MAG+C,S;K;IA6B/C,2B;MAG4D,0BAAe,WAAf,C;K;IAE5D,mC;MAIwF,0BAAe,WAAf,C;K;IAExF,mC;MAKwE,0BAAe,WAAf,C;K;IAGxE,4B;MAI8D,Q;MAH1D,aAAkB,GAAL,O;MACTb,aAAkB,GAAL,O;MACTb,YAAiB,C;MACjB,OAAO,QAAQ,MAAR,IAAkB,QAAQ,MAAjC,C;QAAyC,IAAI,KAAJ,IAAA,IAAI,YAAJ,EAAL,oBAAJ,O;;MACTD,OAAO,G;K;IAIX,wD;MAMuC,Q;MALnC,aAAA,MAAO,OAAM,CAAN,EAAS,OAAT,C;MA0BpB,IAzBc,MAyBL,OAAL,KAAkB,SAAtB,C;QAZBsB,MA0BIB,UA1BU,MA0BS,O;;MAzBvB,YAAiB,MAAO,O;MACxB,IAAL,UAAU,KAAAd,C;QACI,gBAAgB,O;QACHB,OAAO,QAAQ,OAaf,C;UAAwB,OAAO,YAAP,EAAO,oBAAP,UAAkB,Y;;;MAE9C,OAAO,M;K;IAGX,gD;MAKoB,UAAmB,M;MAJnC,aAAa,KAAM,Q;MACnB,MAAO,OAAP,IAAiB,UAAW,K;MAc5B,IAbc,KAAI,OAAL,KAAkB,SAAtB,C;QAbqB

,MAcJB,UAdU,KAcS,O;;MAbvB,YAAiB,KAAM,O;MACP,4B;MAAhB,OAAGB,cAAhB,C;QAAgB,yB;QAAy,O
AAO,cAAP,EAAO,sBAAP,YAAkB,O;;MAC9C,OAAO,M;K;IAGX,yD;MAEoB,UAAGB,M;MADhC,YAAy,U;M
ACI,4B;MAAhB,OAAGB,cAAhB,C;QAAgB,yB;QAAy,IAAI,cAAJ,EAAs,BAAJ,YAAe,O;;MAC3C,OAAO,G;K
;oFAGX,oB;MACI,IAAI,IAAK,OAAL,KAakB,SAAtB,C;QACI,YAAc,IAAK,O;;K;0EAI3B,wB;MAA+D,OAAA,
MAAA,QAAO,GAAP,EAAY,OAAZ,C;K;IS/F5E,mC;MAOI,kBAakB,MAAA,eAAc,SAAd,C;MAC/B,iBAAiB,M
AAa,eAAc,IAAd,C;MAC9B,OAAW,gBA Ae,UAA nB,GAA+B,SAA/B,GAAyC,CAAC,S;K;0ECUrD,2B;MAKyE,
OAAA,MAAA,gBA Ae,IAAf,C;K;4EAyBtF,2B;MAKsE,OAAA,MAAA,eAAc,IAAd,C;K;kEAGnF,qB;MACgD,OA
AA,MAAA,KAAK,UAA S,GAAT,EAAC,IAAd,C;K;wEAC hC,qB;MAAQ,OAAK,SAAY,a;K;0EACxB,qB;MAAQ,
OAAK,SAAY,c;K;IC3D5D,0D;MAGI,OAAO,I;K;ICHX,sC;MAMsD,OAAA,SAAY,UAA S,WAAW,KAA X,CAA
T,C;K;ItDKIE,uC;Mf2nBW,Q;MAAA,IernBgB,Kfq nBZ,IAAS,CAAT,IernBY,Kfq nBE,IAAS,wBAA3B,C;QAAA,
OAA sC,UernBtB,Kfq nBsB,C;;QernBb,MAAM,8BAA0B,iCAAuB,gBAAvB,MAA1B,C;;MAAtC,W;K;IAGJ,uC;
Mf4nBW,Q;MAAA,IetnBgB,KfsnBZ,IAAS,CAAT,IetnBY,KfsnBE,IAAS,0BAA3B,C;QAAA,OAA sC,UetnBtB,Kf
snBsB,C;;QetnBb,MAAM,8BAA0B,iCAAuB,gBAAvB,MAA1B,C;;MAAtC,W;K;IAGJ,uC;Mf6nBW,Q;MAAA,Ie
vnBgB,KfunBZ,IAAS,CAAT,IevnBY,KfunBE,IAAS,0BAA3B,C;QAAA,OAA sC,UevnBtB,KfunBsB,C;;QevnBb,
MAAM,8BAA0B,iCAAuB,gBAAvB,MAA1B,C;;MAAtC,W;K;IAGJ,uC;Mf8nBW,Q;MAAA,IexnBgB,KfwnBZ,I
AAS,CAAT,IexnBY,KfwnBE,IAAS,0BAA3B,C;QAAA,OAA sC,UexnBtB,KfwnBsB,C;;QexnBb,MAAM,8BAA0
B,iCAAuB,gBAAvB,MAA1B,C;;MAAtC,W;K;IAGJ,uC;Mf+nBW,Q;MAAA,IeznBgB,KfynBZ,IAAS,CAAT,Iezn
BY,KfynBE,IAAS,0BAA3B,C;QAAA,OAA sC,UeznBtB,KfynBsB,C;;QeznBb,MAAM,8BAA0B,iCAAuB,gBAAv
B,MAA1B,C;;MAAtC,W;K;IAGJ,uC;MfgoBW,Q;MAAA,Ie1nBgB,Kf0nBZ,IAAS,CAAT,Ie1nBY,Kf0nBE,IAAS,
0BAA3B,C;QAAA,OAA sC,Ue1nBtB,Kf0nBsB,C;;Qe1nBb,MAAM,8BAA0B,iCAAuB,gBAAvB,MAA1B,C;;MA
AtC,W;K;IAGJ,uC;MfioBW,Q;MAAA,Ie3nBgB,Kf2nBZ,IAAS,CAAT,Ie3nBY,Kf2nBE,IAAS,0BAA3B,C;QAAA
,OAA sC,Ue3nBtB,Kf2nBsB,C;;Qe3nBb,MAAM,8BAA0B,iCAAuB,gBAAvB,MAA1B,C;;MAAtC,W;K;IAGJ,uC;
MfkoBW,Q;MAAA,Ie5nBgB,Kf4nBZ,IAAS,CAAT,Ie5nBY,Kf4nBE,IAAS,0BAA3B,C;QAAA,OAA sC,Ue5nBtB,
Kf4nBsB,C;;Qe5nBb,MAAM,8BAA0B,iCAAuB,gBAAvB,MAA1B,C;;MAAtC,W;K;IAGJ,wC;MfmoBW,Q;MAA
A,Ie7nBgB,Kf6nBZ,IAAS,CAAT,Ie7nBY,Kf6nBE,IAAS,0BAA3B,C;QAAA,OAA sC,Ue7nBtB,Kf6nBsB,C;;Qe7n
Bb,MAAM,8BAA0B,iCAAuB,gBAAvB,MAA1B,C;;MAAtC,W;K;IAGJ,2B;MAII,OAAO,cAAa,SAAb,C;K;oFAG
X,yB;MAAA,gD;MAAA,4B;QAKI,OAA sC,OAA/B,SAA+B,C;O;KAL1C,C;oFAQA,yB;MAAA,gD;MAAA,4B;Q
AKI,OAAuC,OAAhC,SAAGC,C;O;KAL3C,C;oFAQA,yB;MAAA,gD;MAAA,4B;QAKI,OAAqC,OAA9B,SAA8B,
C;O;KALzC,C;oFAQA,yB;MAAA,gD;MAAA,4B;QAKI,OAA sC,OAA/B,SAA+B,C;O;KAL1C,C;oFAQA,yB;MA
AA,gD;MAAA,4B;QAKI,OAAuC,OAAhC,SAAGC,C;O;KAL3C,C;oFAQA,yB;MAAA,gD;MAAA,4B;QAKI,OA
AwC,OAAjC,SAAiC,C;O;KAL5C,C;oFAQA,yB;MAAA,gD;MAAA,4B;QAKI,OAAyC,OAAIC,SAAkC,C;O;KA
L7C,C;IAYW,2C;MAAA,8B;MAAS,uB;K;4FACW,Y;MAAQ,OAAA,gBAAY,O;K;6CAC3C,Y;MAAkC,OAAA,g
BfunP/B,YAAQ,C;K;oDetnPX,mB;MAAgD,OAAy,WAAZ,gBAAY,EAAS,OAAT,C;K;iDAC5D,iB;MACI,oCAA
a,2BAakB,KAAIB,EAAYB,SAAZB,C;MACb,OAAO,6BAAY,KAAZ,E;K;mDAEX,mB;MAES,Q;MAAL,IAAI,e
AAC,uFAAD,CAAJ,C;QAAgC,OAAO,E;MACvC,OAAMb,UAAZ,gBAAY,EAAQ,OAAR,C;K;uDAEvB,mB;MA
ES,Q;MAAL,IAAI,eAAC,uFAAD,CAAJ,C;QAAgC,OAAO,E;MACvC,OAAMb,cAAZ,gBAAY,EAAY,OAAZ,C;
K;;IApB/B,6B;MAII,0C;K;IAqBJ,+C;MAAI,OAAy,kBAAL,SAAK,EAakB,KAAIB,C;K;IAqBhB,0C;MASI,OAA
Y,oBAAL,SAAK,C;K;IAehB,0C;MAYI,OAAy,oBAAL,SAAK,C;K;IAkBhB,2C;MAWI,OAAy,cAAL,SAAK,EA
Ac,KAAAd,C;K;IAGhB,2C;MAWI,OAAy,cAAL,SAAK,EAAC,KAAAd,C;K;IAGhB,4C;MAWI,OAAy,cAAL,SA
AK,EAAC,KAAAd,C;K;IAGhB,4C;MAWI,OAAy,cAAL,SAAK,EAAC,KAAAd,C;K;IAGhB,4C;MAWI,OAAy,c
AAL,SAAK,EAAC,KAAAd,C;K;IAGhB,4C;MAWI,OAAy,cAAL,SAAK,EAAC,KAAAd,C;K;IAGhB,4C;MAWI,OA
AY,cAAL,SAAK,EAAC,KAAAd,C;K;IAwHhB,sC;MAOI,OAAy,gBAAL,SAAK,C;K;IAGhB,sC;MAOI,OAAy,gB
AAL,SAAK,C;K;IAGhB,uC;MAOI,OAAy,gBAAL,SAAK,C;K;IAGhB,uC;MAOI,OAAy,gBAAL,SAAK,C;K;IA
GhB,uC;MAOI,OAAy,gBAAL,SAAK,C;K;IAGhB,uC;MAOI,OAAy,gBAAL,SAAK,C;K;IAGhB,uC;MAOI,OA
AY,gBAAL,SAAK,C;K;IAGhB,uC;MAOI,OAAy,gBAAL,SAAK,C;K;IAGhB,uC;MAOI,OAAy,gBAAL,SAAK,
C;K;IAoFhB,sC;MASI,OAAy,gBAAL,SAAK,C;K;IAGhB,sC;MASI,OAAy,gBAAL,SAAK,C;K;IAGhB,uC;MAS
I,OAAy,gBAAL,SAAK,C;K;IAGhB,uC;MASI,OAAy,gBAAL,SAAK,C;K;IAGhB,uC;MASI,OAAy,gBAAL,SA

AK,C;K;IAGhB,uC;MASI,OAAY,gBAAL,SAAK,C;K;IAGhB,uC;MASI,OAAY,gBAAL,SAAK,C;K;IAGhB,uC;MASI,OAAY,gBAAL,SAAK,C;K;IAGhB,uC;MASI,OAAY,gBAAL,SAAK,C;K;wFAsGhB,yB;MAAA,8C;MAAA,kF;QAmB0E,iC;UAAA,oBAAYB,C;QAAG,0B;UAAA,aAAkB,C;QAAG,wB;UAAA,WAAgB,gB;QACvI,UAAU,SAAV,EAAgB,WAAhB,EAA6B,iBAA7B,EAAgD,UAAhD,EAA4D,QAA5D,C;QACA,OAAO,W;O;KArBX,C;wFAwBA,yB;MAAA,8C;MAAA,kF;QAmBoE,iC;UAAA,oBAAYB,C;QAAG,0B;UAAA,aAAkB,C;QAAG,wB;UAAA,WAAgB,gB;QACjI,UAAU,SAAV,EAA0C,WAA1C,EAAiF,iBAAjF,EAAoG,UAApG,EAAgH,QAAhH,C;QACA,OAAO,W;O;KArBX,C;wFAwBA,yB;MAAA,8C;MAAA,kF;QAmBsE,iC;UAAA,oBAAYB,C;QAAG,0B;UAAA,aAAkB,C;QAAG,wB;UAAA,WAAgB,gB;QACnI,UAAU,SAAV,EAA2C,WAA3C,EAAmF,iBAAnF,EAAsg,UAAtG,EAAkH,QAAIH,C;QACA,OAAO,W;O;KArBX,C;wFAwBA,yB;MAAA,8C;MAAA,kF;QAmBkE,iC;UAAA,oBAAYB,C;QAAG,0B;UAAA,aAAkB,C;QAAG,wB;UAAA,WAAgB,gB;QAC/H,UAAU,SAAV,EAAyC,WAAzC,EAA+E,iBAA/E,EAAkG,UAAIG,EAA8G,QAA9G,C;QACA,OAAO,W;O;KArBX,C;wFAwBA,yB;MAAA,8C;MAAA,kF;QAmBoE,iC;UAAA,oBAAYB,C;QAAG,0B;UAAA,aAAkB,C;QAAG,wB;UAAA,WAAgB,gB;QACjI,UAAU,SAAV,EAA0C,WAA1C,EAAiF,iBAAjF,EAAoG,UAApG,EAAgH,QAAhH,C;QACA,OAAO,W;O;KArBX,C;wFAwBA,yB;MAAA,8C;MAAA,kF;QAmBsE,iC;UAAA,oBAAYB,C;QAAG,0B;UAAA,aAAkB,C;QAAG,wB;UAAA,WAAgB,gB;QACnI,UAAU,SAAV,EAA2C,WAA3C,EAAmF,iBAAnF,EAAsg,UAAtG,EAAkH,QAAIH,C;QACA,OAAO,W;O;KArBX,C;uFAwBA,yB;MAAA,8C;MAAA,kF;QAmBwE,iC;UAAA,oBAAYB,C;QAAG,0B;UAAA,aAAkB,C;QAAG,wB;UAAA,WAAgB,gB;QACrI,UAAU,SAAV,EAA4C,WAA5C,EAAqF,iBAArF,EAAwG,UAAxG,EAAoH,QAApH,C;QACA,OAAO,W;O;KArBX,C;yFAwBA,yB;MAAA,8C;MAAA,kF;QAmB0E,iC;UAAA,oBAAYB,C;QAAG,0B;UAAA,aAAkB,C;QAAG,wB;UAAA,WAAgB,gB;QACvI,UAAU,SAAV,EAA6C,WAA7C,EAAuF,iBAAvF,EAA0G,UAA1G,EAAsh,QAAtH,C;QACA,OAAO,W;O;KArBX,C;yFAwBA,yB;MAAA,8C;MAAA,kF;QAmBoE,iC;UAAA,oBAAYB,C;QAAG,0B;UAAA,aAAkB,C;QAAG,wB;UAAA,WAAgB,gB;QACjI,UAAU,SAAV,EAA0C,WAA1C,EAAiF,iBAAjF,EAAoG,UAApG,EAAgH,QAAhH,C;QACA,OAAO,W;O;KArBX,C;oFAwBA,qB;MAOI,OAAY,SAAY,Q;K;oFAG5B,qB;MAOI,OAAY,SAAY,Q;K;oFAG5B,qB;MAOI,OAAY,SAAY,Q;K;qFAG5B,qB;MAOI,OAAY,SAAY,Q;K;IAG5B,8B;MAMW,WAAS,W;MAAT,YAA2B,SAAY,Q;MwCl7B9C,eAAiB,I;MxCk7BjB,OwCj7BO,K;K;qFxCo7BX,qB;MAOI,OAAY,SAAY,Q;K;qFAG5B,qB;MAOI,OAAY,SAAY,Q;K;IAG5B,8B;MAMW,WAAS,c;MAAT,YAA8B,SAAY,Q;MwC/8BjD,eAAiB,I;MxC+8BjB,OwC98BO,K;K;IxCI9BX,8B;MAMW,WAAS,W;MAAT,YAA2B,SAAY,Q;MwCx9B9C,eAAiB,I;MxCw9BjB,OwCv9BO,K;K;IxCO9BX,uC;MD5oCI,IAAI,ECspCI,WAAW,CDtpCf,CAAJ,C;QACI,cCqpCoB,0C;QDppCpB,MAAM,gCAAYB,OAAQ,WAAjC,C;;MCqpCV,OAAO,SAAS,SAAT,EAAe,cAAU,OAAV,CAAF,C;K;IAGX,uC;MD1pCI,IAAI,ECoqCI,WAAW,CDpqCf,CAAJ,C;QACI,cCmqCoB,0C;QDlqCpB,MAAM,gCAAYB,OAAQ,WAAjC,C;;MCmqCV,OAAO,SAAS,SAAT,EAAe,eAAW,OAAX,CAAF,C;K;IAGX,uC;MDxqCI,IAAI,ECkrCI,WAAW,CDlrCf,CAAJ,C;QACI,cCirCoB,0C;QDhrCpB,MAAM,gCAAYB,OAAQ,WAAjC,C;;MCirCV,OAAO,SAAS,SAAT,EAAe,eAAS,OAAT,CAAF,C;K;IAGX,uC;MDtrCI,IAAI,ECgsCI,WAAW,CDhsCf,CAAJ,C;QACI,cC+rCoB,0C;QD9rCpB,MAAM,gCAAYB,OAAQ,WAAjC,C;;MC+rCH,WAAS,W;MAAT,YAAsB,gBAAgB,SAAhB,EAAhB,OAAtB,K;MwChhC7B,eAAiB,I;MxCghCjB,OwC/gCO,K;K;IxChkCX,uC;MDpsCI,IAAI,EC8sCI,WAAW,CD9sCf,CAAJ,C;QACI,cC6sCoB,0C;QD5sCpB,MAAM,gCAAYB,OAAQ,WAAjC,C;;MC6sCV,OAAO,SAAS,SAAT,EAAe,iBAAW,OAAX,CAAF,C;K;IAGX,uC;MDltCI,IAAI,EC4tCI,WAAW,CD5tCf,CAAJ,C;QACI,cC2tCoB,0C;QD1tCpB,MAAM,gCAAYB,OAAQ,WAAjC,C;;MC2tCV,OAAO,SAAS,SAAT,EAAe,iBAAY,OAAs,CAAF,C;K;IAGX,uC;MDhuCI,IAAI,EC0uCI,WAAW,CD1uCf,CAAJ,C;QACI,cCyuCoB,0C;QDxuCpB,MAAM,gCAAYB,OAAQ,WAAjC,C;;MCyuCH,WAAS,c;MAAT,YAAyB,gBAAgB,SAAhB,EAAhB,OAAtB,EAA+B,KAA/B,C;MwCl1jChC,eAAiB,I;MxC0jCjB,OwCzjCO,K;K;IxC4jCX,uC;MD9uCI,IAAI,ECwvCI,WAAW,CDxvCf,CAAJ,C;QACI,cCuvCoB,0C;QDtvCpB,MAAM,gCAAYB,OAAQ,WAAjC,C;;MCuvCH,WAAS,W;MAAT,YAAsB,SAAS,SAAT,EAAe,iBAAU,OAAs,CAAF,C;MwCxc7B,eAAiB,I;MxCwkCjB,OwCvkCO,K;K;IxCOkCX,uC;MD5vCI,IAAI,ECuwCI,WAAW,CDvWCf,CAAJ,C;QACI,cCswCoB,0C;QDwCpB,MAAM,gCAAYB,OAAQ,WAAjC,C;;MCswCV,OAAO,gBAAgB,SAAhB,EAAhB,OAAtB,EAA+B,IAA/B,C;K;IAGX,sD;MAWI,oCAAA,2BAAkB,SAAIb,EAA6B,OAA7B,EAAc,gBAAtC,C;MACb,OAAY,SAAY,OAAM,SAAN,EAAiB,OAajB,C;K;IAG5B,sD;MAUI,oCAAA,2BAAkB,SAAIb,EAA6B,OAA7B,EAAc,gBAAtC,C;MACb,OAAY,SAAY,OAAM,SAAN,EAAiB,O

SAArB,EAA2B,QAA3B,C;O;KALX,C;iFAQA,yB;MAAA,iD;MAAA,sC;QAKI,OAAO,qBAAqB,SAArB,EAA2B,QAA3B,C;O;KALX,C;iFAQA,yB;MAAA,iD;MAAA,sC;QAKI,OAAO,qBAAqB,SAArB,EAA2B,QAA3B,C;O;KALX,C;8FAQA,8B;MAKI,OAAy,SAAY,QAAO,CAAQ,OAAR,CAAP,C;K;IAoBL,2B;MAAsB,OAAA,CAAE,iBAAU,CAAV,C;K;IAP/C,2B;MAOI,IAAI,mBAAO,CAAX,C;QAwQY,eAxQO,WAwQP,C;;K;IAAnhB,2B;MAQI,IAAI,mBAAO,CAAX,C;QAAc,UAAU,SAAV,C;K;IAGIB,wC;MAQI,IAAI,mBAAO,CAAX,C;QAAc,cAAc,SAAd,EAAoB,UAApB,C;K;IAGIB,gD;MAewD,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MACvF,oCAAA,2BAAkB,SAAlB,EAA6B,OAA7B,EAAc,gBAAtC,C;MACb,gBAAc,SAAd,EAAoB,SAApB,EAA+B,OAA/B,EAAwC,cAAxC,C;K;IAGJ,gD;MAaiC,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MACHe,oCAAA,2BAAkB,SAAlB,EAA6B,OAA7B,EAAc,gBAAtC,C;MACb,eAAoB,SAAY,UAAS,SAAT,EAAoB,OAApB,C;MACvB,KAAT,QAAS,C;K;IAGb,gD;MAakC,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MACjE,oCAAA,2BAAkB,SAAlB,EAA6B,OAA7B,EAAc,gBAAtC,C;MACb,eAAoB,SAAY,UAAS,SAAT,EAAoB,OAApB,C;MACvB,KAAT,QAAS,C;K;IAGb,gD;MAagC,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MAC/D,oCAAA,2BAAkB,SAAlB,EAA6B,OAA7B,EAAc,gBAAtC,C;MACb,eAAoB,SAAY,UAAS,SAAT,EAAoB,OAApB,C;MACvB,KAAT,QAAS,C;K;IAGb,gD;MAaiC,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MACHe,oCAAA,2BAAkB,SAAlB,EAA6B,OAA7B,EAAc,gBAAtC,C;MACb,gBAAc,SAAd,EAA8C,SAA9C,EAAyD,OAAzD,EAAkE,cAAIE,C;K;IAGJ,gD;MAakC,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MACjE,oCAAA,2BAAkB,SAAlB,EAA6B,OAA7B,EAAc,gBAAtC,C;MACb,eAAoB,SAAY,UAAS,SAAT,EAAoB,OAApB,C;MACvB,KAAT,QAAS,C;K;IAGb,gD;MAamC,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MACIE,oCAAA,2BAAkB,SAAlB,EAA6B,OAA7B,EAAc,gBAAtC,C;MACb,eAAoB,SAAY,UAAS,SAAT,EAAoB,OAApB,C;MACvB,KAAT,QAAS,C;K;IAGb,gD;MAaiC,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MACHe,oCAAA,2BAAkB,SAAlB,EAA6B,OAA7B,EAAc,gBAAtC,C;MACb,eAAoB,SAAY,UAAS,SAAT,EAAoB,OAApB,C;MACvB,KAAT,QAAS,C;K;iFAGb,iC;MAOI,SAAY,MAAK,UAAL,C;K;iFAGhB,iC;MAOI,SAAY,MAAK,UAAL,C;K;iFAGhB,iC;MAOI,SAAY,MAAK,UAAL,C;K;iFAGhB,iC;MAOI,SAAY,MAAK,UAAL,C;K;iFAGhB,iC;MAOI,SAAY,MAAK,UAAL,C;K;IAGhB,yC;MAMI,IAAI,mBAAO,CAAX,C;QAAc,gBAAc,SAAd,EAAoB,UAApB,C;K;IAGIB,+D;MAa0E,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MACzG,oCAAA,2BAAkB,SAAlB,EAA6B,OAA7B,EAAc,gBAAtC,C;MACb,gBAAc,SAAd,EAAoB,SAApB,EAA+B,OAA/B,EAAwC,UAAxC,C;K;IAGJ,mC;MAII,OAAO,EAAS,MAAM,MAAK,SAAL,C;K;IAGIB,mC;MAII,OAAO,EAAS,MAAM,MAAK,SAAL,C;K;IAGIB,mC;MAII,OAAO,EAAS,MAAM,MAAK,SAAL,C;K;IAGIB,mC;MAII,OAAO,EAAS,MAAM,MAAK,SAAL,C;K;IAGIB,mC;MAII,OAAO,EAAS,MAAM,MAAK,SAAL,C;K;IAOH,kD;MAAA,wB;QAAW,qCAAK,KAAL,E;O;K;IAJIC,oC;MAII,OAAO,iBAAM,gBAAN,EAAY,gCAAZ,C;K;IuDnpEX,oB;MAAA,wB;MAEI,6B;MACA,gC;MAKuB,UAAT,MAAS,EAAT,MAAS,EAAT,M;MAFV,eAAe,kE;MACf,iBAAiB,eAAS,GAAT,C;MACE,sBAAT,QAAS,C;MAAT,mB;MAAA,kB;MAAA,kB;MAAV,8C;QACI,WAAW,oBAAS,CAAT,CzC2BuB,IyC3BIC,IAA+B,C;;MAInC,qBAAqB,48C;MACrB,WAAW,mBAAmB,cAAnB,EAAmC,UAAnC,EAA+C,IAA/C,C;MACX,YAAY,eAAS,IAAK,OAAL,GAAY,CAAZ,IAAT,C;MACZ,0BAAU,IAAV,e;QACI,MAAM,MAAI,CAAJ,IAAN,IAAe,MAAM,GAAN,IAAW,KAAC,GAAL,CAAX,I;;MAEnB,yBAAoB,K;MAGpB,oBAAoB,m/D;MACpB,4BAAuB,mBAAmB,aAAnB,EAAkC,UAAIC,EAA8C,IAA9C,C;K;;IAvB/B,gC;MAAA,+B;QAAA,c;;MAAA,wB;K;IA2BA,qC;MAKkB,IAJP,I;MACH,WAAO,EAAP,C;QA Ae,W;WACf,WAAO,IAAP,C;QAAGB,OAAI,CAAC,KAAO,CAAR,MAAc,CAAIB,GAAqB,QAAS,CAA9B,GAAqC,OAAS,E;;QAEID,QAAM,KAAC,CAAX,C;eACI,C;YAAK,eAAS,E;YAAAd,K;eACA,C;YAAK,OAAC,QAAS,CAAV,GAAiB,E;YAAAtB,K;;YACQ,cAAS,E;YAHrB,K;;MAJR,W;K;IAYJ,qC;MAIL,SAAS,SzCRiC,I;MyCU1C,YAAy,kBAAkB,sBAAS,kBAA3B,EAA8C,EAA9C,C;MACZ,YAAY,sBAAS,kBAAT,CA A2B,KAA3B,C;MACZ,WAAW,sBAAS,qBAAT,CAA8B,KAA9B,C;MACX,YAAY,kBAAkB,IAAIB,EAAwB,KA AK,KAAL,IAAxB,C;MAEZ,OAAW,UAAS,EAAb,GAAYC,mDAAzC,GAAoD,K;K;IAG/D,8D;MAKiB,UAIE,M;MARf,aAAa,eAAS,YAAT,C;MACb,YAAY,C;MACZ,UAAU,C;MACV,YAAY,C;MACC,yB;MAAb,OAAa,cAAb ,C;QAAa,iC;QACT,aAAa,WAAW,IzCxBc,IyCwBzB,C;QACb,MAAM,MAAQ,CAAC,SAAW,EAAZ,KAAkB,K; QACpC,IAAI,SAAS,EAAb,C;UACI,OAAO,cAAP,EAAO,sBAAP,YAAkB,G;UACIB,MAAM,C;UACN,QAAQ,C; ;UAER,gBAAS,CAAT,I;;MAGR,OAAO,M;K;ICIE,X,+B;MAII,eAAe,CAAC,iBAAO,CAAP,IAAD,IAAa,CAAb,I

;MACf,IAAI,WAAW,CAAf,C;QAAkB,M;MACIB,mBAAmB,2B;MACnB,iBAAc,CAAd,WAAiB,QAAjB,U;QAC I,UAAU,sBAAK,KAAL,C;QACV,sBAAK,KAAL,EAAc,sBAAK,YAAL,CAAd,C;QACA,sBAAK,YAAL,EAAqB ,GAARb,C;QACA,mC;;K;IrDbR,wB;MAOI,OAAW,oBAAK,CAAL,MAAJ,GAAY,CAAZ,GAAMb,C;K;mFAG9 B,yB;MAKBA,iB;MAIBA,uB;QAMI,OakBO,MAAO,KAlBC,CakBD,EAlBY,CakBZ,C;O;KAXBIB,C;mFASA,y B;MASA,iB;MATA,uB;QAMI,OASO,MAAO,KATC,CASD,EATY,CASZ,C;O;KAlfB,C;mFASA,yB;MAAA,iB; MAAA,uB;QAMI,OAAO,MAAO,KAAI,CAAJ,EAAO,CAAP,C;O;KANIB,C;mFASA,gB;MAMI,OAAW,kBAAK ,CAAL,MAAJ,GAAY,CAAZ,GAAMb,C;K;mFAG9B,yB;MAAA,iB;MAAA,uB;QAQI,OAAO,MAAO,KAAI,CA AJ,EAAO,CAAP,C;O;KARIB,C;mFAWA,yB;MAAA,iB;MAAA,uB;QAQI,OAAO,MAAO,KAAI,CAAJ,EAAO,C AAP,C;O;KARIB,C;IAWA,2B;MAOI,OAAO,SAAM,CAAN,EAAS,SAAM,CAAN,EAAS,CAAT,CAAT,C;K;mF AGX,yB;MAAA,iB;MAAA,0B;QAMI,OAAO,MAAO,KAAI,CAAJ,EAAO,CAAP,C;O;KARIB,C;mFAWA,yB;M AAA,iB;MAAA,0B;QAMI,OAAO,MAAO,KAAI,CAAJ,EAAO,CAAP,C;O;KANIB,C;mFASA,yB;MAAA,iB;M AAA,0B;QAMI,OAAO,MAAO,KAAI,CAAJ,EAAO,CAAP,C;O;KANIB,C;mFASA,mB;MAMW,UAAe,CAPeX,iBAoEc,CAPeD,MAAJ,GAoEe,CAPeF,GAoEkB,C;MAAzB, OAAa,CAPeF,iBAAK,GAAL,MAAJ,GAoEM,CAPeN,GAAMb,G;K;mFAuE9B,yB;MAAA,iB;MAAA,0B;QAQI, OAAO,MAAO,KAAI,CAAJ,EAAO,CAAP,C;O;KARIB,C;mFAWA,yB;MAAA,iB;MAAA,0B;QA QI,OAAO,MAAO,KAAI,CAAJ,EAAO,CAAP,C;O;KARIB,C;IAWA,4B;MAQc,Q;MADV,UAAU, C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,MAAM,SAAM,GAAN,EAAW,CAAX,C;;MACv B,OAAO,G;K;IAGX,4B;MAMc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAi B,MAxHV,MAAO,KAwHe,GAXHf,EAwHoB,CAXHpB,C;;MAyHd,OAAO,G;K;IAGX,4B;MAMc,Q;MADV,UA AU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,MAIIV,MAAO,KAKIe,GAlIf,EAKIoB,CAlIp B,C;;MAMId,OAAO,G;K;IAGX,4B;MAMc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAA V,M;QAAiB,MA5IV,MAAO,KA4Ie,GA5If,EA4IoB,CA5IpB,C;;MA6Id,OAAO,G;K;IAGX,4B;MAMc,Q;MADV, UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAuB,UAAM,G;QAAZ,MA7IN,oBA6IuB,CA7 IvB,MAAJ,GAAY,GAAZ,GA6I2B,C;;MACIC,OAAO,G;K;IAGX,4B;MAQc,Q;MADV,UAAU,C;MACV,wBAA U,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,MA9IV,MAAO,KA8Ie,GA9If,EA8IoB,CA9IpB,C;;MA+Id,OAAO ,G;K;IAGX,4B;MAQc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,MA/IV, MAAO,KA+Ie,GA/If,EA+IoB,CA/IpB,C;;MAGJd,OAAO,G;K;IAGX,wB;MAOI,OAAW,oBAAK,CAAL,MAAJ,G AAY,CAAZ,GAAMb,C;K;mFAG9B,yB;MAKBA,iB;MAIBA,uB;QAMI,OakBO,MAAO,KAlBC,CakBD,EAlBY, CakBZ,C;O;KAXBIB,C;mFASA,yB;MASA,iB;MATA,uB;QAMI,OASO,MAAO,KATC,CASD,EATY,CASZ,C;O ;KAlfB,C;mFASA,yB;MAAA,iB;MAAA,uB;QAMI,OAAO,MAAO,KAAI,CAAJ,EAAO,CAAP,C;O;KANIB,C;m FASA,gB;MAMI,OAAW,kBAAK,CAAL,MAAJ,GAAY,CAAZ,GAAMb,C;K;mFAG9B,yB;MAAA,iB;MAAA,uB ;QAQI,OAAO,MAAO,KAAI,CAAJ,EAAO,CAAP,C;O;KARIB,C;mFAWA,yB;MAAA,iB;MAAA,uB;QAQI,OAA O,MAAO,KAAI,CAAJ,EAAO,CAAP,C;O;KARIB,C;IAWA,2B;MAOI,OAAO,SAAM,CAAN,EAAS,SAAM,CAA N,EAAS,CAAT,CAAT,C;K;mFAGX,yB;MAAA,iB;MAAA,0B;QAMI,OAAO,MAAO,KAAI,CAAJ,EAAO,CAAP,C;O;KANIB,C;mFASA,yB;MAAA,iB;MAAA,0B;QAMI,OAAO,MAAO,KAAI,CA AJ,EAAO,CAAP,C;O;KANIB,C;mFASA,mB;MAMW,UAAe,CAPeX,iBAoEc,CAPeD,MAAJ,GAo Ee,CAPeF,GAoEkB,C;MAAzB,OAAa,CAPeF,iBAAK,GAAL,MAAJ,GAoEM,CAPeN,GAAMb,G;K;mFAuE9B,y B;MAAA,iB;MAAA,0B;QAQI,OAAO,MAAO,KAAI,CAAJ,EAAO,CAAP,C;O;KARIB,C;mFAWA ,yB;MAAA,iB;MAAA,0B;QAQI,OAAO,MAAO,KAAI,CAAJ,EAAO,CAAP,C;O;KARIB,C;IAW A,4B;MAQc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,MAAM,SAAM,G AAN,EAAW,CAAX,C;;MACvB,OAAO,G;K;IAGX,4B;MAMc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB; QAAU,QAAA,KAAV,M;QAAiB,MAxHV,MAAO,KAwHe,GAXHf,EAwHoB,CAXHpB,C;;MAyHd,OAAO,G;K;I AGX,4B;MAMc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,MAIIV,MAA O,KAKIe,GAlIf,EAKIoB,CAlIpB,C;;MAMId,OAAO,G;K;IAGX,4B;MAMc,Q;MADV,UAAU,C;MACV,wBAAU, KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,MA5IV,MAAO,KA4Ie,GA5If,EA4IoB,CA5IpB,C;;MA6Id,OAAO,G ;K;IAGX,4B;MAMc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAuB,UAAM, G;QAAZ,MA7IN,oBA6IuB,CA7IvB,MAAJ,GAAY,GAAZ,GA6I2B,C;;MACIC,OAAO,G;K;IAGX,4B;MAQc,Q;

MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,MA9IV,MAAO,KA8Ie,GA9If,E
A8IoB,CA9IpB,C;;MA+Id,OAAO,G;K;IAGX,4B;MAQc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,
QAAA,KAAV,M;QAAiB,MA/IV,MAAO,KA+Ie,GA/If,EA+IoB,CA/IpB,C;;MAGJd,OAAO,G;K;IsDvaX,iB;MAA
A,qB;MAEI,0BAA0B,gBACtB,EADsB,EACd,IADc,EACN,IADM,EACE,IADF,EACU,IADV,EACkB,IADIB,EA
C0B,IAD1B,EACkC,IADIC,EAC0C,IAD1C,EACkD,IADID,EAC0D,IAD1D,EACkE,IADIE,EAC0E,IAD1E,EACk
F,IADIF,EAC0F,IAD1F,EACkG,IADIG,EAC0G,IAD1G,EACkH,IADIH,EAC0H,IAD1H,EACKI,IADII,EAETB,IA
FsB,EAEd,IAFc,EAEN,IAFM,EAEE,IAFF,EAEU,IAFV,EAEB,IAFIB,EAEOB,IAF1B,EAekC,IAFIC,EAEOC,IA
F1C,EAekD,KAFID,EAEOB,KAF1D,EAekE,KAFIE,EAEOE,KAF1E,EAekF,KAFIF,EAEOF,KAF1F,EAekG,KA
FIG,EAEOG,KAF1G,E;K;;IAF9B,6B;MAAA,4B;QAAA,W;;MAAA,qB;K;IAQA,0C;MAKI,aAAa,C;MACb,UAA
U,KAAM,OAAN,GAAa,CAAb,I;MACV,aAAa,E;MACb,YAAY,C;MACZ,OAAO,UAAU,GAAjB,C;QACI,SAAS
,CAAC,SAAS,GAAT,IAAD,IAAiB,CAAjB,I;QACT,QAAQ,MAAM,MAAN,C;QACR,IAAI,SAAS,KAAb,C;UAC
I,SAAS,SAAS,CAAT,I;aACR,IAAI,WAAU,KAAc,C;UACD,OAAO,M;;UAEP,MAAM,SAAS,CAAT,I;;MAEd,O
AAO,UAAc,SAAS,KAAb,GAAoB,CAApB,GAA2B,CAArC,K;K;IAGX,mC;MAKI,SAAS,S3CCiC,I;M2CA1C,Y
AAY,kBAaB,mBAAM,mBAAXB,EAAoC,EAAPC,C;MACZ,WAAW,KAAK,mBAAM,mBAAN,CAAiB,KAAj
B,CAAL,I;MACX,OAAW,OAAO,EAAX,GAAe,IAAf,GAAYB,E;K;IAGpC,gC;MAII,OAAO,6BAAoB,C;K;IC7C/
B,kB;MAAA,sB;MAEI,6B;MACA,8B;MACA,gC;MAKuB,UAAT,MAAS,EAAT,MAAS,EAAT,M;MAFV,eAAe,
kE;MACf,iBAAiB,eAAS,GAAT,C;MACE,sBAAT,QAAS,C;MAAT,mB;MAAA,kB;MAAA,kB;MAAV,8C;QACI,
WAAW,oBAAS,CAAT,C5C0BuB,I4C1BIC,IAA+B,C;;MAInC,qBAAqB,sW;MACrB,WAAW,mBAAmB,cAAnB,
EAAMC,UAAnc,EAA+C,GAA/C,C;MACX,YAAY,eAAS,IAAK,OAAc,C;MACZ,0BAAU,IAAV,e;QACI,IAAI,
QAAK,CAAT,C;UAAY,MAAM,GAAN,IAAW,KAAK,GAAL,C;;UACIB,MAAM,GAAN,IAAW,MAAM,MAAI,
CAAJ,IAAN,IAAe,KAAK,GAAL,CAAf,I;;MAEpB,yBAAoB,K;MAGpB,kBAaB,0U;MACIB,0BAAqB,mBAA
mB,WAAAnB,EAAGC,UAAhC,EAA4C,GAA5C,C;MAGrB,oBAAoB,i8B;MACpB,4BAAuB,mBAAmB,aAAnB,E
AAkC,UAAIC,EAA8C,GAA9C,C;K;;IA7B/B,8B;MAAA,6B;QAAA,Y;;MAAA,sB;K;IAiCA,iC;MAII,OAAO,6B
AAmB,C;K;IAG9B,oC;MAIW,wCAAmB,C;MAAnB,U;QAA6B,wB5CRM,a4CQN,C;;MAApC,W;K;IAGJ,oC;M
AIW,wCAAmB,C;MAAnB,U;QAA6B,wB5CfM,a4CeN,C;;MAApC,W;K;IAGJ,kC;MAQI,SAAS,S5C1BiC,I;M4C
2B1C,YAAY,kBAaB,oBAAO,kBAAZB,EAA4C,EAA5C,C;MAEZ,iBAAiB,oBAAO,kBAAP,CAAYB,KAAZB,C
;MACjB,eAAe,aAAa,oBAAO,mBAAP,CAA0B,KAA1B,CAAb,GAAGD,CAAhD,I;MACf,WAAW,oBAAO,qBAA
P,CAA4B,KAA5B,C;MAEX,IAAI,KAAK,QAAT,C;QACI,OAAO,C;;MAGX,kBAaB,OAAS,C;MAE3B,IAAI,g
BAAe,CAAnB,C;QACI,YAAY,C;QACZ,gBAAgB,U;QACHB,aAAU,CAAV,OAAa,CAAb,M;UACI,yBAAc,QAA
S,KAAV,GAAqB,GAAIC,K;UACA,IAAI,YAAY,EAAhB,C;YACI,OAAO,C;;UAEX,gBAAS,CAAT,I;UACA,yB
AAc,QAAS,KAAV,GAAqB,GAAIC,K;UACA,IAAI,YAAY,EAAhB,C;YACI,OAAO,C;;UAEX,gBAAS,CAAT,I;;
QAEJ,OAAO,C;;MAGX,IAAI,QAAQ,CAAZ,C;QACI,OAAO,W;;MAGX,eAAgB,KAAK,UAAL,I;MACHB,cAAg
B,QAAQ,EAAG,GAakB,WAAW,CAA7B,GAAoC,Q;MACHD,OAAQ,SAAU,IAAI,OAAl,IAAV,CAAD,GAA2B,
C;K;ICnGtC,0B;MAAA,8B;MACI,+BAA+B,gBAC3B,GAD2B,EACnB,GADmB,EACX,GADW,EACH,GADG,E
ACK,GADL,EACa,GADb,EACqB,GADrB,EAC6B,IAD7B,EACqC,IADrC,EAC6C,IAD7C,EACqD,IADrD,EAC6
D,IAD7D,EACqE,IADrE,EAC6E,IAD7E,EACqF,IADrF,EAC6F,KAD7F,EACqG,KADrG,EAC6G,KAD7G,EACq
H,KADrH,EAC6H,KAD7H,E;MAG/B,gCAAAGC,gBAC5B,CAD4B,EACzB,CADyB,EACtB,CADsB,EACnB,CAD
mB,EACHB,CADgB,EACb,CADa,EACV,CADU,EACP,EADO,EACH,CADG,EACA,EADA,EACI,CADJ,EACO,
CADP,EACU,EADV,EACc,EADd,EACkB,EADIB,EACsB,CADtB,EACyB,CADzB,EAC4B,CAD5B,EAC+B,CA
D/B,EACKC,CADIC,E;K;;IAJpC,sC;MAAA,qC;QAAA,oB;;MAAA,8B;K;IASA,qC;MACI,YAAY,kBAaB,4BA
Ae,wBAAjC,EAakD,SAaID,C;MACZ,OAAO,SAAS,CAAT,IAAc,aAAO,4BAAe,wBAAf,CAA+B,KAA/B,IAAw
C,4BAAe,yBAAf,CAAGC,KAAhC,CAAxC,IAAP,C;K;ICXzB,qC;MACI,OAAe,IAAR,8BAAgB,IAAhB,KACY,I
AAR,8BAAgB,IADpB,C;K;ICCX,wC;M5CiBW,Q;MAAA,I4CXgB,K5CWZ,IAAS,CAAT,I4CXY,K5CWE,IAAS,
2BAA3B,C;QAAA,OAAsC,qB4CXtB,K5CWsB,C;;Q4CXb,MAAM,8BAA0B,mCAAYB,gBAAzB,MAA1B,C;;M
AAAtC,W;K;ICRJ,sC;MAEI,WAAW,ShDkC+B,I;MgDhC1C,IAAY,GAAR,oBAAgB,GAAhB,KAAkC,GAAR,oBA
AgB,GAA1C,CAAJ,C;QACI,OAA8B,OAAtB,KAAK,CAAC,OAAO,CAAP,IAAD,IAAa,CAAb,IAAL,KAAsB,C;;
MAGIC,IAAY,IAAR,oBAAgB,IAAhB,KAAkC,IAAR,oBAAgB,IAA1C,CAAJ,C;QACI,OAAO,S;;MAEX,OAAO,
wB;K;ICPX,wC;MxCqTe,WwC7SY,KxC6SZ,IAAS,C;MAAT,S;QAAc,OwC7SF,KxC6SE,IAqgHT,gBAAR,iBAA

Q,C;;MArgHT,U;MAAA,S;QAAA,SAAsC,sBwC7StB,KxC6SsB,C;;QwC7Sb,MAAM,8BAA0B,iCAAUb,cAAvB,MAA1B,C;;MAAtC,a;K;IAGJ,wC;MxCsTe,WwC9SY,KxC8SZ,IAAS,C;MAAT,S;QAAC,OwC9SF,KxC8SE,IAigHT,gBAAR,iBAAQ,C;;MAJgHT,U;MAAA,S;QAAA,SAAsC,sBwC9StB,KxC8SsB,C;;QwC9Sb,MAAM,8BAA0B,iCAAUb,cAAvB,MAA1B,C;;MAAtC,a;K;IAGJ,wC;MxCuTe,WwC/SY,KxC+SZ,IAAS,C;MAAT,S;QAAC,OwC/SF,KxC+SE,IA6/GT,gBAAR,iBAAQ,C;;MA7/GT,U;MAAA,S;QAAA,SAAsC,sBwC/StB,KxC+SsB,C;;QwC/Sb,MAAM,8BAA0B,iCAAUb,cAAvB,MAA1B,C;;MAAtC,a;K;IAGJ,wC;MxCwTe,WwChTY,KxCgTZ,IAAS,C;MAAT,S;QAAC,OwChTF,KxCgTE,IAy/GT,gBAAR,iBAAQ,C;;MAz/GT,U;MAAA,S;QAAA,SAAsC,sBwChTitB,KxCgTsB,C;;QwChTb,MAAM,8BAA0B,iCAAUb,cAAvB,MAA1B,C;;MAAtC,a;K;IASO,6C;MAAA,8B;MAAS,uB;K;8FACW,Y;MAAQ,OAAA,gBAAY,K;K;+CAC3C,Y;MAAkC,OAAA,gBAAY,U;K;sDAC9C,mB;MAAgD,OAA A,gBAAY,gBAAS,OAAT,C;K;mDAC5D,iB;MACI,oCAAA,2BAAkB,KAAIB,EAAYB,SAAZB,C;MACb,OAAO,6BAAY,KAAZ,C;K;qDAEX,mB;MAES,Q;MAAL,IAAI,eAAC,0EAAD,OAAJ,C;QAAgC,OAAO,E;MACvC,OxCs rBO,UwCtrBA,gBxCsrBR,QAAQ,EwCtrBoB,O3EgOF,KmCsdIB,C;K;yDwCprBX,mB;MAES,Q;MAAL,IAAI,eAAC,0EAAD,OAAJ,C;QAAgC,OAAO,E;MACvC,OxCy6BO,cwCz6BA,gBxCy6BR,QAAQ,EwCz6BwB,O3E2NN,KmC8sBIB,C;K;;IwC/7BnB,6B;MAMI,4C;K;IA2BO,6C;MAAA,8B;MAAS,uB;K;8FACW,Y;MAAQ,OAAA,gBAAY,K;K;+CAC3C,Y;MAAkC,OAAA,gBAAY,U;K;sDAC9C,mB;MAAiD,OAAA,gBAAY,gBAAS,OAAT,C;K;mDAC7D,iB;MACI,oCAAA,2BAAkB,KAAIB,EAAYB,SAAZB,C;MACb,OAAO,6BAAY,KAAZ,C;K;qDAEX,mB;MAES,Q;MAAL,IAAI,eAAC,0EAAD,QAAJ,C;QAAiC,OAAO,E;MACx C,OxCqqBO,UwCrqBA,gBxCqqBR,QAAQ,EwCrqBoB,O3DgNA,KmBqdpB,C;K;yDwCnqBX,mB;MAES,Q;MAAL,IAAI,eAAC,0EAAD,QAAJ,C;QAAiC,OAAO,E;MACx C,OxCw5BO,cwCx5BA,gBxCw5BR,QAAQ,EwCx5BwB,O3D2MJ,KmB6sBpB,C;K;;IwC96BnB,6B;MAMI,4C;K;IA2BO,6C;MAAA,8B;MAAS,uB;K;8FACW,Y;MAAQ,OAAA,gBAAY,K;K;+CAC3C,Y;MAAkC,OAAA,gBAAY,U;K;sDAC9C,mB;MAAiD,OAAA,gBAAY,gBAAS,OAAT,C;K;mDAC7D,iB;MACI,oCAAA,2BAAkB,KAAIB,EAAYB,SAAZB,C;MACb,OAAO,6BAAY,KAAZ,C;K;qDAEX,mB;MAES,Q;MAAL,IAAI,eAAC,0EAAD,QAAJ,C;QAAiC,OAAO,E;MACx C,OxCopBO,UwCppBA,gBxCopBR,QAAQ,EwCppBoB,O5EkIA,KoCkhBpB,C;K;yDwClpBX,mB;MAES,Q;MAAL,IAAI,eAAC,0EAAD,QAAJ,C;QAAiC,OAAO,E;MACx C,OxCu4BO,cwCv4BA,gBxCu4BR,QAAQ,EwCv4BwB,O5E6HJ,KoC0wBpB,C;K;;IwC75BnB,8B;MAMI,4C;K;IA2BO,6C;MAAA,8B;MAAS,uB;K;8FACW,Y;MAAQ,OAAA,gBAAY,K;K;+CAC3C,Y;MAAkC,OAAA,gBAAY,U;K;sDAC9C,mB;MAAkD,OAAA,gBAAY,gBAAS,OAAT,C;K;mDAC9D,iB;MACI,oCAAA,2BAAkB,KAAIB,EAAYB,SAAZB,C;MACb,OAAO,6BAAY,KAAZ,C;K;qDAEX,mB;MAES,Q;MAAL,IAAI,eAAC,0EAAD,SAAJ,C;QAAkC,OAAO,E;MACzC,OxCmoBO,UwCnoBA,gBxCmoBR,QAAQ,EwCnoBoB,O1EkHE,KkCihBtB,C;K;yDwCjoBX,mB;MAES,Q;MAAL,IAAI,eAAC,0EAAD,SAAJ,C;QAAkC,OAAO,E;MACzC,OxCs3BO,cwCt3BA,gBxCs3BR,QAAQ,EwCt3BwB,O1E6GF,KkCywBtB,C;K;;IwC54BnB,8B;MAMI,4C;K;ICtIJ,qC;MAII,SAAS,SID+BiC,I;MkD9B1C,OAAa,CAAN,gBAAc,EAAd,KACU,EAAN,gBAAc,EADIB,KAEL,OAAM,GAFV,KAGI,KAAK,IAAL,KACC,OAAM,IAAN,KACS,IAAN,gBAAc,IADjB,KAEG,OAAM,IAFT,IAGG,OAAM,IAHT,IAIG,OAAM,IAJT,IAKG,OAAM,IALT,IAMG,OAAM,KAPV,CAHJ,C;K;;;mCCTP,gB;;K;;ICAJ,wB;K;;IAIA,wB;K;;IAIA,wB;K;;IAKiC,uB;MAAC,oB;QAAA,OAA0B,E;MAA1B,gB;K;;IAEIC,kB;K;;IAqCqC,sB;MAAC,gB;K;;IAgCN,4B;MAAC,sB;K;;IAEjC,uB;K;;IA8DmC,4B;MAAC,kB;K;;IAEpC,oB;K;;IAMCA,+B;K;;ICvLA,oB;K;;IAIA,wB;K;;oF7DLA,qB;MAKqE,uCoCHtB,E;K;iGpCK/C,yB;MAAA,kD;MAAA,4B;QAQsE,mBAAY,SAAZ,C;O;KARtE,C;IAUA,iC;MAGI,OAAsB,UAAy,QAAvB,KAAmC,SAa9C,GACe,UAAy,UAD3B,GAGI,gBAAgB,UAAhB,C;K;IAGR,qC;MAEI,YoC1B2C,E;MpC2B3C,eAAe,UAAW,W;MAC1B,OAAO,QAAS,UAAhB,C;QACU,KAAy,MAAK,QAAS,OAAd,C;MACTB,OAAO,K;K;IAGX,8C;MAQc,Q;MANV,IAAI,KAAM,OAAN,GAAa,UAAW,KAA5B,C;QACI,OA AO,gBAAgB,UAAhB,C;;MAEX,eAAe,UAAW,W;MAC1B,YAAy,C;MACZ,OAAO,QAAS,UAAhB,C;QACI,MAAM,YAAN,EAAM,oBAAN,UAAiB,QAAS,O;;MAE9B,IAAI,QAAQ,KAAM,OAAlB,C;QACI,MAAM,KAAAN,IA Ae,I;;MAEnB,OAAO,K;K;IAIX,yB;MAG6C,sBAAY,OAaz,E;K;wGAE7C,yB;MAAA,+D;MAAA,gC;QAI0B,gB Aaf,gB;QAAqB,aJW5B,W;QIXA,OJYO,SIZoC,Q;O;KAJ/C,C;yGAOA,yB;MAAA,4E;MAAA,gE;MAAA,0C;QA II,qBAAqB,QAArB,C;QAC8B,gBAAvB,eAAa,QAAb,C;QAA6B,aJGpC,W;QIHA,OJIO,SIJ4C,Q;O;KALvD,C;IA SA,wB;MAG2C,oBAAU,OAAV,E;K;sGAE3C,yB;MAAA,uE;MAAA,gC;QAI8B,gBAAnB,oB;QAAyB,aJvHc,W; QIUA,OJTO,SISwC,Q;O;KAJnD,C;wGAOA,yB;MAAA,wE;MAAA,0C;QAI5C,gBAA3B,mBAAiB,QAAjB,C;QA AiC,aJjBxC,W;QIiBA,OJhBO,SIgBgD,Q;O;KAJ3D,C;IAQA,qB;MAIuD,oBAAU,IAAV,E;K;sGAEvD,yB;MAAA

,wE;MAAA,gC;QAIiC,gBAAtB,oB;QAA4B,aJ/BnC,W;QI+BA,OJ9BO,SI8B2C,Q;O;KAJtD,C;uGAOA,yB;MAA
A,uE;MAAA,0C;QAIyC,gBAA9B,mBAAoB,QAApB,C;QAAoC,aJtC3C,W;QIsCA,OJrCO,SIqCmD,Q;O;KAJ9D,
C;IAQA,mC;MAOqB,Q;MAAA,kC;MAAJB,iBAAc,CAAd,yB;QACI,sBAAK,KAAL,EAAC,KAAd,C;;K;IAIR,+B;
MAMuD,sBAAQ,4BAAR,C;K;IAEvD,6B;MAIwE,kBAAhB,0B;MAAwB,uB;MAAxB,OJJE7C,W;K;IlmEX,4B;M
AQI,gBAAgB,SAAhB,EAAsB,cAAtB,C;K;IAGJ,2C;MAQI,gBAAgB,SAAhB,EAAsB,UAAtB,C;K;IAGJ,2C;MA
CI,IAAI,IAAK,KAAL,IAAa,CAAJB,C;QAAoB,M;MAEpB,YAAY,YAAY,IAAZ,C;MACZ,gBAAC,KAAd,EAAq
B,UAArB,C;MAEA,aAAU,CAAV,MAAkB,KAAM,OAAxB,M;QACI,iBAAK,CAAL,EAAU,MAAM,CAAN,CA
AV,C;;K;IAIR,uC;MACI,OAAO,gBAAkB,IAAI,B,O;K;IAGX,iF;MAII,oCAAa,2BAAkB,UAAIB,EAA8B,QAA9B
,EAAwC,MAAO,OAA/C,C;MACb,gBAAgB,WAAW,UAAx,I;MACHB,oCAAa,2BAAkB,iBAAI,B,EAAqC,oBAA
oB,SAApB,IAArC,EAAoE,WAAy,OAAhF,C;MAEb,IAAI,WAAkB,QAAO,WAAp,CAAIB,IAAyC,WAAkB,QA
AO,MAAP,CAA/D,C;QACI,eAAsB,MAAY,UAAS,UAAT,EAAqB,QAArB,C;QACtB,WAAy,KAAl,QAAJ,EAA
c,iBAAd,C;;QAExB,IAAI,WAAW,WAAx,IAA0B,qBAAqB,UAAnd,C;UACI,iBAAC,CAAd,UAAsB,SAAtB,U;Y
ACI,YAAY,oBAAoB,KAApB,IAAZ,IAAyC,OAAO,aAAa,KAAb,IAAP,C;;;UAG7C,mBAAC,YAAY,CAAZ,IAA
d,aAAmC,CAAnC,Y;YACI,YAAY,oBAAoB,OAApB,IAAZ,IAAyC,OAAO,aAAa,OAAb,IAAP,C;;;K;8GAMzD,
qB;MAEgF,gB;K;kGAehF,yB;MAAA,4D;MAAA,4B;QAC8E,OAAK,aAAL,SAAK,C;O;KADnF,C;sGAIA,gC;M
AEI,OAAI,SAAJ,GAEL,SAFJ,GAII,SN83BoB,Q;K;IM13B5B,mC;MAEI,IAAI,QAAQ,CAAZ,C;QACI,oB;;MAEJ,
OAAO,K;K;IAGX,mC;MAEI,IAAI,QAAQ,CAAZ,C;QACI,oB;;MAEJ,OAAO,K;K;IAIX,mC;MAIqD,mB;K;IAEr
D,wC;MPzNI,IAAI,EOgOI,YAAY,CPhOhB,CAAJ,C;QACI,cO+NqB,gC;QP9NrB,MAAM,gCAAyB,OAAQ,WA
AjC,C;;K;IOiOd,8C;MAAoE,Y;K;I8D1PV,qC;MAAiC,6B;K;uDAlvF,mB;MACI,qB;MACA,eAAe,e;MACf,OAA
O,QAAS,UAAhB,C;QACI,IAAI,OAAA,QAAS,OAAAT,EAAMB,OAAAnB,CAAJ,C;UACI,QAAS,S;UACT,OAAO,I
;;;MAGf,OAAO,K;K;yDAGX,oB;MAGoB,Q;MAFhB,qB;MACA,eAAe,K;MACC,0B;MAAhB,OAAgB,cAAhB,C
;QAAgB,yB;QACZ,IAAI,eAAI,OAAJ,CAAJ,C;UAAkB,WAAW,I;;MAEjC,OAAO,Q;K;IAKuC,sE;MAAA,qB;Q
AAE,OAAm,gBAAN,mB;O;K;4DAFpD,oB;MAEY,Q;MADR,qB;MACA,OAAoC,YAA5B,iEAA4B,EAAU,oDA
AV,C;K;IAKU,sE;MAAA,qB;QAAE,QAAO,gBAAP,mB;O;K;4DAFpD,oB;MAEY,Q;MADR,qB;MACA,OAAoC
,YAA5B,iEAA4B,EAAU,oDAAV,C;K;gDAGxC,Y;MACI,qB;MACA,eAAe,IAAK,W;MACpB,OAAO,QAAS,UA
AhB,C;QACI,QAAS,O;QACT,QAAS,S;;K;iDAIjB,Y;MAE8B,OAAA,IAAK,U;K;yDAGnC,Y;K;;IC3CgD,+B;MA
AiC,oC;MACJf,gBAA8B,C;K;8CAM9B,mB;MAMI,qB;MACA,iBAAI,SAAJ,EAAU,OAAV,C;MACA,OAAO,I;
K;mDAGX,2B;MAMc,UACF,M;MANR,oCAAa,4BAAMB,KAAAnB,EAA0B,SAAI1B,C;MAEb,qB;MACA,aAAa,
K;MACb,cAAc,K;MACJ,0B;MAAV,OAAU,cAAV,C;QAAU,mB;QACN,kBAAI,eAAJ,EAAI,uBAAJ,WAAc,CA
Ad,C;QACA,UAAU,I;;MAEd,OAAO,O;K;0CAGX,Y;MACI,qB;MACA,yBAAY,CAAZ,EAAe,SAAF,C;K;IAKiB,
gE;MAAA,qB;QAAE,OAAm,gBAAN,mB;O;K;sDAFvB,oB;MACI,qB;MACA,OAAO,kBAAU,8CAAV,C;K;IAK
U,gE;MAAA,qB;QAAE,QAAO,gBAAP,mB;O;K;sDAFvB,oB;MACI,qB;MACA,OAAO,kBAAU,8CAAV,C;K;6C
AIX,Y;MAAqD,iD;K;mDAErD,mB;MAAoD,0BAAQ,OAAr,KAAoB,C;K;kDAExE,mB;MACqB,Q;MAAA,6B;
MAAJB,iBAAC,CAAd,yB;QACI,IAAI,wBAAI,KAAJ,GAAC,OAAAd,CAAJ,C;UACI,OAAO,K;;;MAGf,OAAO,E;K
;sDAGX,mB;MACI,iBAAC,sBAAd,WAA+B,CAA/B,U;QACI,IAAI,wBAAI,KAAJ,GAAC,OAAAd,CAAJ,C;UACI,
OAAO,K;;;MAGf,OAAO,E;K;iDAGX,Y;MAA6D,iCAAa,CAAb,C;K;yDAC7D,iB;MAAuE,sDAAIb,KAAjB,C;K
;oDAGvE,8B;MAA4E,uCAAQ,IAAR,EAAC,SAAd,EAAyB,OAAzB,C;K;wDAE5E,8B;MAII,eAAe,0BAAa,SAAb
,C;MACf,YAAO,UAAU,SAAV,I;MnEuDX,iBAAC,CAAd,UAAsB,KAAtB,U;QmEtDiB,e;QACA,iB;;K;2CAIjB,i
B;MAMI,IAAI,UAAU,IAAd,C;QAAoB,OAAO,I;MAC3B,IAAI,2BAAJ,C;QAAuB,OAAO,K;MAE9B,OAAO,oC
AAa,uBAAC,IAAd,EAAoB,KAApB,C;K;6CAGxB,Y;MAG+B,OAAA,oCAAa,yBAAGB,IAAhB,C;K;IAG5C,kD;
MAAA,oB;MACI,eACsB,C;MACtB,cAIqB,E;K;yDAErB,Y;MAAkC,sBAAQ,gB;K;sDAE1C,Y;MAEW,Q;MADP
,IAAI,CAAC,cAAL,C;QAAgB,MAAM,6B;MACtB,eAAO,mBAAP,EAAO,2BAAP,O;MACA,OAAO,wBAAI,WA
AJ,C;K;wDAGX,Y;MtE5CJ,IAAI,EsE6CU,gBAAQ,EtE7CIB,CAAJ,C;QACI,csE4CwB,sE;QtE3CxB,MAAM,6BA
AsB,OAAQ,WAA9B,C;;MsE6CF,6BAAS,WAAAT,C;MACA,eAAQ,W;MACR,cAAO,E;K;;IAOqB,6D;MAHpC,o
B;MAGmD,wD;MAG3C,oCAAa,4BAAMB,KAAAnB,EAA0B,WAAyB,KAAAnD,C;MACb,eAAa,K;K;iEAGjB,Y;M
AAsC,sBAAQ,C;K;+DAE9C,Y;MAAGC,mB;K;8DAEHc,Y;MACI,IAAI,CAAC,kBAAL,C;QAAoB,MAAM,6B;M
AE1B,eAAO,mCAAP,EAAO,YAAP,C;MACA,OAAO,wBAAI,WAAJ,C;K;mEAGX,Y;MAAoC,sBAAQ,CAAR,I;
K;+DAEpC,mB;MACI,wBAAI,YAAJ,EAAW,OAAx,C;MACA,mC;MACA,cAAO,E;K;+DAGX,mB;MtEIFJ,IAA

I,EsEmFU,gBAAQ,EtEnFIB,CAAJ,C;QACI,csEkFwB,4E;QtEjFxB,MAAM,6BAAsB,OAAQ,WAA9B,C;;MsEkFF ,wBAAI,WAAJ,EAAU,OAAV,C;K;;IAIgb,+D;MAAuF,8B;MAAtF,kB;MAA0C,4B;MAC/D,eAAyB,C;MAGrB,o CAAa,2BAAkB,gBAAIB,EAA6B,OAA7B,EAA5C,WAAK,KAA3C,C;MACb,eAAa,UAAU,gBAAV,I;K;wDAGjB ,0B;MACI,oCAAa,4BAAmB,KAAAnB,EAA0B,YAA1B,C;MAEb,WAAK,aAAI,mBAAY,KAAZ,IAAJ,EAAuB,OA AvB,C;MACL,mC;K;wDAGJ,iB;MACI,oCAAa,2BAAkB,KAAIB,EAAyB,YAAzB,C;MAEb,OAAO,wBAAK,mB AAY,KAAZ,IAAL,C;K;6DAGX,iB;MACI,oCAAa,2BAAkB,KAAIB,EAAyB,YAAzB,C;MAEb,aAAa,WAAK,kB AAS,mBAAY,KAAZ,IAAT,C;MACIB,mC;MACA,OAAO,M;K;wDAGX,0B;MACI,oCAAa,2BAAkB,KAAIB,EA AyB,YAAzB,C;MAEb,OAAO,WAAK,aAAI,mBAAY,KAAZ,IAAJ,EAAuB,OAAvB,C;K;mGAGO,Y;MAAQ,mB ;K;2DAE/B,Y;MAA+C,WAAK,iB;K;;;ICxMN,8B;MAAiC,sB;MAwCnF,uBAAoC,I;MA+CpC,yBAA6C,I;K;IAIF R,oD;MAAC,wB;MAGlC,gBAAqB,K;K;IFAHa,Y;MAAA,yB;K;uGAKZ,Y;MAAQ,oB;K;8DAE9B,oB;MAKI,eA Ae,IAAK,S;MACpB,gBAAc,Q;MACd,OAAO,Q;K;wDAGX,Y;MAA+B,iEAAc,IAAd,C;K;wDAC/B,Y;MAAKC,i EAAc,IAAd,C;K;sDACIC,iB;MAA4C,+DAAY,IAAZ,EAakB,KAAIB,C;K;;IAIB5C,8E;MAAA,wE;MAAsC,2CA AK,KAAM,IAAX,EAagB,KAAM,MAAtB,C;MAAtC,Y;K;IASBJ,+C;MACsE,6B;K;mEACIE,mB;MAAmD,kCA Ac,OAAAd,C;K;iEEnD,mB;MAAiD,gCAAY,OAAZ,C;K;;yCAIrD,Y;MACI,YAAQ,Q;K;IAOQ,+F;MAAA,sD;M AAS,6B;K;uFACb,mB;MAAwC,MAAM,qCAA8B,8BAA9B,C;K;mFAC9C,Y;MACI,4BAAwB,Q;K;4FAG5B,mB ;MAAsD,sDAAY,OAAZ,C;K;IAI3C,oH;MAAA,kD;K;4GACH,Y;MAAkC,OAAA,0BAAc,U;K;yGACHd,Y;MAA yB,OAAA,0BAAc,OAAO,I;K;2GAC9C,Y;MAAwB,0BAAc,S;K;;sFAL9C,Y;MACI,oBAAoB,oCAAQ,W;MAC5 B,6G;K;0FAOJ,mB;MACI,qB;MACA,IAAI,+CAAY,OAAZ,CAAJ,C;QACI,4BAAwB,cAAO,OAAp,C;QACxB,O AAO,I;;MAEX,OAAO,K;K;oIAGY,Y;MAAQ,OAAA,4BAAwB,K;K;4FAEvD,Y;MAAsC,4BAAwB,iB;K;;0FA9B 1E,Y;MACI,IAAI,4BAAJ,C;QACI,6F;;MA+BJ,OAAO,mC;K;kDAKf,gB;MAEyB,Q;MADrB,qB;MACqB,OAAA, I9E8Q2D,QAAQ,W;M8E9QxF,OAAqB,cAArB,C;QAAqB,wB;QAAf,U9EiMsD,U;Q8EjMjD,Y9E8MiD,Y;Q8E7 MxD,iBAAI,GAAJ,EAAS,KAAT,C;;K;IAQc,iG;MAAA,sD;MAAS,oC;K;yFACf,mB;MAAwC,MAAM,qCAA8B, gCAA9B,C;K;qFAC9C,Y;MAAuB,4BAAwB,Q;K;8FAE/C,mB;MAAsD,wDAAc,OAAAd,C;K;IAI3C,sH;MAAA,k D;K;8GACH,Y;MAAKC,OAAA,0BAAc,U;K;2GACHd,Y;MAAyB,OAAA,0BAAc,OAAO,M;K;6GAC9C,Y;MAA wB,0BAAc,S;K;;wFAL9C,Y;MACI,oBAAoB,oCAAQ,W;MAC5B,+G;K;sIAOmB,Y;MAAQ,OAAA,4BAAwB,K; K;8FAEvD,Y;MAAsC,4BAAwB,iB;K;;4FAnB1E,Y;MACI,IAAI,8BAAJ,C;QACI,iG;;MAoBJ,OAAO,qC;K;gDA Gf,e;MACI,qB;MACA,WAAW,YAAQ,W;MACnB,OAAO,IAAK,UAAZ,C;QACI,YAAY,IAAK,O;QACjB,QAA Q,KAAM,I;QACd,IAAI,YAAO,CAAP,CAAJ,C;UACI,YAAY,KAAM,M;UACIB,IAAK,S;UACL,OAAO,K;;MA Gf,OAAO,I;K;kDAIX,Y;K;;IC3I+C,8B;MAAiC,oC;K;0CAEHf,iB;MAMI,IAAI,UAAU,IAAd,C;QAAoB,OAAO,I; MAC3B,IAAI,0BAAJ,C;QAAsB,OAAO,K;MAC7B,OAAO,mCAAY,mBAAU,IAAV,EAagB,KAahB,C;K;4CAG vB,Y;MAG+B,OAAA,mCAAY,2BAAkB,IAAIB,C;K;;ICbT,0B;MAAuD,8B;MAAIC,4B;MACvD,4BAakC,K;K;g CAKBIC,Y;MAEI,qB;MACA,4BAAa,I;MACb,OAAO,I;K;qCAGX,Y;K;iDAGA,uB;K;iFAG8B,Y;MAAQ,OAAA, oBAAM,O;K;sCAC5C,iB;MACyC,Q;MAAA,oCAAM,0BAAW,KAAX,CAAN,4D;K;sCACzC,0B;MAIW,IAAa,I; MAHpB,qB;MACA,0BAAW,KAAX,C;MAEoB,gBAAb,qBAAM,KAAN,C;MAAqB,qC;MAA5B,OAAO,CAAa,O tE8BjB,SsE9BI,2D;K;oCAGX,mB;MACI,qB;MACM,oBAAY,MAAK,OAAL,C;MACIB,qC;MACA,OAAO,I;K;s CAGX,0B;MACI,qB;MACM,oBAAY,QAAO,mCAAOB,KAAPB,CAAP,EAAMC,CAAnC,EAAsC,OAAIC,C;MA ClB,qC;K;yCAGJ,oB;MACI,qB;MACA,IAAI,QAAS,UAAb,C;QAAwB,OAAO,K;MAE/B,uBAAA,oBxEioDoB,Q MjrD0C,YkEgDrD,QIEhDqD,CNirD1C,C;MwEhoDpB,qC;MACA,OAAO,I;K;yCAGX,2B;MACI,qB;MACA,mC AAoB,KAAPB,C;MAEA,IAAI,UAAS,SAAb,C;QAAMB,OAAO,oBAAO,QAAP,C;MACIB,IAAI,QAAS,UAAb,C ;QAAwB,OAAO,K;MAE3B,IADE,KACF,e;QAAQ,OAAO,oBAAO,QAAP,C;WACf,IAFE,KAef,O;QAAK,uBIE 7DqD,YkE6D7C,QIE7D6C,CNirD1C,QwEpnD6B,oBxEonD7B,C;;QwEnnDR,uBAAoC,cAA5B,oBAA4B,EAAV, CAAU,EAAP,KAAO,CAAY,QIE9DE,YkE8DK,QIE9DL,CkE8DF,EAA4C,cAAN,oBAAM,EAAY,KAAZ,EAAM B,SAAnB,CAA5C,C;;MAG5D,qC;MACA,OAAO,I;K;2CAGX,iB;MACI,qB;MACA,0BAAW,KAAX,C;MACA,q C;MACA,OAAW,UAAS,sBAAb,GACG,oBAAY,MADf,GAGG,oBAAY,QAAO,KAAP,EAAC,CAAd,CAAIB,CA AmC,CAAnC,C;K;uCAGR,mB;MAEkB,Q;MADd,qB;MACc,2B;MAAd,mD;QACI,IAAI,4BAAM,KAAN,GAAg B,OAAhB,CAAJ,C;UACU,oBAAY,QAAO,KAAP,EAAC,CAAd,C;UACIB,qC;UACA,OAAO,I;;MAGf,OAAO,K; K;8CAGX,8B;MACI,qB;MACA,qC;MACM,oBAAY,QAAO,SAAP,EAakB,UAAU,SAAV,IAAIB,C;K;gCAGtB, Y;MACI,qB;MACA,uB9BhHuC,E;M8BiHvC,qC;K;wCAIJ,mB;MAA+C,OAAM,QAAN,oBAAM,EAAQ,OAAR,

C;K;4CAErD,mB;MAAmD,OAAM,YAAN,oBAAM,EAAY,OAAZ,C;K;mCAeZD,Y;MAA0B,uBAAc,oBAAd,C;K;0CAE1B,iB;MAGe,UAGL,MAHK,EAMO,M;MAPIB,IAAI,KAAM,OAAN,GAAa,SAAJB,C;QACI,OAAO,2D;;MAGc,gBAAXB,eAAK,SAAL,IAAK,gBAAL,yB;MxEuWBL,UAAU,SAAV,EwEvWbS,C,KxEuWbTc,EAD+F,CA C/F,EADoH,CACpH,EADuI,gBACvI,C;MwErwBI,IAAI,KAAM,OAAN,GAAa,SAAJB,C;QACI,MAAM,SAAN,IAAc,6E;;MAGIB,OAAO,K;K;kCAGX,Y;MACI,OAAO,EAAS,MAAM,MAAK,oBAAL,C;K;yCAI1B,Y;MACI,IAAI,yBAAJ,C;QAAgB,MAAM,oC;K;+CAG1B,iB;MACI,oCAAa,kCAAYB,SAAZB,C;MADoB,Y;K;wDAIrc,iB;MACI,oCAAa,mCAA0B,SAAI1B,C;MAD6B,Y;K;;IAIJ9C,+B;MAAA,mD;MAG8B,sB9BRa,E8BQb,C;MAH9B,Y;K;IAKA,kD;MAAA,mD;MAIkD,sB9BdP,E8BcO,C;MAJID,Y;K;IAMA,2C;MAAA,mD;MAGqD,sBIENa,YkEMR,QIENQ,CkEMb,C;MAHrD,Y;K;ICrBJ,0C;MACI,IAAI,6BAAJ,C;QACU,KAAY,MAAK,UAAAL,C;;QAEIB,UAAU,KAAY,EAAwC,CAAxC,EAAd,cAAN,KAAM,CAAjD,EAA4D,eAAW,UAAAX,CAA5D,C;;K;IAMiB,kD;MAAA,uB;QAAgB,OAAA,kBAAW,SAAQ,CAAR,EAAW,CAAX,C;O;K;IAFpD,4C;MACI,IAAI,6BAAJ,C;QACI,iBAAiB,gC;QACX,KAAY,MAAK,UAAAL,C;;QAEIB,UAAU,KAAY,EAAwC,CAAxC,EAAd,cAAN,KAAM,CAAjD,EAA4D,UAA5D,C;;K;IAIR,gE;MACI,IAAI,aAAY,UAAU,CAAV,IAAZ,CAAJ,C;QACI,UAAU,KAAY,EAAwC,SAAXC,EAAMD,UAAU,CAAV,IAANd,EAAGe,UAAhE,C;;K;IAMiB,gC;MAAgB,OAAE,iBAAF,CAAE,EAAU,C AAV,C;K;IAF3C,0B;MACI,IAAI,6BAAJ,C;QACI,iBAAiB,gB;QACX,KAAY,MAAK,UAAAL,C;;QAEIB,UAAU,KAAY,EAAwC,CAAxC,EAAd,cAAN,KAAM,CAAjD,EAA4D,cAA5D,C;;K;;IAaa,kD;MAAoB,QAAC,IAAM,C AAP,KAAa,IAAM,CAAnB,K;K;IARzC,uC;MACI,sC;QAAiC,OAAjC,yB;;MACA,4BAA4B,K;MAE5B,YAAY,E;MAGZ,iBAAc,CAAd,UAAAsB,GAAtB,U;QAAiC,KAAY,MAAK,KAAL,C;MAC7C,iBAAiB,kC;MACX,KAAY,M AAK,UAAAL,C;MACIB,mBAAc,CAAd,YAAsB,KAAM,OAA5B,Y;QACI,QAAQ,MAAM,UAAQ,CAAR,IAAN,C ;QACR,QAAQ,MAAM,OAAN,C;QACR,IAAI,CAAC,IAAM,CAAP,OAAc,IAAM,CAAPb,KAA0B,KAAC,CAA nC,C;UAAAsC,OAAO,K;;MAEjD,4BAA4B,I;MAC5B,OAAO,I;K;IAIX,2D;MACI,aAAa,gBAAMb,KAAM,OAAz B,O;MACb,aAAa,YAAU,KAAY,EAAiB,MAAJB,EAAyB,KAazB,EAAGC,YAAhC,EAA8C,UAA9C,C;MACb,IA AI,WAAW,KAaf,C;QACI,aAAU,KAAY,OAAiB,YAAjB,M;UAA+B,MAAM,CAAN,IAAW,OAAO,CAAP,C;;K; IAII,4D;MAEI,IAAI,UAAAsB,GAAb,C;QACI,OAAO,K;;MAGX,aAAa,CAAC,QAAQ,GAAR,IAAD,IAAGB,CAA hB,I;MACb,WAAW,YAAU,KAAY,EAAiB,MAAJB,EAAyB,KAazB,EAAGC,MAAhC,EAAwC,UAAxC,C;MAC X,YAAY,YAAU,KAAY,EAAiB,MAAJB,EAAyB,SAAS,CAAT,IAAZB,EAAqC,GAARc,EAA0C,UAA1C,C;MAE Z,aAAiB,SAAS,MAAb,GAAqB,KAArB,GAAGC,M;MAG7C,gBAAGB,K;MAChB,iBAAiB,SAAS,CAAT,I;MACj B,aAAU,KAAY,OAAiB,GAAjB,M;QAEQ,iBAAa,MAAb,IAAuB,cAAc,GAARc,C;UACI,gBAAGB,KAAC,SAAL ,C;UACHB,iBAAiB,MAAM,UAAN,C;UAEjB,IAAI,UAAW,SAAQ,SAAR,EAAmB,UAAnB,CAAX,IAA6C,CAAj D,C;YACI,OAAO,CAAP,IAAY,S;YACZ,6B;;YAEA,OAAO,CAAP,IAAY,U;YACZ,+B;;eAGR,iBAAa,MAAb,C; UACI,OAAO,CAAP,IAAY,KAAC,SAAL,C;UACZ,6B;;UAGA,OAAO,CAAP,IAAY,MAAM,UAAN,C;UACZ,+ B;;MAMZ,OAAO,M;K;ICrGX,4C;MAMoB,UACM,M;MAHtB,IAAI,iBAAJ,C;QAAkB,OAAO,C;MACzB,aAAa, C;MACb,wBAAGB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAEQ,oB;UAAmB,U;;UACnB,I1BFiC,MAAa,Y0BEn C,O1BFmC,C0BE9C,C;YAAwD,iCAAhC,OAAgC,C;iBAExD,uC;YAAmC,2BAAR,OAAQ,C;eACnC,wC;YAAm C,2BAAR,OAAQ,C;eACnC,sC;YAAmC,2BAAR,OAAQ,C;eACnC,uC;YAAmC,2BAAR,OAAQ,C;;YAEA,kBAA R,OAAQ,C;;QATvC,wB;QAYA,SAAS,MAAK,MAAL,QAAC,WAAAd,I;;MAEb,OAAO,M;K;;ICTP,uC;MAAA,2 C;K;2DACI,0B;MAA2D,sBAAU,MAAV,C;K;gEAE3D,iB;MAA6C,Q;MAAA,wEAAqB,C;K;;IAHtE,mD;MAAA ,kD;QAAA,iC;;MAAA,2C;K;;MC0BA,iC;MAKA,8B;MA6CA,0BAAMe,I;IAzEnE,kC;MAAA,oB;MAA+B,8C; K;2CAE3B,mB;MAAYd,MAAM,qCAA8B,iCAA9B,C;K;uCAC/D,Y;MACI,WAAa,Q;K;uDAGjB,mB;MAAGe,O AAA,WAAa,uBAAc,OAAd,C;K;0CAE7E,Y;MAAwE,OAAA,iCAAY,W;K;qDAEPf,mB;MACI,IAAI,iBAAS,OA AT,CAAJ,C;QACI,WAAa,cAAO,OAAQ,IAAf,C;QACb,OAAO,I;;MAEX,OAAO,K;K;wFAGY,Y;MAAQ,OAAA, WAAa,K;K;;8BA6ChD,Y;MACI,0BAAY,Q;K;0CAIhB,e;MAAMd,OAAA,0BAAY,gBAAS,GAAT,C;K;4CAE/D, iB;MAAMe,gBAAZ,0B;MAAY,c;;QvE+mDnD,Q;QADhB,IAAI,wCAAsB,mBAA1B,C;UAAqC,aAAO,K;UAAP, e;;QACrB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAAM,IuE/mDmD,uBAAS,gBvE+mD9C,OuE/mDwD,MA AV,QvE+mD5D,C;YAAwB,aAAO,I;YAAP,e;;;QAC9C,aAAO,K;;MuEhndgD,iB;K;kFAInD,Y;MACI,IAAI,+BA AJ,C;QACI,0BAAW,qB;;MAEf,OAAO,sC;K;uCAGf,Y;MAAGf,iC;K;kCAEHf,e;MAA+C,OAAA,0BAAY,WAAI ,GAAJ,C;K;oCAE3D,sB;MAAGD,OAAA,0BAAY,aAAI,GAAJ,EAAS,KAAT,C;K;qCAE5D,e;MAAYc,OAAA,0B AAY,cAAO,GAAP,C;K;+EAEvB,Y;MAAQ,OAAA,0BAAY,K;K;;IA5DID,0C;MAAA,iD;MAAuD,8B;MAvC3D,

mB;MAwCQ,8BAAmB,W;MACnB,2BAAGB,WAAY,S;MAFhC,Y;K;IAKA,+B;MAAA,iD;MAGuB,aAAK,kEAA
L,Q;MAHvB,Y;K;IAKA,4D;MAAA,iD;MAQ8D,qB;M7EpC9D,IAAI,E6EsCQ,mBAAmB,C7EtC3B,CAAJ,C;QA
CI,c6EqCgC,+C;Q7EpChC,MAAM,gCAAyB,OAAQ,WAAjC,C;MAFV,IAAI,E6EuCQ,cAAc,C7EvCtB,CAAJ,C;
QACI,gB6EsC2B,yC;Q7ErC3B,MAAM,gCAAyB,SAAQ,WAAjC,C;M6E0BV,Y;K;IAcA,gD;MAAA,iD;MAA2C
,eAAK,eAAL,EAAaB,GAAtB,Q;MAA3C,Y;K;IAGA,yC;MAAA,iD;MAG8C,qB;MAC1C,KAAK,gBAAO,QAAP,
C;MAJT,Y;K;IAqCJ,4B;MAK8E,gBAAnE,aAAmB,gEAAnB,C;MAA2E,wB;MAAlF,O1EvCO,S;K;M2EjEP,uB;;
kCAyCA,mB;MACI,UAAU,gBAAl,aAAI,OAAJ,EAAa,IAAb,C;MACd,OAAO,W;K;8BAGX,Y;MACI,gBAAl,Q;
K;uCAOR,mB;MAA6D,OAAA,gBAAl,mBAAY,OAAZ,C;K;gCAEjE,Y;MAAyC,OAAA,gBAAl,U;K;iCAE7C,Y;
MAAqD,OAAA,gBAAl,KAAK,W;K;qCAE9D,mB;MAAkD,OAAA,gBAAl,cAAO,OAAP,CAAJ,Q;K;+EAEPB,Y
;MAAQ,OAAA,gBAAl,K;K;IA5D1C,6B;MAAA,iD;MAGoB,8B;MAZxB,mB;MAAq,oBAAM,gB;MAJV,Y;K;IA
OA,yC;MAAA,iD;MAG2C,8B;MAAnB/C,mB;MAoBQ,oBAAM,eAAGB,QAAS,KAAzB,C;MACN,qBAAO,QAAP,
C;MALJ,Y;K;IAQA,4D;MAAA,iD;MAQ2D,8B;MAhC/D,mB;MAiCQ,oBAAM,eAAGB,eAAhB,EAAiC,UAAjC,
C;MATV,Y;K;IAYA,gD;MAAA,iD;MAA2C,eAAK,eAAL,EAAaB,GAAtB,Q;MAA3C,Y;K;IAEA,oC;MAAA,iD;
MAM0C,8B;MA5C9C,mB;MA6CQ,oBAAW,G;MAPf,Y;K;IAmCJ,+B;MAKuC,gBAA5B,eAAQ,eAAR,C;MAAo
C,6B;MAA3C,O3ENO,S;K;I4EzD6B,uC;MAAC,kC;MAErC,oBAAkC,kB;MACiC,sBAAYB,C;K;2EAHY,Y;MAA
A,8B;K;2FAGrC,Y;MAAA,0B;K,OAAA,gB;MAAA,0B;K;gDAGA,sB;MACI,eAAe,aAAS,qBAAY,GAAZ,C;MA
CxB,mBAAmB,6BAAsB,QAAtB,C;MACnB,IAAI,oBAAJ,C;QAEI,kBAAW,QAAX,IAAuB,mCAAY,GAAZ,EA
AiB,KAAjB,C;QAEvB,IAAI,6BAAJ,C;UAEI,YAA+B,Y;UAC/B,IAAI,aAAS,gBAAO,KAAM,IAAb,EAakB,GA
AIB,CAAb,C;YACI,OAAO,KAAM,gBAAS,KAAT,C;YAEb,kBAAW,QAAX,IAAuB,CAAQ,KAAR,EAae,mCA
AY,GAAZ,EAAiB,KAAjB,CAAf,C;YACvB,6B;YACA,OAAO,I;UAIK,YAAuB,C;YACvC,cAAkB,wBAAN,KA
AM,EAAiB,GAAjB,C;UACIB,IAAI,eAAJ,C;YACI,OAAO,OAAM,gBAAS,KAAT,C;UAEX,KAAy,MAAK,mC
AAy,GAAZ,EAAiB,KAAjB,CAAL,C;MAG1B,6B;MAEA,OAAO,I;K;iDAGX,e;MAEuB,Q;MADnB,eAAe,aAA
S,qBAAY,GAAZ,C;MACL,oCAAsB,QAAtB,C;MAAA,iB;QAAMC,OAAO,I;MAA7D,mBAAmB,I;MACnB,IAA
I,6BAAJ,C;QACI,YAAgC,Y;QACChC,IAAI,aAAS,gBAAO,KAAM,IAAb,EAakB,GAAIB,CAAb,C;U5BzDR,O4B
0D6B,iB5B1DvB,C4B0DmC,Q5B1DnC,C;U4B2DM,6B;UACA,OAAO,KAAM,M;UAEb,OAAO,I;QAGX,YAA
uB,C;YACvC,8BAAc,KAAAd,iB;UACI,cAAy,MAAM,KAAN,C;UACZ,IAAI,aAAS,gBAAO,GAAP,EAAY,OAA
M,IAAIB,CAAb,C;YACI,IAAI,KAAM,OAAN,KAAc,CAAIB,C;cACU,KAAN,UAA2B,C;c5BtE/C,O4BwEqC,iB5
BxE/B,C4BwE2C,Q5BxE3C,C;c4B2EoB,KAAy,QAAM,KAAP,EAAC,CAAd,C;YAEtB,6B;YAEA,OAAO,OAA
M,M;MAIzB,OAAO,I;K;0CAGX,Y;MACI,oBAAa,kB;MACb,YAAO,C;K;mDAGX,e;MAAyC,uBAAS,GAAT,
S;K;8CAEzC,e;MAA+B,Q;MAAA,+BAAS,GAAT,8B;K;+CAE/B,e;MACuB,Q;MAAA,oCAAsB,aAAS,qBAAY,
GAAZ,CAA/B,C;MAAA,iB;QAAoD,OAAO,I;MAA9E,mBAAmB,I;MACnB,IAAI,6BAAJ,C;QACI,YAAgC,Y;Q
AChC,IAAI,aAAS,gBAAO,KAAM,IAAb,EAakB,GAAIB,CAAb,C;UACI,OAAO,K;UAEP,OAAO,I;QAGX,YA
AuC,Y;QACvC,OAAa,wBAAN,KAAM,EAAiB,GAAjB,C;K;uDAlrB,0B;MACI,sB;Q7F+nCY,Q;QAaHb,iD;UA
AgB,cAAhB,e;UAAAsB,I6F/nCK,aAAS,gB7F+nCA,O6F/nCa,IAAb,M7F+nCd,C;YAAwB,qBAAO,O;YAAP,uB;;
QAC9C,qBAAO,I;M6FhoCH,yB;K;IAIO,8E;MAAA,wD;MACH,aAAy,E;MAEZ,YAA0B,MAAa,MAAK,qCAA
L,C;MACvC,gBAAe,E;MAEf,oBAA4B,I;MAC5B,eAAc,K;MACd,iBAAgB,E;MAChB,iBAAqC,I;K;yEAErC,Y;
MACI,IAAI,6BAAwB,YAA5B,C;QACI,gBAAqB,iBAAqD,O;QAC1E,IAAI,4DAAC,SAaIB,C;UACI,OAAO,C;
MAGf,IAAI,yDAAa,SAAK,OAAtB,C;QACI,oBAAe,2CAAW,UAAK,aAAL,CAAX,C;QACf,eAAU,iC;QACV,iB
AAy,C;QACZ,OAAO,C;QAEp,oBAAe,I;QACf,OAAO,C;K;mEAI,f;Y;MACI,IAAI,eAAS,EAAb,C;QACI,aAAQ,
oB;MACZ,OAAO,eAAS,C;K;gEAGpB,Y;MAEOB,Q;MADhB,IAAI,CAAC,cAAL,C;QAAGB,MAAM,6B;MACN,
IAAI,YAAJ,C;QACZ,yBAAqD,cAArD,C;QAEa,OAAb,iB;MAHJ,oB;MAKA,iBAaIB,S;MACjB,aAAQ,E;MAC
R,OAAO,S;K;kEAGX,Y;M/E/CR,I+EgDyB,c/EhDrB,QAAl,C;QACI,cAhByB,0B;QAIzBzB,MAAM,6BAAsB,OA
AQ,WAA9B,C;M+E+CE,6BAAyB,cAAO,6BAAy,IAAnB,C;MACzB,iBAAy,I;MAEZ,uC;K;6CAtdZ,Y;MAEI,
2D;K;4DAyDJ,oB;MACI,mBAAmB,kBAAW,QAAX,C;MACnB,OAAW,iBAaIB,SAArB,GAAGC,IAAhC,GAA0
C,Y;K;wCCtKrD,Y;MACI,aAAR,MAAM,OAAe,CAAP,IAAO,C;MAEb,OAAO,KAAP,IAAGB,C;M7BXpB,O6
BYqB,M7BZf,C6BYuB,K7BZvB,C;M6BaF,OAAO,M;K;ICNuB,qC;MAAC,kC;MAEnC,oBAAkC,kB;MACiC,sB
AAyB,C;K;yEAHU,Y;MAAA,8B;K;yFAGnC,Y;MAAA,0B;K,OAAA,gB;MAAA,0B;K;iDAWA,e;MACI,IAAI,0
BAAJ,C;QAAoB,OAAO,K;MAC3B,OAAO,kBAAW,GAAX,MAAoB,S;K;4CAG/B,e;MACI,IAAI,0BAAJ,C;QA

AoB,OAAO,I;MAC3B,YAAY,kBAAW,GAAX,C;MACZ,OAAW,UAAU,SAArB,GAAGC,KAAhC,GAA2D,I;K;8
CAI/D,sB;MjFVA,IAAI,EiFWQ,uBjFXR,CAAJ,C;QACI,cAda,qB;QAeb,MAAM,gCAAYB,OAAQ,WAAjC,C;;Mi
FUN,eAAe,kBAAW,GAAX,C;MACf,kBAAW,GAAX,IAAkB,K;MAEIB,IAAI,aAAa,SAAjB,C;QACI,6B;QAEA,
OAAO,I;;QAGP,OAAO,Q;;K;+CAIf,e;MACI,IAAI,0BAAJ,C;QAAoB,OAAO,I;MAC3B,YAAY,kBAAW,GAAX,
C;MACZ,IAAI,UAAU,SAAd,C;Q9BnDJ,O8BoDyB,iB9BpDnB,C8BoD+B,G9BpD/B,C;Q8BqDE,6B;QAEA,OAA
O,K;;QAGP,OAAO,I;;K;wCAKf,Y;MACI,oBAAa,kB;MACb,YAAO,C;K;IAKA,0E;MAAA,oD;MACH,cAAkC,M
AAa,MAAK,mCAAL,C;MAC/C,kBAA4B,qBAAL,WAAK,C;MAC5B,iBAA+B,I;K;IEAE/B,Y;MAAkC,OAAA,e
AAS,U;K;8DAE3C,Y;MAIuB,gB;MAHnB,UAAU,eAAS,O;MACnB,iBAAU,G;MAES,+E;MAAnB,OAAO,iD;K;
gEAGX,Y;MAEK,UAA9B,M;MAAA,oC;MAA8B,YAAa,c;MjFchD,uB;MAeP,IAfoB,KAehB,QAAJ,C;QACI,cA
hByB,0B;QAIbZB,MAAM,6BAAsB,OAAQ,WAA9B,C;;QAEN,sBAnBgB,K;;MiFde,oBAAO,sFAAP,C;K;;2CAjB
nC,Y;MACI,yD;K;IAqBkD,0F;MAAA,8B;MAAA,oD;K;kHAC9B,Y;MAAQ,uB;K;oHACN,Y;MAAQ,6CAAuB,g
BAAvB,C;K;2EAE9B,oB;MAAwC,OAAA,2BAAuB,aAAI,gBAAJ,EAAS,QAAT,C;K;qEAE/D,Y;MAA+B,OAA
A,mCAAY,uBAAc,IAAd,C;K;qEAC3C,Y;MAAkC,OAAA,mCAAY,uBAAc,IAAd,C;K;mEAC9C,iB;MAA4C,OA
AA,mCAAY,qBAAY,IAAZ,EAakB,KAAIB,C;K;;gDAR5D,e;MAAsD,iE;K;;;MCItD,sBAOsC,I;MA6CtC,yB;MA
OA,4BAAkC,K;;IArIE,sD;MAZpC,oB;MAYyD,0CAAqC,GAArC,EAA0C,KAA1C,C;MACrD,oBAAuC,I;MACv
C,oBAAuC,I;K;wDAEvC,oB;MACI,WAAmB,iB;MACnB,OAAa,mEAAS,QAAT,C;K;;IAIrB,wC;MAAA,oB;MA
A+B,8C;K;IAE3B,sD;MAAA,oB;MACI,cACsC,I;MAEtC,cACsC,I;MAGIC,cAAO,iC;K;6DAIX,Y;MACI,OAAO,
gBAAS,I;K;0DAGpB,Y;MAEI,IAAI,CAAC,cAAL,C;QAAGB,MAAM,6B;MAEtB,cAAc,0B;MACd,cAAO,O;MA
Ca,gBAAb,OAAQ,a;;MAAf,c/E0DS,S+E1DoB,KAAO,iC/E0DzC,GAAGqB,SAArB,GAA+B,I;M+EzD1B,OAAO,O
;K;4DAGX,Y;MIFwBR,IAAI,EkFvBc,eAAQ,IIFuBtB,CAAJ,C;QACI,cAdW,e;QAeX,MAAM,6BAAsB,OAAQ,W
AA9B,C;;MkFxBE,WAAc,iB;MAGP,oCAAP,0BAAO,C;MACP,gCAAI,cAAO,0BAAO,IAAd,C;MAEJ,cAAO,I;K
;;iDAIf,mB;MAAyD,MAAM,qCAA8B,iCAA9B,C;K;6CAC/D,Y;MACI,WAAmB,Q;K;6DAGvB,mB;MAAgE,OA
AA,WAAmB,uBAAc,OAAd,C;K;gDAEnF,Y;MAAwE,qD;K;2DAExE,mB;MACI,qB;MACA,IAAI,iBAAS,QAAT
,CAAJ,C;QACI,WAAmB,cAAO,OAAQ,IAAf,C;QACnB,OAAO,I;;MAEX,OAAO,K;K;8FAGY,Y;MAAQ,OAAA
,WAAmB,K;K;sDAEID,Y;MAAsC,WAAmB,iB;K;;iDAa7D,qB;MIFrBA,IAAI,EkF0BM,0BAAQ,IAAR,IAAGB,0
BAAQ,IIF1B9B,CAAJ,C;QACI,cAdW,e;QAeX,MAAM,6BAAsB,OAAQ,WAA9B,C;;MkF0BN,YAAY,mB;MAC
Z,IAAI,SAAS,IAAb,C;QACI,sBAAO,S;QACP,yBAAO,S;QACP,yBAAO,S;;QAGK,YAAa,KAAM,a;QIFIBhC,uB
;QAeP,IAfoB,KAehB,QAAJ,C;UACI,gBAhByB,0B;UAIbZB,MAAM,6BAAsB,SAAQ,WAA9B,C;;UAEN,sBAnB
gB,K;;QkFkBZ,+B;QAEA,yBAAO,K;QACP,yBAAO,K;QAEP,qBAAa,S;QACb,qBAAa,S;;K;+CAIrB,qB;MAII,I
AAI,SAAK,aAAL,KAAc,SAAlB,C;QAEI,sBAAO,I;;QAEP,IAAI,wBAAS,SAAb,C;UAEL,sBAAO,sB;;QAEX,qD
AAc,sB;QACd,qDAAc,sB;;MAEIB,yBAAO,I;MACP,yBAAO,I;K;oCA8CX,Y;MAEI,qB;MACA,4BAaA,I;MACb,
OAAO,I;K;oCAGX,Y;MACI,qB;MACA,kBAAI,Q;MACJ,sBAAO,I;K;gDASX,e;MAAmD,OAAA,kBAAI,mBAA
Y,GAAZ,C;K;kDAEvD,iB;MACiC,Q;MAAA,0B;MAAA,iB;QAAQ,OAAO,K;;MAA5C,WAA6B,I;;QAEzB,IAAI,
OAAA,IAAK,MAAL,EAAC,KAAAd,CAAJ,C;UACI,OAAO,I;;QAEX,OAAO,cAAA,IAAK,aAAL,C;;MACF,iBAA
S,mBAAT,C;MACT,OAAO,K;K;6CAIX,Y;MAAoF,uC;K;wCAEPf,e;MAAmD,Q;MAAJ,QAAI,OAAJ,kBAAI,W
AAI,GAAJ,CAAJ,6B;K;0CAE/C,sB;MACI,qB;MAEA,UAAU,kBAAI,WAAI,GAAJ,C;MACd,IAAI,OAAO,IAAX
,C;QACI,eAAe,mCAAW,GAAX,EAAGB,KAAhB,C;QACf,kBAAI,aAAI,GAAJ,EAAS,QAAT,C;QACK,wBAAT,
QAAS,C;QACT,OAAO,I;;QAEP,OAAO,GAAL,gBAAS,KAAT,C;;K;2CAInB,e;MACI,qB;MAEA,YAAY,kBAAI,
cAAO,GAAP,C;MACHb,IAAI,SAAS,IAAb,C;QACU,sBAAN,KAAM,C;QACN,OAAO,KAAM,M;;MAEjB,OAA
O,I;K;qFAGmB,Y;MAAQ,OAAA,kBAAI,K;K;6CAE1C,Y;MACI,IAAI,yBAAJ,C;QAAGB,MAAM,oC;K;;IANg1
B,mC;MAAA,uD;MAGuB,qB;MA9J3B,yB;MA+JQ,sBAAM,gB;MAJV,Y;K;IAOA,iD;MAAA,uD;MAAoD,qB;M
AIKxD,yB;MAoKc,Q;MAAN,sBAAM,+D;MAFV,Y;K;IAKA,kE;MAAA,uD;MAQ8D,eAAM,eAAN,EAauB,UA
AvB,Q;MA/KIE,yB;MAGLQ,sBAAM,gB;MATV,Y;K;IA YA,sD;MAAA,uD;MAA2C,qBAAK,eAAL,EAASB,GA
AtB,Q;MAA3C,Y;K;IAEA,+C;MAAA,uD;MAG2C,qB;MAxL/C,yB;MAyLQ,sBAAM,gB;MACN,KAAK,gBAAO
,QAAP,C;MALT,Y;K;IA6EJ,kC;MAKwD,gBAA7C,qBAAYB,eAAzB,C;MAAqD,wB;MAA5D,O/EjMO,S;K;;oC
gFvCP,Y;MAEK,Q;MAA8B,CAA9B,2EAA8B,S;MAC/B,OAAO,I;K;6CAGX,Y;MAA+C,gBAAI,iB;K;;IAhCnD,
wC;MAAA,uD;MAAmD,eAAM,GAAN,Q;MAPvD,yB;MAOI,Y;K;IAEA,qC;MAAA,uD;MAGuB,eAAM,oBAAN
,Q;MAZ3B,yB;MASI,Y;K;IAKA,+C;MAAA,uD;MAG8C,eAAM,oBAAN,Q;MAjBID,yB;MAkBQ,qBAAO,QAA

P,C;MAJJ,Y;K;IAOA,kE;MAAA,uD;MAQ8D,eAAM,qBAAsB,eAAtB,EAAuC,UAAvC,CAAN,Q;MA7BIE,yB;M
AqBI,Y;K;IAUA,sD;MAAA,uD;MAA2C,qBAAK,eAAL,EAAsB,GAAtB,Q;MAA3C,Y;K;IAGBJ,qC;MAKMD,gB
AAxC,mBAAC,qBAAd,C;MAAgD,6B;MAAvD,OhFoBO,S;K;;;kFiFzEX,uB;MAQI,OAAO,O;K;ICXX,sB;K;mC
ACI,Y;MACI,mBAAM,IAAN,C;K;2CAGJ,mB;MACI,mBAAM,OAAN,C;MACA,c;K;iCAKJ,Y;K;;IAKuB,oC;M
AA8B,qB;MAA7B,gC;K;2CACxB,mB;MAEI,oBA+DyC,OA/Dd,OA+Dc,C;MA9DzC,iBAAa,OAAM,aAAN,C;K;
;IAIrB,8B;MAEoC,qB;K;iDACHC,mB;MACI,OAAQ,KAAI,OAAJ,C;K;mDAGZ,mB;MACI,OAAQ,KAAI,OAAJ,
C;K;2CAGZ,Y;MACI,OAAQ,KAAI,EAAJ,C;K;;IAIhB,0B;MAEqC,qB;MACjC,cAAa,E;K;6CAEb,mB;MACI,eA
oCyC,OApCxB,OAoCwB,C;K;qCAjC7C,Y;MACI,cAAS,E;K;;IAIjB,sC;MAE4C,yB;K;yDACxC,mB;MACI,QAw
ByC,OAxB1B,OAwb0B,C;MAvBzC,QAAQ,CxEqJoF,awErJhE,IxEqJgE,EwErJ1D,CxEqJ0D,C;MwEpJ5F,IAAI,
KAAK,CAAT,C;QACI,4BAAU,CxE+J0E,WwE/J9D,CxE+J8D,EwE/J3D,CxE+J2D,C;QwE9JpF,Y;QACA,IAAI,C
xE0JiE,WwE1JrD,IAAI,CAAJ,IxE0JqD,C;;MwExJzE,4BAAU,C;K;iDAGd,Y;MACI,OAAQ,KAAI,WAAJ,C;MA
CR,cAAS,E;K;;IAWjB,yB;MACiD,cAAa,KAAb,C;K;IAEjD,mB;MAEI,MAAO,U;K;IAGX,4B;MAEI,MAAO,iB
AAQ,OAAR,C;K;IAGX,wB;MAEI,MAAO,eAAM,OAAN,C;K;IAGX,kB;MACqC,MAAM,qCAA8B,sCAA9B,C;
K;IAE3C,wB;MAC4C,MAAM,qCAA8B,4CAA9B,C;K;IClGid,mD;MACI,0B;MASA,gBAA2B,a;K;2FAFvB,Y;M
AAQ,OAAA,eAAS,Q;K;oDAIrB,kB;MACI,UAAU,IAAK,S;MAEX,YAAQ,2CAAR,C;QACI,gBAAC,MAAO,M;
WAEzB,YAAQ,yBAAR,C;QACI,gBAAC,yC;QACd,eAAS,oBAAW,MAAX,C;;QAEI,MAAM,6BAAsB,iBAAtB,
C;K;4CAItB,Y;MAOW,Q;MALP,IAAI,kBAAW,2CAAF,C;QACI,gBAAS,yB;QACT,OAAO,yB;;MAEX,aAAa,IA
AK,S;MAEd,eAAW,yCAAX,C;QAAsB,gC;WACtB,0C;QAA4B,MAAM,MAAO,U;;QACjC,a;MAHZ,W;K;;IA7B
J,gD;MAAA,0D;MACyD,6BAAK,QAAL,EAAe,2CAAF,C;MADzD,Y;K;;;;ICRA,2C;MAAA,+D;MAAuB,iC;MA
F3B,iC;MAEI,Y;K;IACA,sD;MAAA,+D;MAAuC,6BAAM,OAAN,Q;MAH3C,iC;MAGI,Y;K;IACA,6D;MAAA,+
D;MAAmD,kCAAM,OAAN,EAAe,KAAf,C;MAJvD,iC;MAII,Y;K;IACA,oD;MAAA,+D;MAAiC,6BAAM,KAA
N,Q;MALrC,iC;MAKI,Y;K;IxC4CJ,yE;MASI,sC;MAAA,4C;K;IATJ,iGAWY,Y;MAAQ,2B;KAXpB,E;IAAA,0D
AaQ,kB;MACI,wBAAW,MAAX,C;K;IAdZ,sF;IyC5C2E,0C;M1CkKhE,Q;MADP,e0ChKA,M1CgKA,C;MACO,Q
0CjKP,M1CiKO,+D;M0ChKX,W;K;;+FCuHA,gB;MACI,aAAa,IAAb,MAAa,E;MACb,KAAK,MAAL,C;MACA,
OAAO,M;K;wFC3HX,yB;MAAA,uD;MAAA,wC;QAWqG,OAAK,cAAL,SAAK,EAAiB,IAAjB,EAAuB,IAAvB,
C;O;KAX1G,C;wFAaA,yB;MAAA,uD;MAAA,wC;QAWoG,OAAK,cAAL,SAAK,EAAiB,IAAjB,EAAuB,IAAvB,
C;O;KAXzG,C;8ECbA,yB;MAAA,6C;MAAA,sC;QAOyD,OAAK,SAAL,SAAK,EAAy,QAaz,C;O;KAP9D,C;8E
ASA,yB;MAAA,6C;MAAA,wC;QAWkE,OAAK,SAAL,SAAK,EAAa,UAAb,S;O;KAXvE,C;OFaAa,yB;MAAA,
mD;MAAA,wC;QAWqE,OAAK,YAAL,SAAK,EAAgB,UAAhB,S;O;KAX1E,C;kFCZI,yB;MAAA,iD;MAAA,4B;
QAae,OAAK,WAAL,SAAK,C;O;KAApB,C;wFAYA,yB;MAAA,uD;MAAA,4B;QAae,OAAK,cAAL,SAAK,C;O
;KAApB,C;IC5BJ,gC;MAAoE,gCAAqB,OAAR,B;K;IAEiC,uC;MAAC,wB;K;iDAC/B,iB;MACI,eAAQ,KAAr,C
;K;8CAGJ,Y;MAAyC,iCAAuB,cAAvB,M;K;;ICCO,6C;MAAA,8B;MAAS,uB;K;8FACIC,Y;MAAQ,OAAA,gBA
AY,O;K;mDAE3C,iB;MACI,IADoC,KACpC,IAAG,CAAH,IADoC,KACpC,IAAM,sBAAN,C;QAD8B,OACX,gB
AAy,MAAK,KAAL,C;;QACvB,MAAM,8BAA0B,WAAQ,KAAr,6BAAMc,sBAAnC,MAA1B,C;K;;IARtB,8B;
MAGoD,4C;K;wECFpD,yB;MAAA,uC;MAAA,4B;QAOsC,MAAL,SAAK,C;O;KAPtC,c;kFASA,yB;MAAA,iD;
MAAA,kC;QAWuD,OAAK,WAAL,SAAK,EAAc,IAAd,C;O;KAX5D,C;+ECfA,qB;MAI8C,gB;K;iFAE9C,qB;M
AIsE,OAAK,S;K;kFAE3E,qB;MAMyE,gB;K;IAEzE,6B;MAiBa,UAPF,M;MAFP,QAAC,S;MAGV,cAAK,UAAAL,
U;QACI,mBAAK,UAAAL,G;;QACJ,I/CzBqC,MAAa,Y+CyBvC,C/CzBuC,C+CyBID,C;UAC6B,8BAAzB,CAAyB,
C;;UAGN,UAAIB,uDAakB,Y;;MAP3B,a;K;IC9BJ,2B;MAEI,MAAM,yBAAqB,OAAR,B;K;IAGV,sB;MAEI,M
AAM,uBAAMb,cAAnB,C;K;IAGV,2B;MAEI,MAAM,6BAAsB,OAAtB,C;K;IAGV,iC;MAEI,MAAM,4CAAqC,u
BAAqB,YAArB,8BAAR,C;K;ICIBV,8B;MC8CW,kB1GqBiD,oB;M0GM9C,Q;MAAA,OAAK,0B;MAAf,OAAU,
cAAV,C;QAAU,mB;QACN,UAAU,sBAAM,CAAN,C;QACV,kBAakB,sBAAY,GAaz,C;QAKfID,U;QAJFnE,W
1GuKJ,a0GvKgB,G1GuKhB,EyG1OoB,CCmEkC,uBAAuB,CAAC,WAAY,mBAAY,GAaz,CAiFhD,GDpJrC,CC
oJqC,GAA6B,UAJfJC,WaIFiC,6DDpJnD,IAAM,CAAN,IzG0OpB,C;;MyG1OA,OCqEO,W;K;IC3EqC,gD;MAAC
,oC;K;;;;IC0CjD,qB;MAK0B,Q;MADtB,UAAmB,E;MACnB,wBAAsB,KAAtB,gB;QAAsB,aAAA,KAAtB,M;QA
AK,IAAC,0BAAD,EAAO,2B;QACR,IAAI,IAAJ,IAAY,K;;MAEhB,OAAO,G;K;IAGX,+B;MAMgB,Q;MADZ,W
AA0B,MAAa,MAAK,KAAL,C;MACvC,wBAAY,IAAZ,gB;QAAY,UAAA,IAAZ,M;QACI,IAAU,KAAy,gBAae,
GAaf,CAAtB,C;UACI,UAAK,GAAL,IAAY,MAAM,GAAN,C;;MAGpB,OAAO,S;K;qEC5DX,yB;MAAA,iB;M

AAA,oB;QAOkD,OAAA,MAAW,KAAI,CAAJ,C;O;KAP7D,C;qEASA,yB;MAAA,iB;MAAA,oB;QAOkD,OAAA,MAAW,KAAI,CAAJ,C;O;KAP7D,C;qEASA,yB;MAAA,iB;MAAA,oB;QAOkD,OAAA,MAAW,KAAI,CAAJ,C;O;KAP7D,C;uEASA,yB;MAAA,iB;MAAA,oB;QASmD,OAAA,MAAW,MAAK,CAAL,C;O;KAT9D,C;uEAWA,yB;MAAA,iB;MAAA,oB;QASmD,OAAA,MAAW,MAAK,CAAL,C;O;KAT9D,C;uEAWA,yB;MAAA,iB;MAAA,oB;QASmD,OAAA,MAAW,MAAK,CAAL,C;O;KAT9D,C;yEAWA,yB;MAAA,iB;MAAA,uB;QAKB+D,OAAA,MAAW,OAAM,CAAN,EAAS,CAAT,C;O;KAlB1E,C;uEaObA,yB;MAAA,iB;MAAA,oB;QAUmD,OAAA,MAAW,MAAK,CAAL,C;O;KAV9D,C;uEAYA,yB;MAAA,iB;MAAA,oB;QASmD,OAAA,MAAW,MAAK,CAAL,C;O;KAT9D,C;uEAWA,yB;MAAA,iB;MAAA,oB;QAUmD,OAAA,MAAW,MAAK,CAAL,C;O;KAV9D,C;yEAYA,yB;MAAA,iB;MAAA,oB;QAYoD,OAAA,MAAW,OAAM,CAAN,C;O;KAZ/D,C;yEAcA,yB;MAAA,iB;MAAA,oB;QAYoD,OAAA,MAAW,OAAM,CAAN,C;O;KAZ/D,C;yEAcA,yB;MAAA,iB;MAAA,oB;QAoD,OAAA,MAAW,OAAM,CAAN,C;O;Kab/D,C;yEaEa,yB;MAAA,iB;MAAA,uB;QAS+D,OAAA,MAAW,OAAM,CAAN,EAAS,CAAT,C;O;KAT1E,C;uEAWA,yB;MAAA,iB;MAAA,oB;QAQmD,OAAA,MAAW,MAAK,CAAL,C;O;KAR9D,C;qEAUA,yB;MAAA,iB;MAAA,oB;QAUKD,OAAA,MAAW,KAAI,CAAJ,C;O;KAV7D,C;yEAYA,yB;MAAA,iB;MAAA,oB;QAcOD,OAAA,MAAW,OAAM,CAAN,C;O;KAD/D,C;IAgBA,sB;MAcI,IAAI,QAAQ,GAAR,IAAe,SAAQ,GAA3B,C;QAAgC,OAAO,wCAA0,I;MAC9C,OAAO,IAAW,KAAI,CAAJ,CAAX,GAAoB,IAAW,KAAI,IAAJ,C;K;mEAG1C,yB;MAAA,iB;MAAA,oB;QAWiD,OAAA,MAAW,KAAI,CAAJ,C;O;KAX5D,C;yEAaA,yB;MAAA,iB;MAAA,oB;QA0oD,OAAA,MAAW,OAAM,CAAN,C;O;KAP/D,C;uEASA,yB;MAAA,iB;MAAA,oB;QA0mD,OAAA,MAAW,MAAK,CAAL,C;O;KAP9D,C;uEASA,yB;MAAA,iB;MAAA,oB;QAgBmD,OAAA,MAAW,OAAM,CAAN,C;O;KAhB9D,C;uEakBA,yB;MAAA,iB;MAAA,oB;QAUmD,OAAA,MAAW,MAAK,CAAL,C;O;KAV9D,C;yEAYA,yB;MAAA,iB;MAAA,oB;QAUoD,OAAA,MAAW,OAAM,CAAN,C;O;KAV/D,C;+EAYA,yB;MAAA,iB;MAAA,oB;QAUuD,OAAA,MAAW,OAAM,CAAN,C;O;KAVIE,C;IAYA,kB;MAQI,IAAI,IAAI,GA AJ,KA AW,GA Af,C;QACI,OAAO,IAAW,OAAM,CAAN,C;;MAEtB,YAzBgD,MAAW,OAYzC,CazByC,C;MA0B3D,OAAW,QAAQ,CAAR,KAAa,GAAxB,GAA6B,KAA7B,GAtC+C,MAAW,MAcCb,CAtCa,C;K;qEAyC9D,yB;MAAA,iB;MAAA,oB;QAUKD,OAAA,MAAW,KAAI,CAAJ,C;O;KAV7D,C;uEAYA,yB;MAAA,iB;MAAA,oB;QAWmD,OAAA,MAAW,MAAK,CAAL,C;O;KAX9D,C;wEAcA,yB;MAAA,iB;MAAA,uB;QA06D,OAAA,MAAW,KAAI,CAAJ,EAAO,CAAP,C;O;KAPxE,C;wEASA,yB;MAAA,iB;MAAA,uB;QA06D,OAAA,MAAW,KAAI,CAAJ,EAAO,CAAP,C;O;KAPxE,C;qEAWA,yB;MAAA,iB;MAAA,+B;QAayD,OAAA,MAAW,KAAI,SAAJ,EAAU,CAAV,C;O;KAbpE,C;uEaEa,yB;MAAA,iB;MAAA,+B;QA0sD,OAAA,MAAW,KAAI,SAAJ,EAAy,CAAZ,C;O;KAPjE,C;iGAmBsD,yB;MAAA,iB;MAAA,4B;QAAQ,OAAA,MAAW,KAAI,SAAJ,C;O;KAA nB,C;+EAaT,yB;MAAA,iB;MAAA,4B;QAAQ,OAAA,MAAW,MAAK,SAAL,C;O;KAA nB,C;iFAE7C,yB;MAAA,6C;MAAA,kC;QAK8D,OAAK,SAAL,SAAK,EAAC,IAAd,C;O;KALnE,C;IAkBqC,4B;MACjC,gBAAO,CAAP,C;QADyC,OACrB,QAAP,CAAC,SAAM,C;WACpB,IAAK,QAAL,SAAK,CAAL,IAAgB,cAAQ,wCAA0,kBAA/B,C;QAFyC,OA EW,S;WACpD,kBAAQ,wCAA0,UAAf,C;QAHyC,OAGb,YAAY,SAAL,SAAK,C;;QAHc,OAI5B,OAAL,SAAK,CAAL,GAAgB,S;K;IAG5B,2B;MAKI,IAAK,QAAL,SAAK,CAAL,IAAgB,cAAQ,wCAA0,kBAA/B,C;QADwC,OACY,S;WACpD,kBAAQ,GAAR,C;QAFwC,OA EZB,wCAA0,U;;QACP,WAAc,UAAL,SAAK,CAAL,yBA AuB,YAAO,CAAX,GAAC,CAAd,GAAqB,EAAxC,E;QAHgB,OjDhb6B,MAAa,gBA Ae,IAAf,C;;K;IiDsbTf,6B;MAKI,IAAK,QAAL,SAAK,CAAL,IAAgB,cAAQ,wCAA0,kBAA/B,C;QAD0C,OACU,S;WACpD,kBAAQ,GAAR,C;QAF0C,OA E3B,CAAC,wCAA0,U;;QACR,WAAc,UAAL,SAAK,CAAL,yBA AuB,YAAO,CAAX,GAAC,EAAd,GAAsB,CAAzC,E;QAHkB,OjD1b2B,MAAa,gBA Ae,IAAf,C;;K;IiDictF,oC;MAUI,IAAK,QAAL,SAAK,CAAL,IAAmB,QA AH,EAAG,CAAnB,C;QADuD,OACzB,wCAA0,I;WACrC,WAAM,SAAN,C;QAFuD,OA EZC,E;WACd,SAAK,SAAL,C;QAHuD,OAGrC,OAAL,SAAK,C;;QAHqC,OAI1B,SAAL,SAAK,C;K;IAIjC,+B;MAYI,uB;QAAW,MAAM,gCAAYB,yBAAzB,C;WACjB,gBAAO,UAAP,C;QAFyC,OA EJB,U;WACxB,gBAAO,WAAP,C;QAHyC,OAGjB,W;;QAHiB,OAIv,YAAvB,IAAW,OAAM,SAAN,CAAY,C;K;IAGnC,gC;MAYI,uB;QAAW,MAAM,gCAAYB,yBAAzB,C;WACjB,oD;QAF2C,+B;WAG3C,oD;QAH2C,+B;;QAAA,OAIz,uBAAvB,IAAW,OAAM,SAAN,CAAY,C;K;uEASnC,yB;MAAA,iB;MAAA,oB;QA0gD,OAAA,MAA6B,KAAZ,CAAY,C;O;KAP7E,C;uEASA,yB;MAAA,iB;MAAA,oB;QA0gD,OAAA,MAA6B,KAAZ,CAAY,C;O;KAP7E,C;uEASA,yB;MAAA,iB;MAAA,oB;QA0gD,OAAA,MAA6B,KAAZ,CAAY,C;O;KAP7E,C;yEASA,yB;MAAA,iB;MAAA,oB;QASiD,OAAA,MAA8B,MAAZ,CAAY,C;O;KAT/E,C;yEAWA,yB;MAAA,iB;MAAA,oB;QASiD,OAAA,MAA8B,MAAZ,CAAY,C;O;

KAT/E,C;yEAWA,yB;MAAA,iB;MAAA,oB;QASiD,OAAA,MAA8B,MAAZ,CAAY,C;O;KAT/E,C;2EAWA,yB;MAAA,iB;MAAA,uB;QAKB4D,OAAA,MAA6C,OAA1B,CAA0B,EAAZ,CAAY,C;O;KAIbZg,C;yEAoBA,yB;MAAA,iB;MAAA,oB;QAUiD,OAAA,MAA8B,MAAZ,CAAY,C;O;KAV/E,C;yEAYA,yB;MAAA,iB;MAAA,oB;QASiD,OAAA,MAA8B,MAAZ,CAAY,C;O;KAT/E,C;yEAWA,yB;MAAA,iB;MAAA,oB;QAUiD,OAAA,MAA8B,MAAZ,CAAY,C;O;KAV/E,C;2EAYA,yB;MAAA,iB;MAAA,oB;QAYkD,OAAA,MAA+B,OAAZ,CAAY,C;O;KAZjF,C;2EAca,yB;MAAA,iB;MAAA,oB;QAYkD,OAAA,MAA+B,OAAZ,CAAY,C;O;KAZjF,C;2EAca,yB;MAAA,iB;MAAA,oB;QAakD,OAAA,MAA+B,OAAZ,CAAY,C;O;KAbjF,C;2EAeA,yB;MAAA,iB;MAAA,uB;QAS4D,OAAA,MAA6C,OAA1B,CAA0B,EAAZ,CAAY,C;O;KATzG,C;yEAWA,yB;MAAA,iB;MAAA,oB;QAQiD,OAAA,MAA8B,MAAZ,CAAY,C;O;KAR/E,C;uEAUA,yB;MAAA,iB;MAAA,oB;QAUgD,OAAA,MAA6B,KAAZ,CAAY,C;O;KAV7E,C;2EAYA,yB;MAAA,iB;MAAA,oB;QAcKd,OAAA,MAA+B,OAAZ,CAAY,C;O;KAdjF,C;uEAgBA,yB;MAAA,mC;MAAA,0B;QAc6D,OAAMC,IAA7B,CAA6B,EAAZ,IAAY,C;O;KadhG,C;qEAgBA,yB;MAAA,iB;MAAA,oB;QAW+C,OAAA,MAA6B,KAAZ,CAAY,C;O;KAX5E,C;2EAaA,yB;MAAA,iB;MAAA,oB;QAOkD,OAAA,MAA+B,OAAZ,CAAY,C;O;KAPjF,C;yEASA,yB;MAAA,iB;MAAA,oB;QAOiD,OAAA,MAA8B,MAAZ,CAAY,C;O;KAP/E,C;yEASA,yB;MAAA,iB;MAAA,oB;QAgBiD,OAAA,MAA+B,OAAZ,CAAY,C;O;KAhBhF,C;yEakBA,yB;MAAA,iB;MAAA,oB;QAUiD,OAAA,MAA8B,MAAZ,CAAY,C;O;KAV/E,C;2EAYA,yB;MAAA,iB;MAAA,oB;QAUkD,OAAA,MAA+B,OAAZ,CAAY,C;O;KAVjF,C;iFAYA,yB;MA3gBA,iB;MA2gBA,oB;QAUqD,OA3gBE,MAAW,OA2gBF,CA3gBE,C;O;KAigBIE,C;2EAYA,yB;MAAA,uC;MAAA,oB;QAQkD,OAaOB,MAAZ,CAAY,C;O;KARtE,C;uEAWA,yB;MAAA,iB;MAAA,oB;QAUgD,OAAA,MAA6B,KAAZ,CAAY,C;O;KAV7E,C;yEAYA,yB;MAAA,iB;MAAA,oB;QAWiD,OAAA,MAA8B,MAAZ,CAAY,C;O;KAX/E,C;wEAeA,yB;MAAA,iB;MAAA,uB;QAO0D,OAAA,MAAW,KAAI,CAAJ,EAAO,CAAP,C;O;KAPrE,C;wEASA,yB;MAAA,iB;MAAA,uB;QAO0D,OAAA,MAAW,KAAI,CAAJ,EAAO,CAAP,C;O;KAPrE,C;SEAYA,yB;MAAA,iB;MAAA,+B;QAASD,OAAA,MAA8C,KAA1B,SAA0B,EAAZ,CAAY,C;O;KAbpG,C;uEAeA,yB;MAAA,iB;MAAA,+B;QAOoD,OAAA,MAA8C,KAA1B,SAA0B,EAAZ,CAAY,C;O;KAPiG,C;kGAmBoD,yB;MAAA,iB;MAAA,4B;QAAQ,OAAA,MAAgC,KAAZ,SAAY,C;O;KAAxC,C;gFAaT,yB;MAAA,iB;MAAA,4B;QAAQ,OAAA,MAAiC,MAAZ,SAAY,C;O;KAAzC,C;gFAE3C,yB;MAAA,6C;MAAA,kC;QAO8D,OAA0C,SAArC,SAaQc,EAAZ,IAAY,C;O;KAPxG,C;iFASA,yB;MAAA,6C;MAAA,kC;QAK4D,OAA0C,SAArC,SAaQc,EAAZ,IAAY,C;O;KALtG,C;oFAQA,yB;MAAA,iD;MAAA,4B;QAYmD,OAAW,WAAX,SAAW,C;O;KAZ9D,C;sFAca,yB;MAAA,mD;MAAA,4B;QAYqD,OAAW,YAAX,SAAW,C;O;KAZhE,C;IAoBA,kB;MAUqC,OAAI,IAAI,CAAR,GAAY,CAAC,CAAD,OAAAM,CAAIB,GAA0B,C;K;wEAE/D,yB;MAAA,iB;MAAA,uB;QAKoD,OAAA,MAAW,KAAI,CAAJ,EAAO,CAAP,C;O;KAL/D,C;wEAOA,yB;MAAA,iB;MAAA,uB;QAKoD,OAAA,MAAW,KAAI,CAAJ,EAAO,CAAP,C;O;KAL/D,C;mGAIbGd,yB;MAAA,mC;MAAA,4B;QAAQ,WAAL,SAAJ,C;O;KAAR,C;IAShB,+B;MAC5B,gBAAO,CAAP,C;QADoC,OACxB,E;WACZ,gBAAO,CAAP,C;QAFoC,OAEzB,C;QAFwB,OAG5B,C;K;IAKZ,kB;MASuC,OAAI,eAAI,CAAR,GAAY,CAAD,aAAX,GAAMB,C;K;wEAE1D,gB;MAKuD,OAAI,kBAAK,CAAL,MAAJ,GAAY,CAAZ,GAAMB,C;K;wEAE1E,gB;MAKuD,OAAI,kBAAK,CAAL,MAAJ,GAAY,CAAZ,GAAMB,C;K;mGAYxB,yB;MAAA,mC;MAAA,4B;QAAQ,WAAL,SAAJ,C;O;KAAR,C;IASjB,+B;MAC7B,2BAAO,CAAP,C;QADqC,OACzB,E;WACZ,2BAAO,CAAP,C;QAFqC,OAEzB,C;QAFyB,OAG7B,C;K;IC1kCZ,4B;MAI4C,qBAAQ,S;K;IAEpD,4B;MAI2C,qBAAQ,S;K;IAEnD,+B;MAGiD,qBAAQ,wCAAO,kBAAf,IAAoC,cAAQ,wCAAO,kB;K;IAEpG,iC;MAGgD,qBAAQ,uCAAM,kBAAd,IAAmC,cAAQ,uCAAM,kB;K;IAEjG,6B;MAG+C,QAAC,qBAAD,IAAiB,CAAC,kB;K;IAEjE,+B;MAG8C,QAAC,uBAAD,IAAiB,CAAC,kB;K;IAGhE,iC;MAOI,QAAQ,S;MACR,IAAI,CAAC,IAAM,UAAP,KAAASB,CAAE,KAAK,CAAP,GAAC,UAApC,K;MACJ,IAAI,CAAC,IAAM,SAAP,KAAASB,CAAE,KAAK,CAAP,GAAC,SAAPC,K;MACJ,IAAI,CAAC,IAAM,QAAP,KAAASB,CAAE,KAAK,CAAP,GAAC,QAAPC,K;MACJ,IAAI,CAAC,IAAM,KAAP,KAAASB,CAAE,KAAK,EAA7B,K;MACJ,OAAO,C;K;kGAGX,yB;MAAA,iB;MAAA,4B;QAM2D,OAAA,MAAO,OAAM,SAAN,C;O;KANIE,C;IAQA,0C;MAOI,YATuD,MAAO,OAS9B,EAAf,aAAQ,CAAC,SAAD,IAARR,CAAe,CAT8B,CAS9D,I;K;IAEJ,sC;MAOI,OAAI,cAAQ,CAAZ,GA Ae,CAAf,GAASB,CAAE,IAAI,EAAJ,GAIB,+B,MAAO,iB;K;IAoBIE,qC;MAQI,oBAAS,CAAC,SAAD,IAAT,C;K;IAEJ,yC;MAaI,oBAAI,QA AJ,GAaiB,cAAK,EAAAL,GAaQB,Q;K;IAG1C,0C;MAaI,oBAAI,EAAJ,GAaOB,QAAPB,GAaiC,cAAK,Q;K;IAG1C,mC;MAMI,OAAK,alDhEmD,uBkDgEnD,CAAL,GAA0B,alDjE6B,sBkDiE7B,CAA1B,I;K;IAEJ,2C;MAMU,WA AW,SIDxEuC,

B;K;8CAHJ,Y;MAII,qB;K;gDAJJ,kD;MAAA,8BACI,kCADJ,EAEI,uDAFJ,EAGI,8CAHJ,EAIL,iDAJJ,C;K;4CAA
A,Y;MAAA,c;MACI,qD;MACA,4D;MACA,yD;MACA,0D;MAJJ,a;K;0CAAA,iB;MAAA,4IACI,oCADJ,IAEI,kD
AFJ,IAGI,4CAHJ,IAII,8CAJJ,I;K;ICAA,4B;MAAA,gC;MAEI,gBACe,wBAAoB,MAApB,EAA6D,KAA7D,EAAo
E,gCAApE,C;MAEf,mBACKB,wBAAoB,MAApB,EAAGe,QAaHe,EAA0E,mCAA1E,C;MAEIB,oBACmB,+B;M
AEnB,oBACmB,wBAAoB,OAApB,EAaKe,SAAIE,EAA6E,oCAA7E,C;MAEnB,iBACgB,wBAAoB,MAApB,EA
A8D,MAA9D,EAAsE,iCAAtE,C;MAEHb,kBACiB,wBAAoB,MAApB,EAA+D,OAA/D,EAAwE,kCAAxE,C;MA
EjB,gBACe,wBAAoB,MAApB,EAA6D,KAA7D,EAAoE,gCAApE,C;MAEf,kBACiB,wBAAoB,MAApB,EAA+D,
OAA/D,EAAwE,kCAAxE,C;MAEjB,mBACKB,wBAAoB,MAApB,EAAGe,QAaHe,EAA0E,mCAA1E,C;MAEIB,
kBACiB,wBAAoB,KAApB,EAAIE,OAAjE,EAA0E,kCAA1E,C;MAEjB,mBACKB,wBAAoB,MAApB,EAAGe,Q
AAhE,EAA0E,mCAA1E,C;MAEIB,sBACqB,wBAAoB,KAApB,EAaKe,WAAIE,EAA+E,sCAA/E,C;MAErB,yB
ACwB,wBAAoB,KAApB,EAAqE,cAArE,EAAqF,yCAArF,C;MAExB,sBACqB,wBAAoB,WAApB,EAAwE,WA
AxE,EAAqF,sCAArF,C;MAErB,sBACqB,wBAAoB,SAApB,EAAsE,WAAIE,EAAmF,sCAAnF,C;MAErB,uBACs
B,wBAAoB,UAApB,EAAwE,YAAxE,EAAsF,uCAAtF,C;MAEtB,qBACoB,wBAAoB,UAApB,EAAsE,UAAIE,EA
AAkF,qCAAlF,C;MAEpB,sBACqB,wBAAoB,KAApB,EAaKe,WAAIE,EAA+E,sCAA/E,C;MAErB,uBACsB,wB
AAoB,YAApB,EAA0E,YAA1E,EAAwF,uCAAXF,C;MAEtB,wBACuB,wBAAoB,YAApB,EAA2E,aAA3E,EAA0
F,wCAA1F,C;K;IAMkB,qE;MAAA,qB;QAAE,OvE/DD,OuE+DU,EAAT,KAAiB,UAAjB,IAAkC,EAAY,OAAf,K
AA0B,a;O;K;+CAJpG,iB;MAE2B,Q;MAAhB,U;MAAA,KAAgB,OAAhB,eAAGB,CAAI,KAAJ,CAAhB,U;QAAA
,a;QACH,aAAa,wBAAoB,QAApB,EAA+D,kBAA/D,EACoB,mDADpB,C;QAEg,eAAhB,UAAqC,M;QAHIC,SA
IH,M;MAJJ,a;K;IA7D+E,8C;MAAE,6B;K;IAGO,iD;MAAE,0B;K;IAME,kD;MAAE,8B;K;IAGZ,+C;MAAE,6B;
K;IAGC,gD;MAAE,6B;K;IAGR,8C;MAAE,6B;K;IAGI,gD;MAAE,6B;K;IAGC,iD;MAAE,6B;K;IAGH,gD;MAA
E,yB;K;IAGD,iD;MAAE,6B;K;IAGM,oD;MAAE,mC;K;IAGO,uD;MAAE,gC;K;IAGL,oD;MAAE,6B;K;IAGJ,oD
;MAAE,6B;K;IAGE,qD;MAAE,8B;K;IAGR,mD;MAAE,4B;K;IAGJ,oD;MAAE,6B;K;IAGQ,qD;MAAE,8B;K;IA
GC,sD;MAAE,+B;K;IA5DvH,wC;MAAA,uC;QAAA,sB;MAAA,gC;K;ICCA,2B;MAEW,Q;MAAA,IAAI,KAA
Y,SAAQ,MAAR,CAAhB,C;QACH,kBAAW,MAAX,C;QAEA,kBAAW,MAAX,C;MAHJ,W;K;IAOJ,8B;MAC4
E,QAAM,QAAS,OAAf,C;aACxE,C;UADwE,OACnE,WAAW,SAAS,CAAT,CAAX,C;aACL,C;UAFwE,OAEnE,+
B;UAFmE,OAGhE,iB;K;IAGZ,oC;MAEU,IAAN,I;MAAA,QxehB0C,OwEgB3B,CAAf,C;aACI,Q;UAA6B,OAA
jB,8BAAiB,Y;UAA7B,K;aACA,Q;UAAy,OAAl,CAAY,C/DbhC,G+DamC,CAAf,MAAkC,CAAtC,GAAyC,8BA
AiB,SAA1D,GAAwE,8BAAiB,Y;UAArG,K;aACA,S;UAA8B,OAAjB,8BAAiB,a;UAA9B,K;aACA,U;UAA+B,O
AAjB,8BAAiB,eAAGB,CAAY,OAA5B,C;UAA/B,K;UAGQ,6B;YAAAsC,OAAjB,8BAAiB,kB;eACtC,0B;YAAmC
,OAAjB,8BAAiB,e;eACnC,0B;YAAmC,OAAjB,8BAAiB,e;eACnC,2B;YAAoC,OAAjB,8BAAiB,gB;eACpC,yB;
YAAkC,OAAjB,8BAAiB,c;eACIC,0B;YAAmC,OAAjB,8BAAiB,e;eACnC,2B;YAAoC,OAAjB,8BAAiB,gB;eAC
pC,4B;YAAqC,OAAjB,8BAAiB,iB;eACrC,6B;eACA,sB;YAAkC,OAAjB,8BAAiB,W;YAE9B,kBAAkB,MAAA,
gBAAE,CAAf,CAAkB,Y;YAE7C,oBAAGB,MAAhB,C;CAAI,OAAjB,8BAAiB,S;iBACjD,oBAAGB,KAAhB,C;c
AAgD,OAAjB,8BAAiB,e;eAE5C,cAA0B,W;cAC1B,kBAAW,OAAX,C;UAXxB,K;MAAA,W;K;IAGCJ,4B;
MAMW,Q;MAJP,IAAI,WAAW,MAAf,C;QAA6B,OAAO,8BAAiB,Y;MAErD,eAAsB,MAAY,W;MAE3B,IAAI,
gBAAJ,C;QACH,IAAI,QAAS,SAAT,QAAl,C;UACI,aAAa,qBAAiB,MAAjB,C;UACb,oBAAsB,M;UACtB,a;UA
ES,OAAT,QAAS,S;QAGb,4BAAiB,MAAjB,C;MATJ,W;K;ICrCJ,0B;MAII,sBAAY,C;K;qEAChB,4B;MAIkE,iB
AAy,KAAZ,C;K;2EAEIE,qB;MAI8D,gB;K;ICIDb,2C;MAC7C,qBAAwC,Q;K;iDAExC,Y;MACmB,Q;MAAA,yB
;MAAA,iB;QAAe,MAAM,6BAAsB,OCAAtB,C;MAApC,eAAe,I;MACf,qBAAc,I;MACd,OAAO,QAAS,W;K;I
CLa,kD;MADrC,e;MACsC,0B;MAAyB,gB;MAD/D,iB;MAAA,uB;K;IAAA,mC;MAAA,sC;O;MAEI,qEAGW,CA
HX,EAGc,IAHd,C;MAKA,iFAGiB,CAHjB,EAGoB,IAHpB,C;MAKA,iFAGiB,CAHjB,EAGoB,IAHpB,C;MAKA,
iFAGiB,CAHjB,EAGoB,IAHpB,C;MAKA,+EAGgB,CAHhB,EAGmB,IAHnB,C;MAKA,yEAGa,CAHb,EAGgB,I
AHhB,C;MAKA,iFAGiB,CAHjB,EAGoB,IAHpB,C;MAKA,6EAGe,CAHf,EAGkB,IAHIB,C;MAKA,6FAGuB,CA
HvB,EAG0B,IAH1B,C;MAKA,yFAGqB,CAHrB,EAGwB,IAHxB,C;MAKA,4EAGc,EAHd,EAGkB,IAHIB,C;MA
KA,0EAGa,EAHb,EAGiB,IAHjB,C;MAKA,gFAGgB,EAHhB,EAGoB,IAHpB,C;MAKA,8EAGe,EAHf,EAGmB,I
AHnB,C;MAKA,wFAGoB,EAHpB,EAGwB,IAHxB,C;MAKA,gEAGQ,EAHR,EAGY,IAHZ,C;MAKA,8DAGO,E
AHP,EAGW,IAHX,C;MAKA,wEAGY,EAHZ,EAGgB,IAHhB,C;MAKA,oEAGU,EAHV,EAGc,IAHd,C;MAKA,k
FAGiB,EAHjB,EAGqB,IAHrB,C;MAKA,oFAGkB,EAHIB,EAGsB,IAHtB,C;MAKA,gFAGgB,EAHhB,EAGoB,I

AHpB,C;MAKA,4FAGsB,EAHtB,EAG0B,IAH1B,C;MAKA,oFAGkB,EAHIB,EAGsB,IAHtB,C;MAKA,wEAGY,EAHZ,EAGgB,IAHhB,C;MAKA,gFAGgB,EAHhB,EAGoB,IAHpB,C;MAKA,gFAGgB,EAHhB,EAGoB,IAHpB,C;MAKA,0EAGa,EAHb,EAGiB,IAHjB,C;MAKA,oGAG0B,EAH1B,EAG8B,IAH9B,C;MAKA,gGAGwB,EAHxB,EAG4B,IAH5B,C;MAUA,oC;K;;IA3JA,+C;MAAA,yB;MAAA,uC;K;;IAKA,qD;MAAA,yB;MAAA,6C;K;;IAKA,qD;MAAA,yB;MAAA,6C;K;;IAKA,oD;MAAA,yB;MAAA,4C;K;;IAKA,iD;MAAA,yB;MAAA,yC;K;;IAKA,qD;MAAA,yB;MAAA,6C;K;;IAKA,mD;MAAA,yB;MAAA,2C;K;;IAKA,2D;MAAA,yB;MAAA,mD;K;;IAKA,yD;MAAA,yB;MAAA,iD;K;;IAKA,kD;MAAA,yB;MAAA,0C;K;;IAKA,iD;MAAA,yB;MAAA,yC;K;;IAKA,oD;MAAA,yB;MAAA,4C;K;;IAKA,mD;MAAA,yB;MAAA,2C;K;;IAKA,wD;MAAA,yB;MAAA,gD;K;;IAKA,4C;MAAA,yB;MAAA,oC;K;;IAKA,2C;MAAA,yB;MAAA,mC;K;;IAKA,gD;MAAA,yB;MAAA,wC;K;;IAKA,8C;MAAA,yB;MAAA,sC;K;;IAKA,qD;MAAA,yB;MAAA,6C;K;;IAKA,sD;MAAA,yB;MAAA,8C;K;;IAKA,oD;MAAA,yB;MAAA,4C;K;;IAKA,0D;MAAA,yB;MAAA,kD;K;;IAKA,sD;MAAA,yB;MAAA,8C;K;;IAKA,gD;MAAA,yB;MAAA,wC;K;;IAKA,oD;MAAA,yB;MAAA,4C;K;;IAKA,oD;MAAA,yB;MAAA,4C;K;;IAKA,iD;MAAA,yB;MAAA,yC;K;;IAKA,8D;MAAA,yB;MAAA,sD;K;;IAKA,4D;MAAA,yB;MAAA,oD;K;8CAKA,gB;MAG2D,OAAK,iBAAL,IAAK,CAAL,KAA2B,IAAK,c;K;IAE3F,kC;MAAA,sC;K;uDACI,oB;MAEQ,IADE,QACF,IAAG,CAAH,IADE,QACF,IAAM,EAAN,C;QADJ,OACgB,sBAAS,QAAT,C;WACZ,IAFE,QAEF,IAAG,EAH,IAFE,QAEF,IAAO,EAAP,C;QAFJ,OAEiB,sBAAS,WAAW,CAAX,IAAT,C;;QACL,MAAM,gCAAYB,eAAY,QAAZ,qBAAZB,C;K;;IAL1B,8C;MAAA,yB;MAAA,6C;QAAA,4B;;MAAA,sC;K;;IA7JJ,+B;MAAA,+yC;K;;IAAA,oC;MAAA,a;AAAA,Y;UAAA,4C;aAAA,kB;UAAA,kD;aAAA,kB;UAAA,kD;aAAA,kB;UAAA,kD;aAAA,iB;UAAA,iD;aAAA,c;UAAA,8C;aAAA,kB;UAAA,kD;aAAA,gB;UAAA,gD;aAAA,wB;UAAA,wD;aAAA,sB;UAAA,sD;aAAA,e;UAAA,+C;aAAA,c;UAAA,8C;aAAA,iB;UAAA,iD;aAAA,gB;UAAA,gD;aAAA,qB;UAAA,qD;aAAA,S;UAAA,yC;aAAA,Q;UAAA,wC;aAAA,a;UAAA,6C;aAAA,W;UAAA,2C;aAAA,kB;UAAA,kD;aAAA,mB;UAAA,mD;aAAA,iB;UAAA,iD;aAAA,uB;UAAA,uD;aAAA,mB;UAAA,mD;aAAA,a;UAAA,6C;aAAA,iB;UAAA,iD;aAAA,iB;UAAA,iD;aAAA,c;UAAA,8C;aAAA,2B;UAAA,2D;aAAA,yB;UAAA,yD;;UAAA,6D;;K;;ICKiD,2C;uBAA+B,O;;K;;IAC5E,8C;MAAA,kE;MAAuB,qCAAK,IAAL,C;MAAvB,Y;K;ICD8B,gC;MAe9B,gBAAiC,YAAY,SAAhB,GAA2B,OAA3B,GAAwC,E;K;uFAGjE,Y;MAAQ,OAAO,aAAY,O;K;yCAE/B,iB;MACW,gBAAp,a;MrGoGG,Q;MAAA,IqGpGc,KrGoGV,IAAS,CAAT,IqGpGU,KrGoGI,IAAS,2BAA3B,C;QAAA,OAAc,qBqGpGxB,KrGoGwB,C;;QqGpGf,MAAM,8BAA0B,mCAAYB,WAAzB,MAA1B,C;;MAAhC,W;K;kDAEJ,gC;MAAgF,OAAA,a1GiMY,W0GjMK,U1GiML,E0GjMiB,Q1GiMjB,C;K;6C0G/L5F,iB;MACI,qCAAU,KAAV,C;MACA,OAAO,I;K;6CAGX,iB;MACI,iBAAGB,SAAN,KAAM,C;MACHB,OAAO,I;K;6CAGX,uC;MACI,OAAA,IAAK,qBAAY,wBAAS,MAArB,EAA6B,UAA7B,EAAyC,QAAzC,C;K;sCAET,Y;MAAYB,UAEK,M;MAL1B,eAAe,E;MACf,YAAY,aAAO,OAAP,GAAgB,CAAhB,I;MACZ,OAAO,SAAS,CAAhB,C;QACI,UAAU,0BAAO,YAAP,EAAO,oBAAP,Q;QACV,IAAQ,eAAJ,GAAL,CAAJ,IAAwB,SAAS,CAArC,C;UACI,WAAW,0BAAO,cAAP,EAAO,sBAAP,U;UACX,IAAS,gBAAL,IAAK,CAAT,C;YACI,WAAW,+BAAW,iBAAX,wBAAkB,gBAAIB,C;;YAEX,WAAW,+BAAW,gBAAX,wBAAiB,iBAAjB,C;;UAGf,gCAAY,GAAZ,C;;MAGR,gBAAS,Q;MACT,OAAO,I;K;6CAGX,iB;MAOI,iBAAGB,SAAN,KAAM,C;MACHB,OAAO,I;K;6CAGX,iB;MAQI,iBAAU,K;MACV,OAAO,I;K;6CAGX,iB;MAQI,iBAAGB,eAAN,KAAM,C;MACHB,OAAO,I;K;6CAGX,iB;MAC2C,2BAAO,KAAP,C;K;6CAE3C,iB;MAOI,gBAAA,IAAK,SAAL,IAAe,wBAAS,MAAxB,C;MACA,OAAO,I;K;uCAGX,Y;MAU6B,kB;K;qDAE7B,2B;K;8CAcA,kB;MAO0C,OAAA,IAAY,SAAY,SAAQ,MAAR,C;K;8CAEIE,8B;MAQ2D,OAAA,IAAY,SAAY,SAAQ,MAAR,EAAgB,UAAhB,C;K;kDAEnF,kB;MAQ8C,OAAA,IAAY,SAAY,aAAY,MAAZ,C;K;kDAEtE,8B;MASI,IAAI,MpGuGwC,YAAU,CoGvGID,IAAoB,aAAa,CAArC,C;QAAwC,OAAO,E;MAC/C,OAAO,IAAY,SAAY,aAAY,MAAZ,EAAoB,UAApB,C;K;4CAGnC,wB;MAWI,oCAAA,4BAAmB,KAAAnB,EAA0B,WAA1B,C;MAEb,gBAAS,a1GmB+E,W0GnB9D,C1GmB8D,E0GnB3D,K1GmB2D,C0GnB/E,YAA6B,KAA7B,IAAqC,a1GgB2B,W0GhBV,K1GgBU,C;M0GfzE,OAAO,I;K;6CAGX,wB;MAQI,oCAAA,4BAAmB,KAAAnB,EAA0B,WAA1B,C;MAEb,gBAAS,a1GK+E,W0GL9D,C1GK8D,E0GL3D,K1GK2D,C0GL/E,uBAA6B,kBAA7B,IAAqC,a1GE2B,W0GFV,K1GEU,C;M0GDzE,OAAO,I;K;6CAGX,wB;MAUI,oCAAA,4BAAmB,KAAAnB,EAA0B,WAA1B,C;MAEb,gBAAS,a1GX+E,W0GW9D,C1GX8D,E0GW3D,K1GX2D,C0GW/E,GAAmC,eAAN,KAAM,CAAnC,GAAsD,a1GdU,W0GcO,K1GdP,C;M0GezE,OAAO,I;K;6CAGX,wB;MAAa,oCAAA,4BAAmB,KAAAnB,EAA0B,WAA1B,C;MAEb,gBAAS,a1G9B+E,W0G8B9D,C1G9B8D,E0G8B3D,K1G9B2D,C0G8B/E,GAAmC,SAAN,KAAM,CAAnC,GA

AgD,a1GjCgB,W0GiCC,K1GjCD,C;M0GkCzE,OAAO,I;K;6CAGX,wB;MAWI,oCAAa,4BAAmB,KAAAnB,EAA0B,WAA1B,C;MAEb,gBAAS,a1G/C+E,W0G+C9D,C1G/C8D,E0G+C3D,K1G/C2D,C0G+C/E,GAAmC,SAAN,KAAm,CAAnC,GAAgD,a1GIDgB,W0GkDC,K1GIDD,C;M0GmDzE,OAAO,I;K;6CAGX,wB;MACuD,2BAAO,KAAP,EAAC,KAAd,C;K;6CAEvD,wB;MAUI,oCAAa,4BAAmB,KAAAnB,EAA0B,WAA1B,C;MAEb,eAAe,wBAAS,M;MACxB,gBAAC,IAAK,S1GnEqE,W0GmEpD,C1GnEoD,E0GmEjD,K1GnEiD,C0GmE1E,GAAkC,QAAIC,GAA6C,IAAK,S1GtES,W0GsEQ,K1GtER,C;M0GuEzE,OAAO,I;K;gDAGX,qB;MAcI,IAAI,YAAY,CAAhB,C;QACI,MAAM,gCAAyB,0BAAuB,SAAvB,MAAZB,C;MAGV,IAAI,aAAa,WAAjB,C;QACI,gBAAS,a1G1F2E,W0G0F1D,C1G1F0D,E0G0FvD,S1G1FuD,C;Q0G4FpF,aAAU,WAAV,MAAuB,SAAvB,M;UACI,qCAAU,CAAV,C;K;gD AKZ,sB;MAQI,oCAAa,4BAAmB,UAAAnB,EAA+B,WAA/B,C;MAEb,OAAO,a1G/GkE,W0G+GjD,U1G/GiD,C;K;gD0GkH7E,gC;MAQI,oCAAa,4BAAmB,UAAAnB,EAA+B,QAA/B,EAAyC,WAAzC,C;MAEb,OAAO,a1GzHiF,W0GyHhE,U1GzHgE,E0GyHpD,Q1GzHoD,C;K;yC0G4H5F,Y;K;uCACa,Y;MAAkC,oB;K;oCAEIC,Y;MAOI,gBAAS,E;MACT,OAAO,I;K;0CAGX,wB;MAQI,oCAAa,2BAAkB,KAAIB,EAAyB,WAAzB,C;MAEb,gBAAS,a1GjK +E,W0GiK9D,C1GjK8D,E0GiK3D,K1GjK2D,C0GiK/E,uBAA6B,kBAA7B,IAAqC,a1GpK2B,W0GoKV,QAAQ,CAAR,I1GpKU,C;K;+C0GuK7E,uC;MAYI,yBAAkB,UAAIB,EAA8B,QAA9B,EAAwC,WAAxC,C;MAEA,gBA Ac,IAAK,S1GILqE,W0GkLpD,C1GILoD,E0GkLjD,U1GILiD,C0GkL1E,GAAuC,KAAvC,GAA+C,IAAK,S1GrL O,W0GqLU,Q1GrLV,C;M0GsLzE,OAAO,I;K;kDAGX,wC;MACI,IAAI,aAAa,CAAb,IAAkB,aAAa,MAAnC,C;Q ACI,MAAM,8BAA0B,iBAAC,UAAAd,kBAAMC,MAA7D,C;MAEV,IAAI,aAAa,QAAjB,C;QACI,MAAM,gCAAy B,gBAAa,UAAb,qBAAqC,QAArC,MAAZB,C;K;+CAId,iB;MAYI,oCAAa,2BAAkB,KAAIB,EAAyB,WAAzB,C; MAEb,gBAAS,a1G7M+E,W0G6M9D,C1G7M8D,E0G6M3D,K1G7M2D,C0G6M/E,GAA6B,a1GhNmC,W0GgNI B,QAAQ,CAAR,I1GhNkB,C;M0GiNzE,OAAO,I;K;kDAGX,gC;MAWI,yBAAkB,UAAIB,EAA8B,QAA9B,EAA wC,WAAxC,C;MAEA,gBAAS,a1G9N+E,W0G8N9D,C1G9N8D,E0G8N3D,U1G9N2D,C0G8N/E,GAAkC,a1GjO 8B,W0GiOb,Q1GjOa,C;M0GkOzE,OAAO,I;K;kDAGX,gE;MAc+C,iC;QAAA,oBAAyB,C;MAAG,0B;QAAA,aA AkB,C;MAAG,wB;QAAA,WAAgB,IAAK,O;MAKIF,IACf,I;MALhB,oCAAa,4BAAmB,UAAAnB,EAA+B,QAA/B ,EAAyC,WAAzC,C;MACb,oCAAa,4BAAmB,iBAAnB,EAAc,oBAAoB,QAApB,GAA+B,UAA/B,IAAtC,EAaiF ,WAAY,OAA7F,C;MAEb,eAAe,iB;MACf,iBAAC,UAAAd,UAA+B,QAA/B,U;QACI,YAAY,eAAZ,EAAy,uBAAZ, UAA0B,yBAAO,KAAP,C;K;kDAIIC,uC;MAcI,iBAAgB,iBAAN,KAAm,EAAe,UAAf,EAA2B,QAA3B,C;MACH B,OAAO,I;K;kDAGX,uC;MAYI,gBAAgB,KAAm,W;MACTb,oCAAa,4BAAmB,UAAAnB,EAA+B,QAA/B,EAAy C,SAAU,OAAAnD,C;MAEb,iBAAU,S1G3R8E,W0G2R1D,U1G3R0D,E0G2R9C,Q1G3R8C,C;M0G4RxF,OAAO,I; K;kDAGX,8C;MAGBI,oCAAa,4BAAmB,KAAAnB,EAA0B,IAAK,OAA/B,C;MAEb,gBAAS,a1GjT+E,W0GiT9D,C 1GjT8D,E0GiT3D,K1GjT2D,C0GiT/E,GAAmC,iBAAN,KAAm,EAAe,UAAf,EAA2B,QAA3B,CAAnC,GAA0E,a 1GpTV,W0GoT2B,K1GpT3B,C;M0GqTzE,OAAO,I;K;kDAGX,8C;MAGBI,oCAAa,4BAAmB,KAAAnB,EAA0B,W AA1B,C;MAEb,gBAAgB,KAAm,W;MACTb,oCAAa,4BAAmB,UAAAnB,EAA+B,QAA/B,EAAyC,SAAU,OAAAnD ,C;MAEb,gBAAS,a1G1U+E,W0G0U9D,C1G1U8D,E0G0U3D,K1G1U2D,C0G0U/E,GAA6B,S1G1UkD,W0G0U9 B,U1G1U8B,E0G0UIB,Q1G1UkB,C0G0U/E,GAAyE,a1G7UT,W0G6U0B,K1G7U1B,C;M0G8UzE,OAAO,I;K;I AliBX,6C;MAAA,uD;MAKOC,2B;MALpC,Y;K;IAQA,8C;MAAA,uD;MAC4C,0BAAK,OAAQ,WAAb,C;MAD5 C,Y;K;IAGA,qC;MAAA,uD;MACuB,0BAAK,EAAL,C;MADvB,Y;K;2EA4hBJ,qB;MAOG,E,OAAA,SAAK,Q;K;u EAER,e,mC;MAQ+E,SAAK,aAAI,KAAJ,EAAW,KAAx,C;K;+EAEPf,kD;MAaI,OAAA,SAAK,kBAAS,UAAT,E AAqB,QAArB,EAA+B,KAA/B,C;K;+EAET,4B;MAY6E,OAAA,SAAK,kBAAS,KAAAT,C;K;qFAEIF,2C;MAWo G,OAAA,SAAK,qBAAY,UAAZ,EAawB,QAAxB,C;K;uFAEZg,2E;MAe2E,iC;QAAA,oBAAyB,C;MAAG,0B;Q AAA,aAAkB,C;MAAG,wB;QAAA,WAAgB,SAAK,O;MAC7I,SAAK,qBAAY,WAAZ,EAAyB,iBAAzB,EAA4C, UAA5C,EAAwD,QAAxD,C;K;qFAET,kD;MAeI,OAAA,SAAK,qBAAY,KAAZ,EAAMB,UAAAnB,EAA+B,QAA/ B,C;K;uFAET,kD;MAaI,OAAA,SAAK,qBAAY,KAAZ,EAAMB,UAAAnB,EAA+B,QAA/B,C;K;qFAET,yD;MAiBI ,OAAA,SAAK,qBAAY,KAAZ,EAAMB,KAAAnB,EAA0B,UAA1B,EAAc,QAAAT,C;K;uFAET,yD;MAiBI,OAA A,SAAK,qBAAY,KAAZ,EAAMB,KAAAnB,EAA0B,UAA1B,EAAc,QAAAT,C;K;qF3GhsBT,qB;MAMoD,OA6B W,8BAAY,cAfrB,YAAY,CAAZ,C;K;yFAZtD,qB;MAYsD,OAeS,8BAAY,cAfrB,YAAY,CAAZ,C;K;iFAEtD,qB; MAaoD,OAaw,8BAAY,c;K;qFAE3E,yB;MAAA,uD;MAAA,4B;QAMoD,+B;O;KANpD,C;IAQA,kC;MAYI,gB AiB2D,8BAAY,c;MAhBvE,OAaw,SAAU,OAAV,GAAmB,CAAvB,GAA0B,SAA1B,GAAoC,qBAAU,CAAV,C; K;iFAG/C,qB;MAaoD,OAaw,8BAAY,c;K;IAE3E,kC;MAU+C,mC;K;IAE/C,oC;MAGoD,QAAQ,cAAA,sCAAK,

mBAAL,EAAYB,sCAAK,mBAA9B,CAAR,6B;K;IAEpD,mC;MAGmD,QAAQ,cAAA,sCAAK,kBAAL,EAawB,s
CAAK,kBAA7B,CAAR,6B;K;IAO/C,iC;MAAQ,OAAA,oCAAA,iBAAQ,2BAAR,C;K;IAEzB,8B;MAOI,IAAI,YA
AO,GAAX,C;QACI,OAAO,I;MAEX,OAAO,gCAA8C,mD;K;IAGzD,6B;MAUI,IAAI,CAAQ,kBAAK,GAAL,CA
AR,iCAAoB,CAAQ,kBAAK,EAAL,CAAR,6BAAxB,C;QACI,OAAO,I;MAEX,IAAI,YAAO,GAAX,C;QACI,OA
AO,K;MAEX,OAAO,uB;K;IAGX,oC;MAUI,IAAI,CAAQ,kBAAK,GAAL,CAAR,iCAAoB,CAAQ,kBAAK,EAA
L,CAAR,6BAApB,IAAwC,CAAQ,kBAAK,EAAL,CAAR,6BAA5C,C;QACI,OAAO,I;MAEX,IAAI,YAAO,GAA
X,C;QACI,OAAO,K;MAGX,OAAO,0BAAiB,uB;K;IAG5B,4B;MASI,IAAI,CAAQ,kBAAK,EAAL,CAAR,6BAA
J,C;QACI,OAAO,I;MAEX,IAAI,YAAO,GAAX,C;QACI,OAAO,K;MAEX,OAAO,sB;K;IAGX,gC;MAUI,IAAI,
CAAQ,kBAAK,EAAL,CAAR,6BAAJ,C;QACI,OAAO,I;MAEX,IAAI,YAAO,GAAX,C;QACI,OAAO,K;MAEX,
OAAO,0B;K;IAGX,gC;MAUI,IAAI,CAAQ,kBAAK,GAAL,CAAR,6BAAJ,C;QACI,OAAO,I;MAEX,IAAI,YAA
O,GAAX,C;QACI,OAAO,K;MAEX,OAAO,0B;K;IAGX,gC;MASI,IAAI,YAAO,GAAX,C;QACI,OAAO,K;MAE
X,OAAO,gCAAoD,yD;K;IAG/D,iC;MAUI,OAAO,aAAQ,EAAR,IAAoB,CAAQ,mBAAU,GAAV,CAAR,6B;K;IA
G/B,iC;MAMiD,kC;K;iF4GtPjD,yB;MAAA,+C;MAAA,4B;QAMuD,OAAK,UAAL,SAAK,C;O;KAN5D,C;IAQA,
gC;MAMiD,4B;MAAA,S;QAAGB,cAAA,S3G4LC,c2G5LD,EAaoB,MAApB,C;MAAhB,W;K;IAEjD,6B;MAI0C
,Q;MAAA,yDAaKB,kBAaKB,SAaIB,C;K;IAE5D,oC;MAKoD,Q;MAAA,yCAAA,KAAb,oBAAuB,kBAaKB,SA
aIB,C;K;IAG3E,8B;MAI4C,Q;MAAA,0DAaMB,kBAaKB,SAaIB,C;K;IAE/D,qC;MAKsD,Q;MAAA,0CAAc,KAA
d,oBAAwB,kBAaKB,SAaIB,C;K;IAE9E,0B;MAIwC,Q;MAAA,wDAaIB,kBAaKB,SAaIB,C;K;IAEzD,mC;MA
KkD,Q;MAAA,wCAAY,KAaZ,oBAAsB,kBAaKB,SAaIB,C;K;IAExE,2B;MAI0C,Q;MAAA,yDAaKB,kBAaKB,
SAaIB,C;K;IAE5D,oC;MAKoD,Q;MAAA,yCAAA,KAAb,oBAAuB,kBAaKB,SAaIB,C;K;IAE3E,6B;MAIyF,kB
AA1C,CAAO,S;MACID,IAAO,QrHeD,WqHfC,CAAH,IAAc,CAAM,kBAApB,KrHeE,WqHf6B,KAAM,GAAN,I
AAKB,kBAAjD,CAAJ,C;QACI,4B;MAFsC,OrHiBnC,W;K;6EqHZX,yB;MAAA,6C;MAAA,4B;QAKmD,0B;O;K
ALnD,C;IAOA,mC;MAIgG,kBAA1C,CAAO,S;MAAR,OACjD,EAAK,QrH2BgB,WqH3BhB,CAAH,IAAc,CAA
M,kBAApB,KrH2BmB,WqH3BY,KAAM,GAAN,IAaKB,kBAAjD,CAAF,CrH2BO,GAAqB,WAArB,GAA+B,I;K
;yFqHxB1C,yB;MAAA,yD;MAAA,4B;QAK0D,gC;O;KAL1D,C;iFAOA,yB;MAAA,6C;MAAA,mC;QAO6D,OA
Aa,SAAR,SAAQ,EAAS,KAAT,C;O;KAP1E,C;iFASA,yB;MAAA,6C;MAAA,mC;QAO8D,OAAa,SAAR,SAAQ,E
AAS,KAAT,C;O;KAP3E,C;IASA,sC;MAMqD,OAAA,SAAY,UAAS,WAAW,KAAX,CAAT,C;K;IAEjE,4B;MAA
sC,QAAM,S3G4EsB,c2G5E5B,C;aACIC,K;aAAA,M;aAAA,M;UADkC,OACT,I;UADS,OAE1B,K;K;IAGZ,2B;
MAKI,IAAI,EAU,CAAV,sBAAa,EAAb,CAAJ,C;QACI,MAAM,gCAAYB,WAAQ,KAAR,kCAAzB,C;MAEV,
OAAO,K;K;IAGX,8B;MAA2D,Q;MACvD,YAAQ,EAAR,IAAe,QAAQ,EAavB,C;QAA8B,cAAO,E;WACrC,YA
AQ,EAAR,IAAe,QAAQ,EAavB,C;QAA8B,cAAO,EAAP,GAAa,EAAb,I;WAC9B,YAAQ,EAAR,IAAe,QAAQ,G
AAvB,C;QAA8B,cAAO,EAAP,GAAa,EAAb,I;WAC9B,WAAO,GAAP,C;QAAMb,S;WACnB,YAAQ,KAAR,IA
AoB,QAAQ,KAA5B,C;QAAwC,cAAO,KAAP,GAaKB,EAaIB,I;WACxC,YAAQ,KAAR,IAAoB,QAAQ,KAA5B,
C;QAAwC,cAAO,KAAP,GAaKB,EAaIB,I;QAC3B,sBAAL,IAAK,C;MrH9CN,a;MqHuCgD,OAQ/C,WAAJ,GA
AiB,EAaJB,GAAyB,E;K;ICIJG,2C;MAHpC,e;MAGqC,kB;MAHrC,iB;MAAA,uB;K;IAAA,kC;MAAA,qC;O;MA
II,qEACY,GADZ,C;MAEA,iEAIU,GAJV,C;K;IAFA,+C;MAAA,wB;MAAA,uC;K;IAEA,6C;MAAA,wB;MAAA
,qC;K;IANJ,8B;MAAA,mF;K;IAAA,mC;MAAA,a;aAAA,a;UAAA,4C;aAAA,W;UAAA,0C;UAAA,4D;K;IAa
wG,4B;MAAE,OAAA,EAAG,M;K;IAA7G,qC;MAAqE,iCAAA,EAAb,EAA0B,OAA1B,0BAAmC,cAAAnC,C;K;IA
QIC,2B;MAAC,kB;K;sCALpC,Y;MAKoC,iB;K;wCALpC,iB;MAAA,sBAKoC,qCALpC,C;K;oCAAA,Y;MAAA,
OAKoC,iDALpC,M;K;oCAAA,Y;MAAA,c;MAKoC,sD;MALpC,a;K;kCAAA,iB;MAAA,2IAKoC,sCALpC,G;K;I
AqB0B,iC;MA8PtB,6B;MArPA,eACoC,O;MACpC,eACsD,QAAR,OAAQ,C;MACTd,uBAAoC,WAAO,OAAP,E
AAwB,QAAR,OAAQ,EAQ,IAAR,CAAxB,C;MACpC,6BAA2C,I;MAI3C,oCAAKd,I;K;0CAHID,Y;MACI,Q;M
AAA,U;MAAA,gD;QAAA,a;QAA8D,gBAAvC,WAAO,YAAP,EAawB,QAAR,YAAQ,EAQ,IAAR,CAAxB,C;
QAA8C,6BtHmCnE,S;QsHnCF,StHoCG,S;MsHpCH,a;K;iDAGJ,Y;MACI,Q;MAAA,U;MAAA,uD;QAAA,a;QtH
VG,gB;QsHWC,IAAY,aAAR,YAAQ,EAawB,EAAX,CAAR,IAAmC,WAAO,YAAQ,EAAS,EAAT,CAAvC,C;UA
AA,eACI,oB;UAEA,OAAO,WAAO,MAA2B,UAAf,YAAR,YAAQ,qBAAU,EAavB,EAaE,qBAAQ,EAAR,EAa3
B,MAAP,EAa2D,QAAR,YAAQ,EAQ,IAAR,CAA3D,C;QACb,4B;QAAO,oCtH0BP,S;QsH/BF,StHgCG,S;Ms
HhCH,a;K;sCAQJ,iB;MAEKb,MAAd,oBAAc,C;MACd,YAAy,oBAAc,MAAK,KAAM,WAAx,C;MAC1B,OAA
O,iBAAiB,KAAM,MAAN,KAAe,CAAhC,IAAqC,oBAAc,UAAAd,KAA2B,KAAM,O;K;8CAGjF,iB;MAEKb,MAA

d,oBAAc,C;MACd,OAAO,oBAAc,MAAK,KAAM,WAAX,C;K;wCAGzB,wB;MAGI,IAAI,QAAQ,CAAR,IAAa,
QAAQ,KAAM,OAA/B,C;QACI,MAAM,8BAA0B,0BAAuB,KAAvB,wBAA8C,KAAM,OAA9E,C;;MAEV,cAAc,
0B;MACd,oBAAoB,K;MACpB,OAAO,OAAQ,MAAK,KAAM,WAAX,C;K;mCAGnB,6B;MAS4C,0B;QAAA,aA
AkB,C;MAC1D,IAAI,aAAa,CAAb,IAAkB,aAAa,KAAM,OAAzC,C;QACI,MAAM,8BAA0B,gCAA6B,UAA7B,w
BAAyD,KAAM,OAAzF,C;;MAEV,OAAqB,SAAd,oBAAc,EAAS,KAAM,WAAf,EAA2B,UAA3B,EAAuC,oBAA
vC,C;K;IAeG,6E;MAAA,mB;QAAE,+BAAK,aAAL,EAAY,kBAAZ,C;O;K;IAA2B,uC;MAAW,OAAA,KAAM,O;
K;sCAZ1E,6B;MAQ+C,0B;QAAA,aAAkB,C;MAC7D,IAAI,aAAa,CAAb,IAAkB,aAAa,KAAM,OAAzC,C;QACI,
MAAM,8BAA0B,gCAA6B,UAA7B,wBAAyD,KAAM,OAAzF,C;;MAEV,OAAO,mBAAiB,6CAAjB,EAA8C,sB
AA9C,C;K;0CAGX,iB;MAMI,OAA2B,SAA3B,iCAA2B,EAAS,KAAM,WAAf,EAA2B,CAA3B,EAA8B,oBAA9
B,C;K;sCAE/B,wB;MAGI,IAAI,QAAQ,CAAR,IAAa,QAAQ,KAAM,OAA/B,C;QACI,MAAM,8BAA0B,0BAAuB
,KAAvB,wBAA8C,KAAM,OAA9E,C;;MAEV,OAA2B,SAAPB,0BAAoB,EAAS,KAAM,WAAf,EAA2B,KAA3B,
EAAkC,oBAAIC,C;K;IA4BL,mD;MAAA,qB;QAAE,2BAAoB,EAAPB,EAAWB,mBAAxB,C;O;K;sCAxB5B,8B;
MAqBI,IAAI,CAAA,YAAZ,WAAZ,EAAS,EAAT,CAAb,IAA+B,CAAA,YAAZ,WAAZ,EAAS,EAAT,CAAhD,C;
QACI,OAAO,KAAM,W5G2E4E,S4G3EnD,oB5G2EmD,E4G3EpC,W5G2EoC,C;;M4GzE7F,OAAO,qBAAQ,KA
AR,EAAe,iCAAf,C;K;sCAGX,4B;MAMI,YAAY,kBAAK,KAAL,C;MACZ,IAAI,aAAJ,C;QAAmB,OAAO,KAA
M,W;MAEhC,gBAAGB,C;MAChB,aAAa,KAAM,O;MACnB,SAAS,mBAAC,MAAd,C;;QAEI,iBAAiB,oB;QACj
B,EAAG,gBAAO,KAAP,EAAC,SAAd,EAAYB,UAAW,MAAM,MAA1C,C;QACH,EAAG,gBAAO,UAAU,UAAV
,CAAP,C;QACH,YAAY,UAAW,MAAM,aAAjB,GAAgC,CAAhC,I;QACZ,QAAQ,UAAW,O;;MACd,oBAAy,M
AAZ,IAAsB,aAAtB,C;MAET,IAAI,YAAY,MAAhB,C;QACI,EAAG,gBAAO,KAAP,EAAC,SAAd,EAAYB,MAAZ
B,C;;MAGP,OAAO,EAAG,W;K;2CAGd,8B;MA0BgB,Q;MALZ,IAAI,CAAA,YAAZ,WAAZ,EAAS,EAAT,CAAb
,IAA+B,CAAA,YAAZ,WAAZ,EAAS,EAAT,CAAhD,C;QACI,uBAA+B,QAAR,YAAQ,EAAQ,GAAR,C;QAC/B,
OAAO,KAAM,W5GoB4E,S4GpBnD,WAAO,YAAP,EAAGB,gBAAhB,C5GoBmD,E4GpBhB,W5GoBgb,C;;M4G
jBjF,yBAAK,KAAL,C;MAAA,iB;QAAe,OAAO,KAAM,W;;MAAxC,YAAY,I;MCoLO,gBAAhB,sB;MDjLC,yBt
G2LgF,0BsG3LzD,CtG2LyD,EsG3LhD,WAAM,MtG2L0C,CAAKC,WsG3LIH,C;MACA,yBAAO,uCAAP,C;MA
CA,yBtGyLgF,0BsGzLnD,WAAM,KAAZ,GAAmB,CAAnB,ItGyLyD,EsGzL7B,YtGyL6B,CAAKC,WsGzLIH,C;
MAHJ,OtHIJG,SuHoUqC,W;K;oCD3K5C,wB;MAO6C,qB;QAAA,QAAa,C;MAMxC,Q;MALd,wBAAWB,KAAx
B,C;MtHrIG,SsHsIW,qBAAQ,KAAR,C;MAAd,cAAuC,UAAS,CAAb,GAAgB,EAAhB,GAA2B,OAAH,EAAG,E
AAK,QAAQ,CAAR,IAAL,C;MAC9D,ajI3JgD,gB;Mii4JhD,gBAAGB,C;MAEF,yB;MAAd,OAAc,cAAc,C;QAAc,
uB;QACV,MAAO,WAAU,mBAAN,KAAM,EAAY,SAAZ,EAAuB,KAAM,MAAM,MAAnC,CAA0C,WAApD,C;
QACP,YAAY,KAAM,MAAM,aAAZ,GAA2B,CAA3B,I;;MAEhB,MAAO,WAAU,mBAAN,KAAM,EAAY,SAAZ
,EAAuB,KAAM,OAA7B,CAAqC,WAA/C,C;MACP,OAAO,M;K;IAgBS,yI;MAAA,wC;MAAA,6B;MAAA,yB;M
AAA,0C;MAAA,oC;MAAA,0C;MAAA,yB;MAAA,6B;MAAA,8B;MAAA,8B;MAAA,kC;K;;;gEAAA,Y;;;iCA
CA,mCAAK,wBAAL,C;cACZ,IAAI,4BAAiB,6BAAS,CAA9B,C;gBACI,gB;gCAAA,iCAAM,wBAAM,WAAZ,O
;oBAAA,2C;yBAAA,yB;gBAAA,Q;;gBADJ,gB;;;cAEI,M;;qCAGY,C;sCACCC,C;cAEjB,gB;;sCACqB,+B;cACj
B,gB;8BAAA,iCtGuI4E,mBsGvItE,wBtGuIsE,EsGvItD,oBtGuIsD,EsGvI3C,qBAAW,MAAM,MtGuI0B,CAAKC,
WsGvI9G,O;kBAAA,2C;uBAAA,yB;cAAA,Q;;cACA,uBAAY,qBAAW,MAAM,aAAjB,GAAgC,CAAhC,I;cACZ,
mBAAQ,qBAAW,O;cAJvB,KAKS,qDALt,EAKS,qBALt,OAKyB,2BAAQ,CAAR,IALzB,KAKsC,gBALtC,S;gB
AAA,gB;;;cAAA,gB;;;cAOA,gB;8BAAA,iCtGkIgF,mBsGIIIE,wBtGkI0E,EsGII1D,oBtGkI0D,EsGII/C,wBAAM,
OtGkIyC,CAAKC,WsGIIH,O;kBAAA,2C;uBAAA,yB;cAAA,Q;;cAhBA,OAgBA,a;;;K;IAjBY,sF;MAAA,
yD;uBAAA,6H;YAAA,S;iBAAA,Q;;iBAAA,uB;O;K;8CAbpB,wB;MAUuD,qB;QAAA,QAAa,C;MACH,E,wBAA
wB,KAAxB,C;MAEA,OAAO,SAAS,gDAAT,C;K;+BAxBX,Y;MAMyC,OAAA,oBAAc,W;K;IAEvD,2B;MAAA,+
B;MAmBI,uBAA4B,WAAO,uBAAP,EAaiC,GAAjC,C;MAC5B,2BAAgC,WAAO,SAAP,EAaoB,GAApB,C;MA
GhC,iCAAsC,WAAO,KAAP,EAaiB,GAAjB,C;K;oDatBtC,mB;MAIwD,oBAAM,oBAAO,OAAP,CAAN,C;K;+C
AExD,mB;MAIoD,OAAA,O5GnEyC,S4GmEnB,oB5GnEmB,E4GmEJ,M5GnEI,C;K;0D4GqE7F,mB;MAI+D,OA
AA,O5GzE8B,S4GyER,wB5GzEQ,E4GyEW,M5GzEX,C;K;gE4G8E7F,mB;MAAG,E,OAAA,O5G9E6B,S4G8EP,8
B5G9EO,E4G8EkB,M5G9EIB,C;K;;;I4GwDjG,uC;MAAA,sC;QAAA,qB;;MAAA,+B;K;;IA5PA,4C;MAAA,+C;M
ACKe,kBAAK,OAAL,EAAC,MAAM,MAAN,CAAd,C;MADIE,Y;K;IAGA,sC;MAAA,+C;MAC6C,kBAAK,OAAL,
EAAC,UAAd,C;MAD7C,Y;K;IA4RO,kG;MAAA,kC;MAAA,8C;MAAA,kC;MAAA,kC;MACH,uBAA+B,a;MA

I/B,sF;MAOA,sBAA0C,I;K;+FAX1C,Y;MAAA,2B;K;+FAEI,Y;MAAQ,qBAAA,kBN/R8C,CM+RxC,CN/RwC,C
M+R9C,C;K;gGAEZ,Y;MAAA,4B;K;IAY2B,oG;MAAA,kC;MAAS,uB;K;mJACG,Y;MAAQ,OAAA,kBAAM,O;
K;wGACrC,iB;MAAuC,Q;MAAA,eAAA,kBN/SG,CM+SG,KN/SH,CM+SH,mBAAGB,E;K;;qGAJnE,Y;MACI,IA
AI,2BAAJ,C;QACI,yH;;MAKJ,OAAO,kC;K;4CAGf,Y;MACI,OAAAY,SAAZ,wBAAY,EAAS,kBAAT,EAAoB,kB
AAM,UAAV,GAAqB,8BAAuB,kBAAM,MAA7B,CAArB,GAA8D,kBAAM,aAN,GAAqB,CAArB,IAA9E,EAA
sG,wBAATg,C;K;gEAEhB,iB;MACI,IAAI,QAAc,iBAAN,kBAAM,CAAIb,C;QACI,YAAkB,kBAAY,YAAW,KA
AX,C;QAC9B,IAAa,KAAT,sBAAiB,KAArB,C;UACI,YAAkB,kBAAY,YAAW,QAAQ,CAAR,IAAX,C;UAC9B,I
AAa,KAAT,sBAAiB,KAArB,C;YACI,OAAO,QAAQ,CAAR,I;;;MAInB,OAAO,QAAQ,CAAR,I;K;IAjCiC,oE;M
AAA,kC;MAA+B,6B;K;mHACHd,Y;MAAQ,OAAA,kBAAM,O;K;IACqC,4E;MAAA,qB;QAAE,yBAAK,EAAL,
C;O;K;qEAA5E,Y;MAAiD,OAAqB,OAAb,aAAR,oBAAQ,CAAA,EAAl,iEAAJ,CAAIb,W;K;wEACvF,iB;MAA4
C,Q;MAAA,eAAA,kBNpSU,CMoSJ,KNpSI,CMoSv,YAAoB,oBAApB,O;K;;;IAdxD,uD;MACI,sBAAiB,I;MACj
B,YAAy,eAAK,KAAL,C;MACZ,IAAI,aAAJ,C;QAAMb,OAAO,I;MAC1B,YAAy,aAAA,KAAM,MAAN,EAAa,
sBAAy,CAAZ,IAAb,C;MAEZ,mE;K;IA2CJ,iD;MAM+B,UAKO,M;MATIC,YAAy,C;MACZ,aAAa,mBAAc,WA
AY,OAA1B,C;MAEb,OAAO,QAAQ,WAAy,OAA3B,C;QACI,WAAW,wBAAY,YAAZ,EAAY,oBAAZ,Q;QACX
,IAAI,SAAQ,EAZ,C;UACI,IAAI,UAAS,WAAy,OAAzB,C;YACI,MAAM,gCAAyB,mCAAZB,C;UAEV,MAAO
,gBAAO,wBAAY,cAAZ,EAAY,sBAAZ,UAP,C;eACJ,IAAI,SAAQ,EAZ,C;UACH,IAAI,UAAS,WAAy,OAAz
B,C;YACI,MAAM,gCAAyB,kCAAZB,C;UAEV,IAAI,uBAAY,KAZ,MAAsB,GAA1B,C;YACI,MAAM,gCAAy
B,4DAAZB,C;UAEV,IAAI,EAAuB,kBAAK,EAAL,CAAvB,0CAAY,KAZ,EAJ,C;YACI,MAAM,gCAAyB,mC
AAZB,C;UAEV,eAA2B,eAAZ,WAAy,EAe,KAaf,EAAsB,KAAM,YAAy,KAAX,C;UAC3B,iBAAWd,MAAv
C,W5G7Kme,W4G6K7C,K5G7K6C,E4G6KtC,Q5G7KsC,C4G6K5B,C;UAExD,IAAI,cAAc,KAAM,YAAy,KA
pC,C;YACI,MAAM,8BAA0B,sBAAmB,UAAAnB,oBAA1B,C;UAEV,MAAO,gBAAO,KAAM,YAAN,aAAkB,UA
AlB,CAAP,C;UACP,QAAQ,Q;;UAER,MAAO,gBAAO,IAAP,C;;;MAGf,OAAO,MAAO,W;K;IAG1B,2D;MAEI,Y
AAy,aAAa,CAAb,I;MACZ,iBAAiB,qBAAK,UAl,IAAmB,E;MAGpC,OAAO,QAAQ,gBAAR,IAAkB,CAAe,k
BAAK,EAAL,CAAF,wCAAK,KAAL,EAZB,C;QACI,oBAAoB,CAAC,aAAa,EAAb,IAAD,KAaqB,qBAAK,KA
AL,IAAc,EAAnC,K;QACpB,IAAQB,CAAJB,qCAAyB,UAA7B,C;UACI,aAAa,a;UACb,qB;;UAEA,K;;;MAGR,O
AAO,K;K;I5GraX,yB;MAQiB,Q;MADb,aAAa,E;MACb,wBAAa,KAAb,gB;QAAa,WAAb,UAAa,KAAb,O;QACI,
8BAAU,IAAV,C;;MAEJ,OAAO,M;K;IAGX,yC;MAa+B,Q;MAH3B,IAAI,SAAS,CAAT,IAAc,SAAS,CAAvB,IA
A4B,CAAA,KAAM,OAAN,GAAa,MAAb,QAAsB,MAAtD,C;QACI,MAAM,8BAA0B,WAAAS,KAAM,OAaf,kB
AA+B,MAA/B,kBAAGD,MAA1E,C;MACV,aAAa,E;MACc,gBAAS,MAAT,I;MAA3B,iBAAc,MAAd,wB;QACI,
8BAAU,MAAM,KAAN,CAAV,C;;MAEJ,OAAO,M;K;IAGX,mC;MAOiB,Q;MADb,aAAa,E;MACb,wBAAa,SAA
b,gB;QAAa,WAAb,UAAa,SAAb,O;QACI,8BAAU,IAAV,C;;MAEJ,OAAO,M;K;IAGX,2D;MAY2C,0B;QAAA,a
AAkB,C;MAAG,wB;QAAA,WAAgB,SAAK,O;MACjF,oCAAA,4BAAMb,UAAAnB,EAA+B,QAA/B,EAAYC,SAA
K,OAA9C,C;MACb,aAAa,E;MACb,iBAAc,UAAAd,UAA+B,QAA/B,U;QACI,8BAAU,UAAK,KAAL,CAAV,C;;M
AEJ,OAAO,M;K;IASkB,gD;MAAA,qB;QAAE,+CAAI,EAJ,E;O;K;IAN/B,kC;MAMI,OAAO,kBAAU,gBAAV,
EAAkB,+BAAIB,C;K;IAiBiC,oE;MAAA,qB;QAAE,+CAAI,qBAAa,EAAb,IAAJ,E;O;K;IA9C,wD;MAYqC,0B;
QAAA,aAAkB,C;MAAG,wB;QAAA,WAAgB,SAAK,O;MAC3E,oCAAA,4BAAMb,UAAAnB,EAA+B,QAA/B,EA
AyC,gBAAZC,C;MACb,OAAO,kBAAU,WAAW,UAAAX,IAAV,EAaiC,2CAAjC,C;K;IAGX,mC;MAQI,OAAO,W
AAW,SAAX,EAaiB,CAAJB,EAAoB,gBAAPB,EAA0B,KAA1B,C;K;IAGX,mF;MAeI,0B;QAAA,aAAkB,C;MAC
IB,wB;QAAA,WAAgB,SAAK,O;MACrB,sC;QAAA,yBAAKC,K;MAEIC,oCAAA,4BAAMb,UAAAnB,EAA+B,QA
A/B,EAAYC,SAAK,OAA9C,C;MACb,OAAO,WAAW,SAAX,EAaiB,UAAjB,EAA6B,QAA7B,EAAuC,sBAAvC,
C;K;IAGX,sC;MAQI,OAAO,WAAW,SAAX,EAaiB,CAAJB,EAAoB,gBAAPB,EAA4B,KAA5B,C;K;IAGX,sF;M
AeI,0B;QAAA,aAAkB,C;MACIB,wB;QAAA,WAAgB,SAAK,O;MACrB,sC;QAAA,yBAAKC,K;MAEIC,oCAAA,
4BAAMb,UAAAnB,EAA+B,QAA/B,EAAYC,gBAAZC,C;MACb,OAAO,WAAW,SAAX,EAaiB,UAAjB,EAA6B,Q
AA7B,EAAuC,sBAAvC,C;K;uFAGX,qB;MAMwD,OAAA,SAAY,c;K;mFAEpE,qB;MAWsD,OAAA,SAAY,c;K;
uFAEIE,qB;MAMwD,OAAA,SAAY,c;K;mFAEpE,qB;MAWsD,OAAA,SAAY,c;K;yFAEIE,qC;MACoF,OAAA,S
AAy,SAAQ,GAAR,EAAa,SAAb,C;K;iGAehG,qC;MACwF,OAAA,SAAY,aAAy,GAZ,EAaiB,SAAJB,C;K;+F
AepG,kC;MACiF,OAAA,SAAY,YAAW,CAAX,EAAC,QAAd,C;K;2FAE7F,wB;MACgE,OAAA,SAAY,UAAAS,C
AAT,C;K;iFAE5E,iC;MACqE,OAAA,SAAY,WAAU,UAAV,C;K;mFAEjF,2C;MACoF,OAAA,SAAY,WAAU,U

AAV,EAAsB,QAAtB,C;K;4EAEhG,0B;MAGuD,OAAA,SAAY,QAAO,GAAP,C;K;wEAEnE,4B;MAGgE,OAAA,SAAY,OAAM,KAAN,C;K;yFAK5E,2C;MACyF,OAAA,SAAY,SAAQ,OAAR,EAAiB,WAAjB,C;K;IAErG,iD;M AOkD,0B;QAAA,aAAsB,K;MACpE,IAAI,UAAJ,C;QACI,SAAS,SAAK,O;QACd,SAAS,KAAM,O;QACf,UTGG,MAAO,KSHM,ETGN,ESHU,ETGV,C;QSFV,IAAI,QAAO,CAAX,C;UAAc,OAAO,KAAK,EAAL,I;QACrB,iBA Ac,CAAd,UAAsB,GAAtB,U;UACI,eAAe,qBAAK,KAAL,C;UACf,gBAAgB,iBAAM,KAAN,C;UAEhB,IAAI,aA AY,SAAhB,C;YACI,WAAoB,cAAT,QAAS,C;YACpB,YAAsB,cAAV,SAAU,C;YAEtB,IAAI,aAAY,SAAhB,C;c ACwB,kBAAT,Q;cAAX,WDI02C,gCAAY,cAfrB,YAAY,CAAZ,C;cCkPZ,kBAAV,S;cAAZ,YDnO2C,gCAAY,c AfrB,YAAY,CAAZ,C;cCoPIC,IAAI,aAAY,SAAhB,C;gBACI,OAAgB,iBAAT,QAAS,EAAU,SAAV,C;:::QAKhC ,OAAO,KAAK,EAAL,I;QAEP,OAAO,4BAAU,KAAV,C;K;IAIf,4C;MAOqF,oCAAkB,KAAIB,C;K;IAErF,wD; MASI,OAAW,UAAJ,GACE,4BAAL,SAAK,EAA4B,KAA5B,CADF,GAGE,kBAAL,SAAK,EAakB,KAAIB,C;K; IAIkD,oD;MAAU,OAAE,UAAF,CAAE,EAAU,CAAV,EAA0B,IAA1B,C;K;IAIvE,+C;MAAQ,oC;K;2F8G/SZ,oC ;MACiF,O9G2Me,kB8G3ME,oBAAH,EAAG,C9G2MF,E8G3Mc,S9G2Md,C;K;mG8GzMHg,oC;MACqF,O9G2M e,sB8G3MM,oBAAH,EAAG,C9G2MN,E8G3MkB,S9G2MIB,C;K;I8GzMpG,mD;MAIoD,0B;QAAA,aAAsB,K;M ACtE,IAAI,CAAC,UAAAL,C;QACI,O9GsMqF,qB8GtM7D,M9GsM6D,E8GtMrD,C9GsMqD,C;Q8GpMrF,OAAO, yBAAc,CAAd,EAAiB,MAAjB,EAAyB,CAAzB,EAA4B,MAAO,OAAnc,EAA2C,UAA3C,C;K;IAGf,iE;MAIqE,0 B;QAAA,aAAsB,K;MACvF,IAAI,CAAC,UAAAL,C;QACI,O9G2LqF,qB8G3L7D,M9G2L6D,E8G3LrD,U9G2LqD, C;Q8GzLrF,OAAO,yBAAc,UAAAd,EAA0B,MAA1B,EAakC,CAAIC,EAAqC,MAAO,OAA5C,EAAoD,UAApD, C;K;IAGf,iD;MAIkD,0B;QAAA,aAAsB,K;MACpE,IAAI,CAAC,UAAAL,C;QACI,O9GmLoE,mB8GnL9C,M9Gm L8C,C;Q8GjLpE,OAAO,yBAAc,mBAAS,MAAO,OAAhB,IAAd,EAAc,MAAtC,EAA8C,CAA9C,EAAiD,MAA O,OAAxD,EAAgE,UAAhE,C;K;IAGf,mC;MAGI,aACa,S9G0L2D,O8G1LhD,K9G0LgD,C;M8GzLxE,OAAO,kB AAKB,MAAO,OAAP,KAAe,C;K;IAG5C,4B;MAKoD,gCAAU,C;MAAV,U;QAAuB,kBAAR,yB;QAAQ,c;UrH2n DvD,U;UADhB,IAAI,OCAAsB,qBAA1B,C;YAAqC,aAAO,I;YAAP,e;UACrB,+B;UAAhB,OAAgB,gBAAhB,C;Y AAgB,2B;YAAM,IAAI,CqH3nD4D,aAAT,qBrH2nDxC,OqH3nDwC,CAAS,CrH2nDhE,C;CAAYB,aAAO,K;cAAP ,e;UAC/C,aAAO,I;QqH5nDgE,iB;MAAvB,W;K;IAEpD,gD;MASiD,0B;QAAA,aAAsB,K;MAOxC,Q;MAN3B, IAAI,iBAAJ,C;QAakB,OAAO,a;MACzB,IAAI,aAAJ,C;QAAMB,OAAO,K;MAC1B,IAAI,CAAC,UAAAL,C;QAAi B,OAAO,kBAAQ,KAAR,C;MAExB,IAAI,SAAK,OAAL,KAAe,KAAM,OAazB,C;QAAiC,OAAO,K;MAEb,OA AL,SAAK,O;MAA3B,iBAAc,CAAd,wB;QACI,eAAe,qBAAK,KAAL,C;QACf,gBAAgB,iBAAM,KAAN,C;QACH B,IAAI,CAAU,SAAT,QAAS,EAAO,SAAP,EAakB,UAAIB,CAAd,C;UACI,OAAO,K;MAIf,OAAO,I;K;IAIX,sF ;MACKH,0B;QAAA,aAAsB,K;MACpI,oCAAkB,UAAIB,EAA8B,KAA9B,EAAqC,WAArC,EAakD,MAAID,EA A0D,UAA1D,C;K;IAGJ,+B;MAYI,OxGmMmD,mBAAS,CwGnM5D,G9GwH4F,oB8GxHzD,C9GwHyD,E8GxHt D,C9GwHsD,CaVc9B,c8GjFrC,G9GqHoD,oB8GrHZ,C9GqHY,C8GrH7E,GAAyE,S;K;IAG7E,iC;MASI,OxGuL mD,mBAAS,CwGvL5D,G9G4G4F,oB8G5GzD,C9G4GyD,E8G5GtD,C9G4GsD,CAIB9B,c8G1FrC,G9GyGoD,oB 8GzGZ,C9GyGY,C8GzG7E,GAAyE,S;K;IAG7E,8B;MAOiB,IAAN,I;M3H/FP,IAAI,E2H8FI,KAAK,C3H9FT,CA AJ,C;QACI,c2H6Fc,oD;Q3H5Fd,MAAM,gCAAyB,OAAQ,WAAjC,C;M2H6FH,QAAM,CAAN,C;aACH,C;UAA K,S;UAAAL,K;aACA,C;UAAU,OAAL,SAAK,W;UAAV,K;UAEL,aAAa,E;UACb,IAAI,ExGgKoC,qBAAU,CwGh K9C,CAAJ,C;YACI,QAAQ,SAAK,W;YACb,YAAY,C;YACZ,OAAO,IAAP,C;cACI,IAAI,CAAC,QAAU,CAAX, MAAiB,CAArB,C;gBACI,UAAU,C;cAEd,QAAQ,UAAW,C;cACnB,IAAI,UAAAS,CAAb,C;gBACI,K;cAEJ,KAA K,C;UAGb,OAAO,M;MANbF,W;K;IAwBJ,4D;MAOqE,0B;QAAA,aAAsB,K;MACvF,O9GkFiG,kB8GIFnF,W AAO,6BAAM,gBAAO,QAAP,CAAb,EAAMC,UAAJ,GAAGB,KAAhB,GAA2B,IAA1D,C9GkFmF,E8GIFIB,6BA AM,iCAAwB,QAAXB,C9GkFY,C;K;I8GhFrG,4D;MAM+D,0B;QAAA,aAAsB,K;MACjF,O9GyEiG,kB8GzEnF, WAAO,6BAAM,gBA Ae,oBAAR,OAAQ,CAAf,CAAb,EAA6C,UAAJ,GAAGB,KAAhB,GAA2B,IAApE,C9GyEm F,E8GzEA,oBAAR,OAAQ,C9GyEA,C;K;I8GvErG,iE;MAC0E,0B;QAAA,aAAsB,K;MAC5F,O9GqEiG,kB8GrEn F,WAAO,6BAAM,gBAAO,QAAP,CAAb,EAAMC,UAAJ,GAAGB,IAAhB,GAA0B,GAAzD,C9GqEmF,E8GrEpB, 6BAAM,iCAAwB,QAAXB,C9GqEc,C;K;I8GnErG,iE;MACoE,0B;QAAA,aAAsB,K;MACTF,O9GiEiG,kB8GjEnF, WAAO,6BAAM,gBA Ae,oBAAR,OAAQ,CAAf,CAAb,EAA6C,UAAJ,GAAGB,IAAhB,GAA0B,GAAne,C9GiEm F,E8GjEF,oBAAR,OAAQ,C9GiEE,C;K;I+G7OrG,kD;MAEL,IAAI,gBAAJ,C;QAAsB,MAAM,6BAAyB,qCAAKC, QAAQ,CAAR,IAAIC,CAAzB,C;MAC5B,OAAO,CAAC,IAAD,I;K;IAGX,iF;MAQL,IAAI,EAAS,KAAT,oBAAiB, KAAjB,KAA2B,SAAS,QAAXC,C;QACI,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAAPB,C;MAEX,UAA

U,kBAAO,KAAP,C7GwBgC,I;M6GvB1C,IAAI,EAAQ,KAAR,kBAAgB,KAAhB,CAAJ,C;QACI,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;MAEX,OAAO,SAAW,CAAC,OAAS,IAAV,KAAqB,EAAhC,IAAwC,MAAQ,I;K;IAG3D,yE;MAQI,IAAI,SAAU,EA AV,MAAkB,CAAI B,IAAuB,SAAS,QAApC,C;QACI,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;MAEX,YAAY,KAAa,CAAP,KAAO,C;MACzB,IAAI,SAAU,GA AV,MAAkB,GAAtB,C;QACI,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;MAEX,OAAQ,SAAU,CAAX,GA AkB,KAAIB,GAA4B,I;K;IAGvC,yE;MASI,IAAI,SAAS,QAAb,C;QACI,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;MAGX,YAAY,KAAa,CAAP,KAAO,C;MACzB,IAAI,SAAU,EA AV,MAAiB,CAArB,C;QACI,IAAI,SAAU,GA AV,MAAkB,GAAtB,C;UAEI,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;aAER,IAAI,SAAU,EA AV,MAAiB,EAArB,C;QACH,IAAI,SAAU,GA AV,MAAkB,GAAtB,C;UAEI,OAAO,UAAU,CA AV,EAAa,KAAb,EAAoB,gBAApB,C;;aAER,IAAI,SAAU,GA AV,MAAkB,GAAtB,C;QACH,OAAO,UAAU,CA AV,EAAa,KAAb,EAAoB,gBAApB,C;;MAGX,IAAI,SAAQ,CAAR,UAAa,QAAjB,C;QACI,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;MAEX,YAAY,KAAiB,CAAX,QAAQ,CAAR,IAAW,C;MAC7B,IAAI,SAAU,GA AV,MAAkB,GAAtB,C;QACI,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;MAGX,OAAQ,SAAU,EAAX,GAAoB,SAAU,CAA9B,GAAqC,KAArC,GAA+C,O;K;IAG1D,yE;MASI,IAAI,SAAS,QAAb,C;QACI,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;MAGJ,YAAY,KAAa,CAAP,KAAO,C;MACzB,IAAI,SAAU,EA AV,MAAiB,CAArB,C;QACI,IAAI,SAAU,GA AV,KAAkB,GAAtB,C;UAEI,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;aAER,IAAI,SAAU,EA AV,MAAiB,CAArB,C;QACH,IAAI,SAAU,GA AV,MAAkB,GAAtB,C;UAEI,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;aACJ,IAAI,SAAU,GA AV,MAAkB,GAAtB,C;QACH,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;MAGX,IAAI,SAAQ,CAAR,UAAa,QAAjB,C;QACI,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;MAEX,YAAY,KAAiB,CAAX,QAAQ,CAAR,IAAW,C;MAC7B,IAAI,SAAU,GA AV,MAAkB,GAAtB,C;QACI,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;MAGX,IAAI,SAAQ,CAAR,UAAa,QAAjB,C;QACI,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;MAEX,YAAY,KAAiB,CAAX,QAAQ,CAAR,IAAW,C;MAC7B,IAAI,SAAU,GA AV,MAAkB,GAAtB,C;QACI,OA AO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;MAEX,OAAQ,SAAU,EAAX,GAAoB,SAAU,EA A9B,GA AuC,SAAU,CAAjD,GAAwD,KAAxD,GA AkE,O;K;;IAMb7E,oE;MAkB0B,UAGJ,MAHI,EAKJ,MALI,EAMJ,M ANI,EASJ,MATI,EAUJ,MAVI,EA WJ,MAXI,EA gBA,MAhBA,EAiBA,MAjBA,EAkBA,MAIBA,EAoBA,MApBA,EAqBA,OArBA,EAsBA,OAtBA,EAuBA,O;M5H9JtB,IAAI,E4HgII,cAAc,CAAd,IAAmB,YAAY,MAAO,OAAtC,IAAgD,cAAc,Q5HhIIE,CAAJ,C;QACI,cAda,qB;QAeb,MAAM,gCAAYB,OAAQ,WAAjC,C;;M4HgIV,YAAY,cA AU,CAAC,WAAW,UAAAX,IAAD,IAA0B,CAA1B,IAAV,C;MACZ,gBAAgB,C;MACHb,gBAAgB,U;MAEHb,OA AO,YAAY,QAA nB,C;QACI,WAAW,mBAAO,gBAAP,EAAO,wBAAP,Q7G1H2B,I;Q6G4HIC,WAAO,GAAP,C;UACI,MAAM,kBAAN,EAAM,0BAAN,YAA0B,OAAL,IAAK,C;eAC9B,WAAO,IAAP,C;UACI,MAAM,kBAAN,EAAM,0BAAN,YAA4C,OAArB,QAAS,CAAV,GAAGB,GAAM,C;UAC5C,MAAM,kBAAN,EAAM,0BAAN,YAA+C,OAAXB,OAAS,EA AV,GAAM B,GAAM,C;eAEnD,WAAO,KAAP,IAAiB,QAAQ,KAAzB,C;UACI,MAAM,kBAAN,EAAM,0BAAN,YAA6C,OAAtB,QAAS,EA AV,GA AiB,GAAM,C;UAC7C,MAAM,kBAAN,EAAM,0BA AN,YAAuD,OAA/B,QAAS,CAAV,GA AiB,EA AiB,GAA2B,GAAM,C;UACvD,MAAM,kBAAN,EAAM,0BAAN,YAA+C,OAAXB,OAAS,EA AV,GAAM B,GAAM,C;;UAG/C,gBAAgB,uBAAuB,MAAvB,EAA+B,IAA/B,EA AqC,SAArC,EAAGD,QA AhD,EA A0D,gBAA1D,C;UACHb,IAAI,aAAa,CAAjB,C;YACI,MAAM,kBAAN,EAAM,0BA AN,YAAqB,0BAA0B,CAA1B,C;YACrB,MAAM,kBAAN,EAAM,0BAAN,YAAqB,0BAA0B,CAA1B,C;YACrB,MAAM,kBAAN,EAAM,0BAAN,YAAqB,0BAA0B,CAA1B,C;;YAErB,MAAM,kBAAN,EAAM,0BAAN,YAAkD,OA A3B,aAAc,EA Af,GAAsB,GAAM,C;YACID,MAAM,mBAAN,EAAM,2BAAN,aAA6D,OA ArC,aAAc,EA Af,GA AuB,EA AxB,GA AiC,GAAM,C;YAC7D,MAAM,mBAAN,EAAM,2BAAN,aAA4D,OA ApC,aAAc,CA Af,GA AsB,EA AvB,GA AgC,GAAM,C;YAC5D,MAAM,mBAAN,EAAM,2BAAN,aAAoD,OA A7B,YAAc,EA Af,GA AwB,GAAM,C;YACpD,6B;;MAMhB,OAAW,KAAM,OAAN,KAAc,SAAlB,GAA6B,KAA7B,GAA8C,UAAN,KAA M,EA AO,SAAP,C;K;;IAQzD,mE;MAiByB,Q;M5H9LrB,IAAI,E4HwLI,cAAc,CAAd,IAAmB,YAAY,KAAM,OA ArC,IAA6C,cAAc,Q5HxL/D,CAAJ,C;QACI,cAda,qB;QAeb,MAAM,gCAAYB,OAAQ,WAAjC,C;;M4HwLV,gBA AgB,U;MACHb,oBAAoB,sB;MAEpB,OAAO,YAAY,QAA nB,C;QACI,WAAW,KAAmB,CAAb,gBAAa,EAAb,wBAAa,O;QAE1B,YAAQ,CAAR,C;UACI,aAAc,gBAAY,OAAL,IAAK,CAAZ,C;aACIB,YAAS,CAAT,KAAc,EA

Ad,C;UACI,WAAW,eAAe,KAAf,EAAsB,IAAtB,EAA4B,SAA5B,EAAuC,QAAvC,EAAiD,gBAAjD,C;UACX,IAAI,QAAQ,CAAZ,C;YACI,aAAc,gBAAO,gBAAP,C;YACd,yBAAa,CAAC,IAAD,IAAb,K;;YAEA,aAAc,gBAAY,OAAL,IAAK,CAAZ,C;YACd,wBAAa,CAAb,I;;eAGR,YAAS,CAAT,KAAc,EAAd,C;UACI,aAAW,eAAe,KAAf,EAAsB,IAAtB,EAA4B,SAA5B,EAAuC,QAAvC,EAAiD,gBAAjD,C;UACX,IAAI,UAAQ,CAAZ,C;YACI,aAAc,gBAAO,gBAAP,C;YACd,yBAAa,CAAC,MAAD,IAAb,K;;YAEA,aAAc,gBAAY,OAAL,MAAK,CAAZ,C;YACd,wBAAa,CAAb,I;;eAGR,YAAS,CAAT,KAAc,EAAd,C;UACI,aAAW,eAAe,KAAf,EAAsB,IAAtB,EAA4B,SAA5B,EAAuC,QAAvC,EAAiD,gBAAjD,C;UACX,IAAI,UAAQ,CAAZ,C;YACI,aAAc,gBAAO,gBAAP,C;YACd,yBAAa,CAAC,MAAD,IAAb,K;;YAEA,WAAy,MAAD,GAAQ,KAAR,IAAqB,EAARb,GAA2B,K;YACtC,UAAW,SAA S,IAAV,GAAoB,K;YAC9B,aAAc,gBAAY,OAAL,IAAK,CAAZ,C;YACd,aAAc,gBAAW,OAAL,GAAL,CAAX,C;YACd,wBAAa,CAAb,I;;UAIJ,UAAU,CAAV,EAAsB,SAAb,EAAbW,gBAAxB,C;UACA,aAAc,gBAAO,gBAAP,C;;MAK1B,OAAO,aAAc,W;K;ICtQzB,uC;MAU2D,OAAwB,CAAxB,2BAAwB,mBAAS,SAAT,C;K;IAEnF,oC;MAKI,OAAQ,OAAW,mBAAL,SAAK,CAAX,C;K;IAGZ,6C;MAMI,IAAI,cAAS,SAAb,C;QACI,iBAAsB,SAAY,Y;QACIC,IAAI,kBAAJ,C;UACS,SAAL,eAA+B,iBAAc,SAAd,E;;UAE/B,UAAW,WAAI,SAAJ,C;;K;IAUnB,6C;MAC4B,UAAjB,M;MAAP,OAAO,WAAiB,OAAsB,SAAY,YAAjB,4CAA+D,W;K;IAI9E,iC;MACI,gBAAqB,sB;MACrB,iBAAsB,E;MACTb,kBAA+B,E;MAC/B,uBAAiC,C;K;uDAEjC,qB;MACc,qBAAV,SAAU,EAAC,EAAd,EAAkB,EAAIB,C;MACV,OAAO,aAAO,W;K;gDAGlB,qB;MAA6D,gBAAR,c;MAAQ,c;;Q3I4IY7C,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UAAsB,IAAc,O2I5IY+B,c3I4IY7C,C;YAAwB,aAAO,I;YAAP,e;;QAC9C,aAAO,K;;M2I7IY8C,iB;K;sDAErD,wC;MACI,KAAK,qBAAL,SAAK,EAAC,MAAd,EAAsB,SAAtB,CAAL,C;QAAyC,M;MAEzC,YAAY,SAAK,M;MACjB,OAAO,aAAP,C;QACI,KAAM,qBAAN,KAAM,EAAC,MAAd,EAAsB,aAAtB,CAAN,C;UAA8C,M;QAC9C,QAAQ,KAAM,M;;K;sDAItB,wC;MASgB,IAAiB,IAAjB,EA2BE,M;MANCd,aAAO,gBAAO,MAAP,CAAe,gBAAO,SAAP,C;MACTb,gBAAgB,SAAK,W;MACrB,IAAI,eAAQ,SAAR,CAAJ,C;QACI,aAAO,gBAAO,kCAAP,CAA2C,gBAAO,SAAP,CAAkB,gBAAO,KAAP,C;QACpE,OAAO,K;;MAEH,cAAY,MAAK,SAAL,C;MAEpB,YAAY,CAAiB,OAAsB,SAAY,MAAjB,2D;MACZ,IAAI,aAAJ,C;Q1HyBG,S0HxBwB,WAAN,KAAM,EAQ,SAAR,C;QAAvB,iBAAOd,KAAK,CAAT,GAAY,CAAZ,GAAMB,KAAe,gBAAf,I;QACnE,IAAI,eAAc,CAAIb,C;UAAqB,aAAO,gBAAO,SAAP,CAAkB,gBAAO,IAAP,C;QAC9C,IAAI,e1G8MoC,YAAU,C0G9MID,C;UACI,kBAAW,K;UACX,uBAAgB,U;;UAEhB,QAAQ,wBAAiB,KAAjB,EAAbW,UAAxB,C;;QAEZ,IAAI,M1GgNuC,UAAS,C0GhNpD,C;UAEuB,U;UAAA,IAAI,eAAc,CAAIb,C;YAAA,SAAQ,B,C;;Y3Gq+BpC,U;YADhB,YAAY,C;YACI,oB2Gr+B+C,S3Gq+B/C,C;YAAhB,OAAGB,gBAAhB,C;CAAGB,sC;CAAM,I2Gr+BgE,U3Gq+BID,oB2Gr+BkD,MAAK,E3Gq+BrE,C;gBAAwB,qB;;Y2Gr+Bf,SAA4B,I3Gs+BpD,K2Gt+BoD,I;;UAA/C,yB;U7GorCC,kB;UADb,YAAY,C;UACC,S6GnrCK,aAAN,KAAM,C7GmrCL,W;UAAb,OAAa,gBAAb,C;YAAa,wB;Y6GlrCG,I7GkrCU,oBAAmB,cAAnB,EAAMb,sBAAnB,U6GlrCN,gBAAJ,C;CAA2B,aAAO,uB;YACI,C,aAAO,gB7GirCgC,I6GjrChC,CAAa,gBAAO,IAAP,C;;UAGxB,aAAO,gBAAO,KAAP,CAAc,gBAAO,IAAP,C;;QAGzB,aAAO,gBAAO,SAAP,CAAkB,gBAAO,IAAP,C;;MAG7B,iBAAiB,mC;MACjB,IrIuHoD,CqIvHhD,UrIuHiD,UqIvHrD,C;QACI,uBAAuB,SAAS,M;QACTb,8B;QAAV,OAAU,gBAAV,C;UAAU,qB;UACJ,qBAAF,CAAe,EAAC,gBAAd,EAAGC,cAAhC,C;;MAGV,OAAO,I;K;yDAGX,6B;MAIwB,Q;MAHpB,mBAAwB,C;MACxB,gBAAqB,C;MACrB,mBAAwB,C;MACJ,OzHyIjB,MAAO,KyHzIgB,eAAS,OAAT,GAAkB,oBAAIb,IzHyIhB,EyHzIiD,KAAM,OAAN,GAAe,UAAf,IzHyIjD,C;MyHzIV,eAAY,CAAZ,oB;QACI,QAAQ,iBAAY,iBAAN,KAAM,CAAN,GAAkB,GAAIB,IAAN,C;QACR,IAAI,MAAK,2BAAkB,iBAAT,eAAS,CAAT,GAAqB,GAARb,IAAT,CAAT,C;UAA6C,K;QAC7C,IAAI,MAAK,EAAT,C;UACI,8BAAGB,CAAhB,I;UACA,eAAe,S;UACf,YAAY,G;;MAGpB,IAAI,gBAAGB,CAApB,C;QAAuB,OAAO,K;MAC9B,OAAO,eAAe,CAAf,IAAoB,iBAAY,iBAAN,KAAM,CAAN,IAAmB,YAAnB,GAAC,CAAIC,KAAN,MAA+C,EAA1E,C;QACI,8BAAGB,CAAhB,I;MAGJ,OAAa,YAAN,KAAM,EAAS,YAAT,CAAN,IAA+B,cAAW,eAAe,CAAf,IAAX,uCAA/B,C;K;;yHC/H+C,Y;MAAQ,W;K;IAEtE,gD;MACkB,UAMP,M;MANO,IAAI,aAAY,CAAhB,C;QACV,Y;;QAEA,UxBsY8C,MAAW,KwBtY/C,IxBsY+C,EwBtYtC,QxBsYsC,C;QwBrYzD,OAAA,IAAO,OxB2UmC,MAAW,KwB3UpC,KxB2UoC,CwB3UxC,GAAa,GAAnB,CAAP,GAAiC,GAAjC,GxBwV2C,MAAW,MwBxVV,KxBwVU,C;;MwB5V1D,kB;MAMO,IxByUuC,MAAW,KwBzU1C,OxBuU0C,CwBzU9C,GAAe,MAAnB,C;QAEmC,SAA9B,OAAY,SAAQ,QAAR,C;;QAGpB,exBoU0C,MAAW,KwBpUIC,OxBuUkC,C;QwBnUrD,qBAA8B,QAAy,axBgRC,MAAW,MAvCV,MAAW,OwBzOU,QxByOV,CAuCD,CwBhRA,GAAwB,QAAP,C;QAC1C,SAAL,UAAU,CAAd,GAAiB,MAAG,cAAPB,GAAyC,c;;M

AP7C,a;K;IAWJ,6C;MACI,OAAa,KAAY,gBA Ae,OAAf,EAAwB,MAAK,4BAA2B,QAA3B,CAAL,EAAxB,C;K;I
CtBQ,4C;MAFrC,e;MAEsC,0B;MAFtC,iB;MAAA,uB;K;IAAA,mC;MAAA,sC;O;MAGI,uEAGY,GAHZ,C;MAIA
,yEAGa,MAHb,C;MAIA,yEAGa,SAHb,C;MAIA,+DAGQ,KAHR,C;MAIA,+DAGQ,MAHR,C;MAIA,2DAGM,M
AHN,C;MAIA,yDAGK,OAHL,C;K;IAxBA,gD;MAAA,yB;MAAA,wC;K;IAIA,iD;MAAA,yB;MAAA,yC;K;IAI
A,iD;MAAA,yB;MAAA,yC;K;IAIA,4C;MAAA,yB;MAAA,oC;K;IAIA,4C;MAAA,yB;MAAA,oC;K;IAIA,0C;
MAAA,yB;MAAA,kC;K;IAIA,yC;MAAA,yB;MAAA,iC;K;IA3BJ,+B;MAAA,4Q;K;IAAA,oC;MAAA,a;AAAA,
a;UAAA,6C;aAAA,c;UAAA,8C;aAAA,c;UAAA,8C;aAAA,S;UAAA,yC;aAAA,S;UAAA,yC;aAAA,O;UAAA,uC;
aAAA,M;UAAA,sC;UAAA,6D;K;IAiCA,4D;MAGW,Q;MADP,0BAA2C,iBAAjB,UAAW,cAAM,EAAU,UAA
W,cAArB,C;MAEvC,0BAAsB,CAAtB,C;QAA2B,gBAAS,UAAW,cAAX,GAAMB,UAAW,cAAvC,C;WAC3B,0B
AAsB,CAAtB,C;QAA2B,gBAAS,UAAW,cAAX,GAAMB,UAAW,cAAvC,C;QACnB,Y;MAHZ,W;K;IAOJ,oE;M
AGW,Q;MADP,0BAA2C,iBAAjB,UAAW,cAAM,EAAU,UAAW,cAArB,C;MAEvC,0BAAsB,CAAtB,C;QAA2B,
sBAA8C,uBAArC,UAAW,cAAX,GAAMB,UAAW,cAAO,CAA9C,C;WAC3B,0BAAsB,CAAtB,C;QAA2B,iBAA
8C,uBAArC,UAAW,cAAX,GAAMB,UAAW,cAAO,CAA9C,C;QACnB,Y;MAHZ,W;K;IAOJ,8D;MAGW,Q;MA
DP,0BAA2C,iBAAjB,UAAW,cAAM,EAAU,UAAW,cAArB,C;MAEvC,0BAAsB,CAAtB,C;QACI,YAAkD,uBAA
rC,UAAW,cAAX,GAAMB,UAAW,cAAO,C;QACID,aAAa,eAAQ,KAAR,C;QAET,sBAAS,KAAT,GAakB,KAAL
B,E;UAA2B,a;AAC3B,uBAAQ,CAAR,C;AIR,0BAAsB,CAAtB,C;QAA2B,iBAA8C,uBAArC,UAAW,cAAX,G
AAMB,UAAW,cAAO,CAA9C,C;QACnB,Y;MAXZ,W;K;ICrDJ,+B;MAAA,mC;MAUuB,wB;MALf,aAAR,OAA
O,OAAQ,KAAL,WAA,Y,IAAG,OAAO,SAAX,IAAwB,CAAC,CAAC,OAAO,SAAS,K;MADpE,sBAGQ,MAHR,G
AIQ,iBAAa,OAAb,CAJR,GAMQ,qBAAW,OAAX,IAAA,4GACO,+B;K;4CAIf,Y;MAAmC,OAAA,mBAAa,U;K;
;IAfpD,2C;MAAA,0C;QAAA,yB;MAAA,mC;K;IAwB2B,+B;MAAC,sB;K;IAEW,+D;MAAA,0C;MAAS,mB;M
ACxC,iBAAgB,yBAAQ,S;K;8DACxB,Y;M7HyEG,Q6HxEC,8BAAQ,QAAO,cAAP,C;MAAyB,c9IZIC,EAAI,CA
AJ,C;M8IY2C,Y9IuF3C,EAAI,CAAJ,C;M8IvFC,OAA4D,aAAR,OAAQ,qCAAR,aAAiD,aAN,KAAM,yCAAjD,
C;K;QCAH5D,Y;MAAmC,mD;K;sCAMnC,Y;MAAkC,qC;K;IAKF,4C;MAAiC,4E;MAAhC,8B;K;2CACjC,Y;M
AA8B,OAAA,gBAAY,M;K;+CAC1C,Y;MAAkC,2C;K;IAGtC,6B;MAAA,iC;MAEoC,4E;K;uCAChC,Y;MAA8B
,OAAe,U;K;2CAC7C,Y;MAAkC,+B;K;IAJtC,yC;MAAA,wC;QAAA,uB;MAAA,iC;K;IC1CA,gD;MAQ+B,kBA
ApB,wBAAC,IAAd,C;MAA0B,I9HgEjC,a;M8HhEA,O9HiEO,W;K;I8H9DX,gD;MAQqD,kBAA1B,gBAAhB,sCA
AgB,EAAc,IAAd,EAAoB,IAApB,C;MAAiC,sB9HoEID,W8HpEkD,C;MAAxD,O9HqEO,W;K;I+HzFX,yC;MAEk
D,8B;MAAA,OCGN,aDHwB,yBAAa,QAAb,mCCGxB,ChH+xBgC,sB;K;I+GhyB5E,2C;MhJggIW,kBAAY,gB;M
AoGH,Q;MAAhB,wBgJ7IIqB,UhJ6IIrB,gB;QAAGB,cgJ7IIK,UhJ6IIrB,M;QAAsB,IAAL,CgJ7IIkB,sBhJ6IIP,OgJ7II
O,ChJ6IIrB,C;UAAyB,WAA,Y,WAAI,OAAJ,C;MgJ7II3D,qBhJ8II,O,W;MgJ7IIP,I1IgNwD,C0IhNpD,c1IgNqD,U0
IhNzD,C;Q/GgKuC,U;Q+G/JnC,qB/G+JyD,OAAtB,+B+G/Jd,mB/G+Jc,uBAAsB,CAAQ,W;QuGkO7C,kBAAhB,s
B;QQ/XC,0C;QACA,IAAL,E/G8QoC,0BAAU,C+G9Q9C,CAAJ,C;UACI,2BAAO,GAAP,C;QAEW,sCAAa,GAA
b,C;QALnB,sB/H4DG,WuHoUqC,W;QQzXxC,OAAO,I;MAGX,OAAO,K;K;IAGX,8C;MAOmB,c;QhJi3YC,Q;
QAAhB,wBgJj3YI,UhJi3YJ,gB;UAAgB,cgJj3YZ,UhJi3YJ,M;UAAsB,IgJj3YD,sBhJi3Ye,OgJj3Yf,ChJi3YC,C;YA
AwB,aAAO,I;YAAP,e;;QAC9C,aAAO,K;MgJl3YP,e;QACI,kBAA6B,MAAX,UAAW,C;Q/GyIM,U;Q+GxIb,a/
GwImC,OAAtB,+B+GxIvB,mB/GwIuB,uBAAsB,CAAQ,W;Q+GxIX,kBC/BjB,aD+BD,MC/BC,ChHg1C6C,uBA
AzB,CAAyB,C;QbnmB9E,kBAAS,gB;QA2FA,U;QAAA,+B;QAAhB,OAAgB,gBAAhB,C;UAAgB,6B;UAAAM,I4
HzyB4C,4B5HyB9B,S4HzyB8B,C5HyB5C,C;YAAwB,WAA,Y,WAAI,SAAJ,C;Q4HzyBtD,sBAAMf,e5H0yBh
F,W4H1yBgF,EAAa,GAAb,C;QACnF,OAAO,I;MAGX,OAAO,K;K;IEnCP,iC;MAAQ,8BAAY,IAAK,UAAjB,IA
A8B,uBAAY,IAAK,mB;K;IAOvD,oC;MAAQ,8BAAY,IAAK,a;K;ICZ7B,4B;MAGI,OAAO,yBAAP,C;QACI,sBA
AY,mCAAZ,C;K;IAIR,uC;MAOI,sBAAY,sCAAgB,gBA Ae,IAAf,CAA5B,C;MACA,OAAO,S;K;ICbP,4B;MAA
Q,mB;K;IACR,mC;MACI,eAAO,K;K;IAKX,4B;MAAQ,mB;K;IACR,mC;MACI,eAAO,K;K;iHCofB,sJ;MAEyC,q
B;QAAA,QAakB,I;MAAM,qB;QAAA,QAakB,I;MAAM,uB;QAAA,UAAoB,K;MAAO,yB;QAAA,YAAsB,I;MA
AM,kC;QAAA,qBAA+B,I;MAAM,qC;QAAA,wBAakC,K;MAAO,+C;QAAA,kCAA4C,K;MAAO,4C;QAAA,+B
AAyC,K;MACtT,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,OAAF,IAAa,K;MACb,EAAE,SAAF,IAA
e,O;MACf,EAAE,WAAF,IAAiB,S;MACjB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,uBAAF,IAA6B,qB;MAC7
B,EAAE,iCAAF,IAAuC,+B;MACvC,EAAE,8BAAF,IAAoC,4B;MACpC,OAAO,C;K;+GAw0BX,wD;MAEwC,6B
;QAAA,gBAAYB,E;MAAL,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;M

AC/I,QAAQ,E;MACR,EAAE,eAAF,IAAqB,a;MACrB,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MA
CIB,EAAE,UAAF,IAAgB,Q;MACHB,OAAO,C;K;6EA6CX,4B;MAE6D,iBAAy,KAAZ,C;K;6EAE7D,mC;MAEo
E,UAAy,KAAZ,IAAqB,K;K;6EAuBzF,4B;MAE8D,iBAAy,KAAZ,C;K;6EAE9D,mC;MAEqE,UAAy,KAAZ,IA
AqB,K;K;6EAuB1F,4B;MAEqE,iBAAy,KAAZ,C;K;6EAErE,mC;MAE4E,UAAy,KAAZ,IAAqB,K;K;6EAuBjG,
4B;MAE+D,iBAAy,KAAZ,C;K;6EAE/D,mC;MAEsE,UAAy,KAAZ,IAAqB,K;K;6EAuB3F,4B;MAEgE,iBAAy,
KAAZ,C;K;6EAEhE,mC;MAEuE,UAAy,KAAZ,IAAqB,K;K;6EAuB5F,4B;MAE6D,iBAAy,KAAZ,C;K;6EAE7
D,mC;MAEoE,UAAy,KAAZ,IAAqB,K;K;6EAuBzF,4B;MAE8D,iBAAy,KAAZ,C;K;6EAE9D,mC;MAEqE,UAA
y,KAAZ,IAAqB,K;K;6EAuB1F,4B;MAEiE,iBAAy,KAAZ,C;K;6EAEjE,mC;MAEwE,UAAy,KAAZ,IAAqB,K;
K;6EAuB7F,4B;MAEkE,iBAAy,KAAZ,C;K;6EAEIE,mC;MAEyE,UAAy,KAAZ,IAAqB,K;K;6GC3oC9F,wD;M
AEqC,6B;QAAA,gBAA+B,I;MAAM,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WA
AqB,K;MACpJ,QAAQ,E;MACR,EAAE,eAAF,IAAqB,a;MACrB,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAA
kB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHB,OAAO,C;K;mIAiCX,+B;MAEgD,mC;QAAA,sBAAgC,K;MAC5
E,QAAQ,E;MACR,EAAE,qBAAF,IAA2B,mB;MAC3B,OAAO,C;K;4EC9CX,4B;MAEgE,iBAAy,KAAZ,C;K;4E
AgChE,4B;MAEyE,iBAAy,KAAZ,C;K;4EaiBzE,4B;MAEmE,iBAAy,KAAZ,C;K;4EayYnE,4B;MAE0E,iBAA
y,KAAZ,C;K;oIC7a1E,4H;MAE8C,qB;QAAA,QAAiB,E;MAAI,6B;QAAA,gBAAgC,E;MAAW,iC;QAAA,oBAA
2D,E;MAAW,iC;QAAA,oBAA2D,E;MAAW,qC;QAAA,wBAmjvJ,U;;MAnJqO,+B;QAAA,kBAmjro,U;;MAnJ6
S,4B;QAAA,eAA+B,S;MAC3a,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,eAAF,IAAqB,a;MACrB,E
AAE,mBAAF,IAAyB,iB;MACzB,EAAE,mBAAF,IAAyB,iB;MACzB,EAAE,uBAAF,IAA6B,qB;MAC7B,EAAE,i
BAAF,IAAuB,e;MACvB,EAAE,cAAF,IAAoB,Y;MACpB,OAAO,C;K;wIAYX,mC;MAEgD,2B;QAAA,cAAuB,E;
MAAI,0B;QAAA,aAAsB,E;MAC7F,QAAQ,E;MACR,EAAE,aAAF,IAAmB,W;MACnB,EAAE,YAAF,IAAkB,U;
MACIB,OAAO,C;K;8HakEX,+D;MAEqG,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAA
A,WAAqB,K;MAC/K,QAAQ,E;MACR,EAAE,aAAF,IAAmB,W;MACnB,EAAE,SAAF,IAAe,O;MACf,EAAE,SA
AF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHB,OAAO,C;K;4HAwBX,iE;M
AE0C,4B;QAAA,eAAwB,E;MAAI,wB;QAAA,WAAyB,I;MAAM,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAA
uB,K;MAAO,wB;QAAA,WAAqB,K;MAC/K,QAAQ,E;MACR,EAAE,cAAF,IAAoB,Y;MACpB,EAAE,UAAF,IA
AgB,Q;MACHB,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHB,
OAAO,C;K;sGAUqE,qB;MAAQ,OAAW,U;K;sGAEnB,qB;MAAQ,OAAW,U;K;4GAehB,qB;MAAQ,OAAc,a;K;
wGAS1B,qB;MAAQ,OAAy,W;K;0HAEX,qB;MAAQ,OAAqB,oB;K;kGASnD,qB;MAAQ,OAAAS,Q;K;oGAehB,q
B;MAAQ,OAAU,S;K;sGAEjB,qB;MAAQ,OAAW,U;K;wHAEV,qB;MAAQ,OAAoB,mB;K;wHAE5B,qB;MAAQ,
OAAoB,mB;K;kHAE/B,qB;MAAQ,OAAiB,gB;K;kHAEzB,qB;MAAQ,OAAiB,gB;K;oHASd,qB;MAAQ,OAAkB,
iB;K;oHAE1B,qB;MAAQ,OAAkB,iB;K;oHAE1B,qB;MAAQ,OAAkB,iB;K;wIAEHb,qB;MAAQ,OAA4B,2B;K;4
FC1MnI,uD;MAE8B,oB;QAAA,OAAgB,I;MAAM,sB;QAAA,SA Ae,C;MAAG,uB;QAAA,UAAoB,K;MAAO,0B;
QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACHJ,QAAQ,E;MACR,EAAE,MAAF,IAAY,I;MACZ,EAAE,
QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;M
ACHB,OAAO,C;K;kGAuBX,sE;MAEiC,6B;QAAA,gBAA8B,I;MAAM,oB;QAAA,OAAgB,I;MAAM,sB;QAAA,S
AAe,C;MAAG,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACvL,QAA
Q,E;MACR,EAAE,eAAF,IAAqB,a;MACrB,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,S
AAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHB,OAAO,C;K;kGA8DX,8U;
MAEiC,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,UA
AgB,C;MAAG,sB;QAAA,SAAiB,C;MAAG,uB;QAAA,UAAkB,C;MAAG,6B;QAAA,gBAA8B,I;MAAM,sB;QA
AA,SAAkB,I;MAAM,uB;QAAA,UAAoB,K;MAAO,wB;QAAA,WAAqB,K;MAAO,sB;QAAA,SAAmB,K;MAA
O,uB;QAAA,UAAoB,K;MAAO,gC;QAAA,mBAA6B,K;MAAO,gC;QAAA,mBAA6B,K;MAAO,0B;QAAA,aAA
uB,K;MAAO,8B;QAAA,iBAA2B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,+B;QAAA,kBAA4B,K;MAAO,kC;Q
AAA,qBAA+B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,8B;QAAA,iBAA2B,K;MAAO,kC;QAAA,qBAA+B,K;
MAAO,oB;QAAA,OAAgB,I;MAAM,sB;QAAA,SA Ae,C;MAAG,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAu
B,K;MAAO,wB;QAAA,WAAqB,K;MAC3wB,QAAQ,E;MACR,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,
O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,I
AAe,O;MACf,EAAE,eAAF,IAAqB,a;MACrB,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,

UAAF,IAAgB,Q;MACHB,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,eAAF,IAAqB,a;MACrB,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,eAAF,IAAqB,a;MACrB,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHB,OAAO,C;K;wGAgDX,kQ;MAEoC,uB;QAAA,UAAoB,K;MAAO,wB;QAAA,WAAqB,K;MAAO,sB;QAAA,SAAmB,K;MAAO,uB;QAAA,UAAoB,K;MAAO,gC;QAAA,mBAA6B,K;MAAO,gC;QAAA,mBAA6B,K;MAAO,0B;QAAA,aAAuB,K;MAAO,8B;QAAA,iBAA2B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,+B;QAAA,kBAA4B,K;MAAO,kC;QAAA,qBAA+B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,8B;QAAA,iBAA2B,K;MAAO,kC;QAAA,qBAA+B,K;MAAO,oB;QAAA,OAAGB,I;MAAM,sB;QAAA,SA Ae,C;MAAG,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MAC7IB,QAAQ,E;MACR,EAAE,SAAF,IAAe,O;MACf,EAAE,UAAF,IAAgB,Q;MACHB,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,eAAF,IAAqB,a;MACrB,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,eAAF,IAAqB,a;MACrB,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHB,OAAO,C;K;kGAsCX,iX;MAEiC,sB;QAAA,SAAkB,G;MAAK,sB;QAAA,SAAkB,G;MAAK,sB;QAAA,SAAkB,G;MAAK,yB;QAAA,YAAkB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,sB;QAAA,SAAiB,C;MAAG,uB;QAAA,UAAkB,C;MAAG,6B;QAAA,gBAA8B,I;MAAM,sB;QAAA,SAAkB,I;MAAM,uB;QAAA,UAAoB,K;MAAO,wB;QAAA,WAAqB,K;MAAO,sB;QAAA,SAAmB,K;MAAO,uB;QAAA,UAAoB,K;MAAO,gC;QAAA,mBAA6B,K;MAAO,gC;QAAA,mBAA6B,K;MAAO,0B;QAAA,aAAuB,K;MAAO,8B;QAAA,iBAA2B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,+B;QAAA,kBAA4B,K;MAAO,kC;QAAA,qBAA+B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,8B;QAAA,iBAA2B,K;MAAO,kC;QAAA,qBAA+B,K;MAAO,oB;QAAA,OAAGB,I;MAAM,sB;QAAA,SA Ae,C;MAAG,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACr2B,QAAQ,E;MACR,EAAE,QAAF,IAAc,M;MACd,EAAE,QAAF,IAAc,M;MACd,EAAE,QAAF,IAAc,M;MACd,EAAE,WAAAF,IAAiB,S;MACjB,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,eAAF,IAAqB,a;MACrB,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,UAAF,IAAgB,Q;MACHB,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,eAAF,IAAqB,a;MACrB,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,eAAF,IAAqB,a;MACrB,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHB,OAAO,C;K;kGA2BX,0E;MAEiC,oB;QAAA,OAAGB,E;MAAI,2B;QAAA,cAAwB,K;MAAO,oB;QAAA,OAAGB,I;MAAM,sB;QAAA,SA Ae,C;MAAG,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACtM,QAAQ,E;MACR,EAAE,MAAF,IAAY,I;MACZ,EAAE,aAAF,IAAmB,W;MACnB,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHB,OAAO,C;K;wGAmDX,4S;MAEoC,mB;QAAA,MAAe,E;MAAI,oB;QAAA,OAAGB,E;MAAI,wB;QAAA,WAAiB,C;MAAG,sB;QAAA,SAAmB,K;MAAO,2B;QAAA,cAAwB,K;MAAO,uB;QAAA,UAAoB,K;MAAO,wB;QAAA,WAAqB,K;MAAO,sB;QAAA,SAAmB,K;MAAO,uB;QAAA,UAAoB,K;MAAO,gC;QAAA,mBAA6B,K;MAAO,gC;QAAA,mBAA6B,K;MAAO,0B;QAAA,aAAuB,K;MAAO,8B;QAAA,iBAA2B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,+B;QAAA,kBAA4B,K;MAAO,kC;QAAA,qBAA+B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,8B;QAAA,iBAA2B,K;MAAO,kC;QAAA,qBAA+B,K;MAAO,oB;QAAA,OAAGB,I;MAAM,sB;QAAA,SA Ae,C;MAAG,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACjtB,QAAQ,E;MACR,EAAE,KAAF,IAAW,G;MACX,EAAE,MAAF,IAAY,I;MACZ,EAAE,UAAF,IAAgB,Q;MACHB,EAAE,QAAF,IAAc,M;MACd,EAAE,aAAF,IAAmB,W;MACnB,EAAE,SAAF,IAAe,O;MACf,EAAE,UAAF,IAAgB,Q;MACHB,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,kBAAF,IAAwB,gB;

F,IAAwB,gB;MACxB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,eAAF,IAAqB,a;MACrB,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,eAAF,IAAqB,a;MACrB,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;8GAuBX,6D;MAEuC,oB;QAAA,OAAgB,E;MAAI,oB;QAAA,OAAgB,I;MAAM,sB;QAAA,SA Ae,C;MAAG,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MAC7K,QAAQ,E;MACR,EAAE,MAAF,IAAY,I;MACZ,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;wECnbX,4B;MAEyE,iBAA Y,KAAZ,C;K;wEAEzE,2B;MAEgG,iBAA Y,IAAZ,C;K;wEAwBhG,oC;MAE+F,UAA Y,KAAZ,IAAqB,M;K;wEAmFpH,2B;MAEqE,iBAA Y,IAAZ,C;K;wEAErE,kC;MAE2E,UAA Y,IAAZ,IAAoB,K;K;wEAssC/F,4B;MAEyE,iBAA Y,KAAZ,C;K;wEA0BzE,4B;MAEyE,iBAA Y,KAAZ,C;K;wEAsBzE,4B;MAEuE,iBAA Y,KAAZ,C;K;wEAyBvE,4B;MAE6E,iBAA Y,KAAZ,C;K;2FA4C7E,gD;MAEiC,qB;QAAA,QAAiD,I;MAAM,uB;QAAA,UAAoB,K;M AAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACiK,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MAC b,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;uE A+UX,4B;MAEuE,iBAA Y,KAAZ,C;K;wEAEvE,2B;MAE6F,iBAA Y,IAAZ,C;K;wEAqN7F,4B;MAEyE,iBAA Y,KAAZ,C;K;wEAEzE,oC;MAE2F,UAA Y,KAAZ,IAAqB,M;K;+FAuehH,wD;MAEmC,6B;QAAA,gBAA8B,I;M AAM,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACjJ,QAAQ,E;MACR,EAAE,eAAF,IAAqB,a;MACrB,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAA gB,Q;MACHb,OAAO,C;K;uGAuIX,mB;MAEuC,uB;QAAA,UAAoB,K;MACvD,QAAQ,E;MACR,EAAE,SAAF,IA Ae,O;MACf,OAAO,C;K;+HAyCX,iB;MAEmD,qB;QAAA,QAAkB,I;MACjE,QAAQ,E;MACR,EAAE,OAAF,IA Aa,K;MACb,OAAO,C;K;+FA0MX,sE;MAEmC,oB;QAAA,OAAgB,I;MAAM,wB;QAAA,WA0+G4B,S;;MA1+G wB,kB;QAAA,KAAc,E;MAAI,wB;QAAA,WAAoB,I;MAAM,sB;QAAA,SAAkB,S;MAAW,uB;QAAA,UAAoB,I; MAAM,qB;QAAA,QAAiB,I;MAAM,oB;QAAA,OAAgB,I;MACnP,QAAQ,E;MACR,EAAE,MAAF,IAAY,I;MAC Z,EAAE,UAAF,IAAgB,Q;MACHb,EAAE,IAAF,IAAU,E;MACV,EAAE,UAAF,IAAgB,Q;MACHb,EAAE,QAAF, IAAC,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,OAAF,IAAa,K;MACb,EAAE,MAAF,IAAY,I;MACZ,OAAO,C;K;qIAGDX,iB;MAEsD,qB;QAAA,QAAkB,I;MACpE,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,OAAO,C; K;+GAKBX,qB;MAE2C,yB;QAAA,YAAmB,S;MAC1D,QAAQ,E;MACR,EAAE,SAAF,IAAe,S;MACf,OAAO,C; K;wEAkCX,4B;MAEqF,iBAA Y,KAAZ,C;K;yFAGCrF,4V;MAEgC,4B;QAAA,eAA8B,I;MAAM,uB;QAAA,UAA gB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,sB;QAA A,SAAiB,C;MAAG,uB;QAAA,UAAkB,C;MAAG,6B;QAAA,gBAA8B,I;MAAM,sB;QAAA,SAAkB,I;MAAM,uB ;QAAA,UAAoB,K;MAAO,wB;QAAA,WAAqB,K;MAAO,sB;QAAA,SAAmB,K;MAAO,uB;QAAA,UAAoB,K;M AAO,gC;QAAA,mBAA6B,K;MAAO,gC;QAAA,mBAA6B,K;MAAO,0B;QAAA,aAAuB,K;MAAO,8B;QAAA,iB AA2B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,+B;QAAA,kBAA4B,K;MAAO,kC;QAAA,qBAA+B,K;MAAO,6 B;QAAA,gBAA0B,K;MAAO,8B;QAAA,iBAA2B,K;MAAO,kC;QAAA,qBAA+B,K;MAAO,oB;QAAA,OAAgB,I ;MAAM,sB;QAAA,SA Ae,C;MAAG,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WA AqB,K;MAC9yB,QAAQ,E;MACR,EAAE,cAAF,IAAoB,Y;MACpB,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IA Ae,O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,QAAF,IAAc,M;MACd,EAAE,S AAF,IAAe,O;MACf,EAAE,eAAF,IAAqB,a;MACrB,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf, EAAE,UAAF,IAAgB,Q;MACHb,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,kBAAF,IAA wB,gB;MACxB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,gBAAF,IAAsB,c;M ACtB,EAAE,eAAF,IAAqB,a;MACrB,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE ,eAAF,IAAqB,a;MACrB,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,MAAF,IAA Y,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,U AAF,IAAgB,Q;MACHb,OAAO,C;K;wEAwEX,2B;MAE+D,iBAA Y,IAAZ,C;K;iGA2D/D,gD;MAEoC,qB;QAAA, QAAc,I;MAAM,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACII,QAA Q,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,U AAF,IAAgB,Q;MACHb,OAAO,C;K;qGA2BX,yD;MAEsC,sB;QAAA,SAAkB,E;MAAI,sB;QAAA,SAAkB,E;MA AI,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MAC5J,QAAQ,E;MACR,

EAAE,QAAF,IAAc,M;MACd,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MAChB,OAAO,C;K;6GAuBX,oD;MAE0C,yB;QAAA,YAAaB,K;MAAO,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACjJ,QAAQ,E;MACR,EAAE,WAAF,IAAiB,S;MACjB,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MAChB,OAAO,C;K;2FAoFX,kF;MAEiC,uB;QAAA,UAAmB,E;MAAI,wB;QAAA,WAAoB,E;MAAI,sB;QAAA,SA Ae,C;MAAG,qB;QAAA,QAAc,C;MAAG,qB;QAAA,QAAc,I;MAAM,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACjN,QAAQ,E;MACR,EAAE,SAAF,IAAe,O;MACf,EAAE,UAAF,IAAgB,Q;MAChB,EAAE,QAAF,IAAc,M;MACd,EAAE,OAAF,IAAa,K;MACb,EAAE,OAAF,IAAa,K;MACb,EA AE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MAChB,OAAO,C;K;iHAYBX ,0D;MAEqE,sB;QAAA,SA Ae,S;MAAW,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACzK,QAAQ,E;MACR,EAAE,SAAF,IAAe,O;MACf,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IA Ae,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MAChB,OAAO,C;K;wEAmXX,4B;MAE kE,iBAAY,KAAZ,C;K;wEAEIE,2B;MAEoE,iBAAY,IAAZ,C;K;wEAUpe,4B;MAEsE,iBAAY,KAAZ,C;K;wEAE tE,2B;MAEwE,iBAAY,IAAZ,C;K;wEAaxE,4B;MAE+D,iBAAY,KAAZ,C;K;wEAE/D,2B;MAEiE,iBAAY,IAAZ, C;K;mGA0CjE,8G;MAEqC,gC;QAAA,mBAooF8C,M;;MAPoFe,gC;QAAA,mBAmpFT,S;;MANpFyE,oC;QAAA, uBA8pFjE,S;;MA9pF6I,2B;QAAA,cAAoB,S;MAAW,4B;QAAA,eAAqB,S;MAAW,6B;QAAA,gBAyqFIO,K;;MA xqFvE,QAAQ,E;MACR,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,sBAAF,IA AA4B,oB;MAC5B,EAAE,aAAF,IAAmB,W;MACnB,EAAE,cAAF,IAAoB,Y;MACpB,EAAE,eAAF,IAAqB,a;MA CrB,OAAO,C;K;+FAwCX,mF;MAEmC,oB;QAAA,OAAa,I;MAAM,sB;QAAA,SAAkB,E;MAAI,2B;QAAA,cAA uB,E;MAAI,sB;QAAA,SAAYC,I;MAAM,qB;QAAA,QAA6B,E;MAAW,uB;QAAA,UAAoB,K;MAAO,0B;QAAA ,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACxQ,QAAQ,E;MACR,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF ,IAAc,M;MACd,EAAE,aAAF,IAAmB,W;MACnB,EAAE,QAAF,IAAc,M;MACd,EAAE,OAAF,IAAa,K;MACb,EA AAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MAChB,OAAO,C;K;6FA4 BX,2B;MAEkC,+B;QAAA,kBAA4B,K;MAC1D,QAAQ,E;MACR,EAAE,iBAAF,IAAuB,e;MACvB,OAAO,C;K;2 FA2DX,iE;MAEiC,wB;QAAA,WAAqB,K;MAAO,oB;QAAA,OAAe,C;MAAG,sB;QAAA,SAAkB,E;MAAI,uB;Q AAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MAC/K,QAAQ,E;MACR,EAAE,U AAF,IAAgB,Q;MAChB,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MAC f,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MAChB,OAAO,C;K;yFA8FX,6B;MAEgC,oB;QAAA, OA+7E6C,S;;MA/7EL,2B;QAAA,cCl2He,M;;MDm2HnF,QAAQ,E;MACR,EAAE,MAAF,IAAY,I;MACZ,EAAE, aAAF,IAAmB,W;MACnB,OAAO,C;K;wEAOdx,0B;MAE+D,iBAAY,GAAZ,C;K;wEAE/D,iC;MAEqE,UAAAY,G AAZ,IAAmB,K;K;+FAoDxF,oF;MAEmC,mB;QAAA,MAAe,I;MAAM,wB;QAAA,WAAoB,I;MAAM,wB;QAAA ,WAAoB,I;MAAM,mB;QAAA,MAAe,E;MAAI,2B;QAAA,cAAwB,I;MAAM,uB;QAAA,UAAoB,K;MAAO,0B;Q AAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACvO,QAAQ,E;MACR,EAAE,KAAF,IAAW,G;MACX,EAAE, UAAF,IAAgB,Q;MAChB,EAAE,UAAF,IAAgB,Q;MAChB,EAAE,KAAF,IAAW,G;MACX,EAAE,aAAF,IAAmB ,W;MACnB,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MAChB,OAA O,C;K;iFAwNX,yC;MAE4B,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K; MACtG,QAAQ,E;MACR,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q; MAChB,OAAO,C;K;6FAwBX,iD;MAEkC,sB;QAAA,SA Ae,I;MAAM,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,a AAuB,K;MAAO,wB;QAAA,WAAqB,K;MACjI,QAAQ,E;MACR,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IA Ae,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MAChB,OAAO,C;K;uGASX,mB;MAEuC, uB;QAAA,UAAoB,K;MACvD,QAAQ,E;MACR,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;6GAYX,kC;MAE0C,u B;QAAA,UAAoB,K;MAAO,oB;QAAA,OAAiB,K;MAAO,uB;QAAA,UAAoB,K;MAC7G,QAAQ,E;MACR,EAA E,SAAF,IAAe,O;MACf,EAAE,MAAF,IAAY,I;MACZ,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;wEAkEX,4B;M AE6D,iBAAY,KAAZ,C;K;wEAU7D,4B;MAEsE,iBAAY,KAAZ,C;K;wEAEtE,2B;MAEwE,iBAAY,IAAZ,C;K;u GAsCxE,oH;MAEuC,yB;QAAA,YAAaB,K;MAAO,0B;QAAA,aAAuB,S;MAAW,6B;QAAA,gBAA0B,S;MAAW, uB;QAAA,UAAoB,K;MAAO,iC;QAAA,oBAA8B,S;MAAW,qC;QAAA,wBAAkC,S;MAAW,+B;QAAA,kBAAk C,S;MAC1R,QAAQ,E;MACR,EAAE,WAAF,IAAiB,S;MACjB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,eAAF,IA AqB,a;MACrB,EAAE,SAAF,IAAe,O;MACf,EAAE,mBAAF,IAAYB,iB;MACzB,EAAE,uBAAF,IAA6B,qB;MAC

7B,EAAE,iBAAF,IAAuB,e;MACvB,OAAO,C;K;mGAgFX,oB;MAEqC,wB;QAAA,WAAqB,K;MAcTD,QAAQ,E;MACR,EAAE,UAAF,IAAgB,Q;MACHB,OAAO,C;K;wEA+MX,2B;MAEiE,iBAAY,IAAZ,C;K;2GakCjE,c;MAE yC,kB;QAAA,KAAgB,S;MACrD,QAAQ,E;MACR,EAAE,IAAF,IAAU,E;MACV,OAAO,C;K;2FAuMX,gB;MAG I,QAAQ,E;MACR,EAAE,MAAF,IAAY,I;MACZ,OAAO,C;K;wEAgBX,4B;MAEiE,iBAAY,KAAZ,C;K;wEAejE, oC;MAE4E,iBAAY,aAAZ,C;K;wEAuT5E,4B;MAEmE,iBAAY,KAAZ,C;K;uFA2CnE,sB;MAE+B,iB;QAAA,IAA a,G;MAAK,iB;QAAA,IAAa,G;MAAK,iB;QAAA,IAAa,G;MAAK,iB;QAAA,IAAa,G;MAC9F,QAAQ,E;MACR,E AAE,GAAF,IAAS,C;MACT,EAAE,GAAF,IAAS,C;MACT,EAAE,GAAF,IAAS,C;MACT,EAAE,GAAF,IAAS,C; MACT,OAAO,C;K;qFA0CX,+B;MAE8B,iB;QAAA,IAAa,G;MAAK,iB;QAAA,IAAa,G;MAAK,qB;QAAA,QAAi B,G;MAAK,sB;QAAA,SAakB,G;MACtG,QAAQ,E;MACR,EAAE,GAAF,IAAS,C;MACT,EAAE,GAAF,IAAS,C ;MACT,EAAE,OAAF,IAAa,K;MACb,EAAE,QAAF,IAAc,M;MACd,OAAO,C;K;wEAOX,4B;MAEmE,iBAAY,K AAZ,C;K;yFAiHnE,oB;MAEgC,wB;QAAA,WAY2B+C,M;MAx2B3E,QAAQ,E;MACR,EAAE,UAAF,IAAgB,Q; MACHB,OAAO,C;K;6FAeX,+B;MAEkC,oB;QAAA,OAAGB,S;MAAW,mB;QAAA,MAAe,S;MAAW,wB;QAAA, WAq1BR,M;MAp1B3E,QAAQ,E;MACR,EAAE,MAAF,IAAY,I;MACZ,EAAE,KAAF,IAAW,G;MACX,EAAE, UAAF,IAAgB,Q;MACHB,OAAO,C;K;6GAwCX,yD;MAE0C,qB;QAAA,QAAiB,E;MAAI,uB;QAAA,UAAoB,K; MAAO,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACpK,QAAQ,E;M ACR,EAAE,OAAF,IAAa,K;MACb,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAA kB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHB,OAAO,C;K;yGAiCX,mC;MAEwC,qB;QAAA,QA2wByD,Q;MA 3wBK,sB;QAAA,SA2wBL,Q;MA3wBoE,wB;QAAA,WA4vBtF,M;MA3vB3E,QAAQ,E;MACR,EAAE,OAAF,I AaA,K;MACb,EAAE,QAAF,IAAc,M;MACd,EAAE,UAAF,IAAgB,Q;MACHB,OAAO,C;K;2FAYX,2B;MAEiC,m B;QAAA,MAuwB0C,Q;MAvwBJ,0B;QAAA,aAAsB,S;MACzF,QAAQ,E;MACR,EAAE,KAAF,IAAW,G;MACX ,EAAE,YAAF,IAAkB,U;MACIB,OAAO,C;K;+GAYX,0B;MAE2C,uB;QAAA,UaqvBgC,Q;MArvBU,qB;QAAA, QaqvBV,Q;MApvBvE,QAAQ,E;MACR,EAAE,SAAF,IAAe,O;MACf,EAAE,OAAF,IAAa,K;MACb,OAAO,C;K; wEAgCX,4B;MAE+D,iBAAY,KAAZ,C;K;qFAyaY,qB;MAAQ,OAAU,S;K;6FAEd,qB;MAAQ,OAAc,a;K;uFAEz B,qB;MAAQ,OAAW,U;K;iFASxB,qB;MAAQ,OAAG,E;K;iFAEX,qB;MAAQ,OAAQ,O;K;uFAEb,qB;MAAQ,OA AW,U;K;uFAS3B,qB;MAAQ,OAAW,U;K;mFAErB,qB;MAAQ,OAAS,Q;K;qFAEhB,qB;MAAQ,OAAU,S;K;yFA ShB,qB;MAAQ,OAAy,W;K;uFAErB,qB;MAAQ,OAAW,U;K;+FAEf,qB;MAAQ,OAAe,c;K;uFAE3B,qB;MAAQ, OAAW,U;K;uFAEnB,qB;MAAQ,OAAW,U;K;mFASrB,qB;MAAQ,OAAS,Q;K;iFAEiB,qB;MAAQ,OAAQ,O;K;6 EAEiB,qB;MAAQ,OAAM,K;K;uFAET,qB;MAAQ,OAAW,U;K;qFASiB,qB;MAAQ,OAAU,S;K;qFAEiB,qB;MA AQ,OAAU,S;K;6EASr,qB;MAAQ,OAAM,K;K;mFAEX,qB;MAAQ,OAAS,Q;K;+EAEnB,qB;MAAQ,OAAO,M; K;+EAS/B,qB;MAAQ,OAAO,M;K;iFAEd,qB;MAAQ,OAAQ,O;K;mFAEf,qB;MAAQ,OAAS,Q;K;mFAShB,qB; MAAQ,OAAQ,O;K;iFAEhB,qB;MAAQ,OAAQ,O;K;iFAEhB,qB;MAAQ,OAAQ,O;K;mFASd,qB;MAAQ,OAAQ, O;K;+EAEiB,qB;MAAQ,OAAM,K;K;+EAEb,qB;MAAQ,OAAO,M;K;iFAEd,qB;MAAQ,OAAQ,O;K;mFAEf,qB; MAAQ,OAAS,Q;K;6EASd,qB;MAAQ,OAAM,K;K;qFAEV,qB;MAAQ,OAAU,S;K;mFAEnB,qB;MAAQ,OAAS, Q;K;2FAEb,qB;MAAQ,OAAa,Y;K;6FAEpB,qB;MAAQ,OAAc,a;K;mFAE3B,qB;MAAQ,OAAS,Q;K;6EAS1B,qB ;MAAQ,OAAM,K;K;6EAEd,qB;MAAQ,OAAM,K;K;qFAEV,qB;MAAQ,OAAU,S;K;+EASjB,qB;MAAQ,OAAO, M;K;mFAEb,qB;MAAQ,OAAS,Q;K;+EASrB,qB;MAAQ,OAAO,M;K;iFAEd,qB;MAAQ,OAAQ,O;K;iFASjB,qB; MAAQ,OAAO,M;K;6FAER,qB;MAAQ,OAAc,a;K;qFAE1B,qB;MAAQ,OAAU,S;K;iFASb,qB;MAAQ,OAAO,M; K;uFAEZ,qB;MAAQ,OAAU,S;K;yFAS9B,qB;MAAQ,OAAy,W;K;+EAE1B,qB;MAAQ,OAAM,K;K;qFAEX,qB; MAAQ,OAAS,Q;K;iFAEnB,qB;MAAQ,OAAO,M;K;+EASrB,qB;MAAQ,OAAO,M;K;6FAER,qB;MAAQ,OAAc, a;K;qFAS1B,qB;MAAQ,OAAU,S;K;mFAEnB,qB;MAAQ,OAAS,Q;K;+EASX,qB;MAAQ,OAAO,M;K;mFAEb,q B;MAAQ,OAAS,Q;K;iFASnB,qB;MAAQ,OAAO,M;K;qFAEZ,qB;MAAQ,OAAU,S;K;mFAEnB,qB;MAAQ,OAA S,Q;K;kFASj,qB;MAAQ,OAAQ,O;K;oFAEf,qB;MAAQ,OAAS,Q;K;8EAEpB,qB;MAAQ,OAAM,K;K;oFAEV,q B;MAAQ,OAAU,S;K;mFASzC,qB;MAAQ,OAAS,Q;K;mFAEjB,qB;MAAQ,OAAS,Q;K;qFAEhB,qB;MAAQ,OA AU,S;K;qFAEiB,qB;MAAQ,OAAU,S;K;wIEx+M7E,wM;MAEiD,qB;QAAA,QAakB,I;MAAM,sB;QAAA,SAAM B,I;MAAM,2B;QAAA,cAAwB,I;MAAM,yB;QAAA,YAAsB,I;MAAM,0B;QAAA,aAAuB,I;MAAM,0B;QAAA,a AAuB,I;MAAM,sB;QAAA,SAAM,B,I;MAAM,0B;QAAA,aAAuB,I;MAAM,0B;QAAA,aAAuB,I;MAAM,gC;QAA A,mBAA6B,I;MAAM,+B;QAAA,kBAA4B,I;MAAM,gC;QAAA,mBAA6B,I;MAAM,uB;QAAA,UAAoB,I;MAA M,4B;QAAA,eAAyB,I;MAAM,wB;QAAA,WAAqB,I;MAAM,uB;QAAA,UAAoB,I;MACrF,QAAQ,E;MACR,EA

AE,OAAF,IAAa,K;MACb,EAAE,QAAF,IAAc,M;MACd,EAAE,aAAF,IAAmB,W;MACnB,EAAE,WAAF,IAAiB,S;MACjB,EAAE,YAAF,IAAkB,U;MACiB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,QAAF,IAAc,M;MACd,EAAE,YAAF,IAAkB,U;MACIB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,SAAF,IAAe,O;MACf,EAAE,cAAF,IAAoB,Y;MACpB,EAAE,UAAF,IAAgB,Q;MACHb,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;wHAsDX,wM;MAEyC,qB;QAAA,QAAqB,S;MAAW,sB;QAAA,SAAsB,S;MAAW,2B;QAAA,cAA4B,S;MAAW,yB;QAAA,YAA0B,S;MAAW,0B;QAAA,aAA6B,S;MAAW,0B;QAAA,aAA6B,S;MAAW,sB;QAAA,SAAuB,S;MAAW,0B;QAAA,aAA0B,S;MAAW,0B;QAAA,aAA0B,S;MAAW,gC;QAAA,mBAAoC,S;MAAW,+B;QAAA,kBAAmC,S;MAAW,gC;QAAA,mBAAoC,S;MAAW,uB;QAAA,UAAwB,S;MAAW,4B;QAAA,eAA4B,S;MAAW,wB;QAAA,WAAoB,S;MAAW,uB;QAAA,UAAmB,S;MACtnB,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,QAAF,IAAc,M;MACd,EAAE,aAAF,IAAmB,W;MACnB,EAAE,WAAF,IAAiB,S;MACjB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,QAAF,IAAc,M;MACd,EAAE,YAAF,IAAkB,U;MACIB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,SAAF,IAAe,O;MACf,EAAE,cAAF,IAAoB,Y;MACpB,EAAE,UAAF,IAAgB,Q;MACHb,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;sHAYX,kN;MAEwC,wB;QAAA,WAA4C,S;MAAW,qB;QAAA,QAAiB,S;MAAW,sB;QAAA,SAAkB,S;MAAW,2B;QAAA,cAAuB,S;MAAW,yB;QAAA,YAAqB,S;MAAW,0B;QAAA,aAAsB,S;MAAW,0B;QAAA,aAAsB,S;MAAW,sB;QAAA,SAAkB,S;MAAW,0B;QAAA,aAAsB,S;MAAW,0B;QAAA,aAAsB,S;MAAW,gC;QAAA,mBAA4B,S;MAAW,+B;QAAA,kBAA2B,S;MAAW,gC;QAAA,mBAA4B,S;MAAW,uB;QAAA,UAAmB,S;MAAW,4B;QAAA,eAAwB,S;MAAW,wB;QAAA,WAAoB,S;MAAW,uB;QAAA,UAAmB,S;MAC9IB,QAAQ,E;MACR,EAAE,UAAF,IAAgB,Q;MACHb,EAAE,OAAF,IAAa,K;MACb,EAAE,QAAF,IAAc,M;MACd,EAAE,aAAF,IAAmB,W;MACnB,EAAE,WAAF,IAAiB,S;MACjB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,QAAF,IAAc,M;MACd,EAAE,YAAF,IAAkB,U;MACIB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,SAAF,IAAe,O;MACf,EAAE,cAAF,IAAoB,Y;MACpB,EAAE,UAAF,IAAgB,Q;MACHb,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;0HAsDX,wM;MAE0C,qB;QAAA,QAAiB,S;MAAW,sB;QAAA,SAAkB,S;MAAW,2B;QAAA,cAAuB,S;MAAW,yB;QAAA,YAAqB,S;MAAW,0B;QAAA,aAAsB,S;MAAW,0B;QAAA,aAAsB,S;MAAW,sB;QAAA,SAAkB,S;MAAW,0B;QAAA,aAAsB,S;MAAW,0B;QAAA,aAAsB,S;MAAW,gC;QAAA,mBAA4B,S;MAAW,+B;QAAA,kBAA2B,S;MAAW,gC;QAAA,mBAA4B,S;MAAW,uB;QAAA,UAAmB,S;MAAW,4B;QAAA,eAAwB,S;MAAW,wB;QAAA,WAAoB,S;MAAW,uB;QAAA,UAAmB,S;MACziB,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,QAAF,IAAc,M;MACd,EAAE,aAAF,IAAmB,W;MACnB,EAAE,WAAF,IAAiB,S;MACjB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,QAAF,IAAc,M;MACd,EAAE,YAAF,IAAkB,U;MACIB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,SAAF,IAAe,O;MACf,EAAE,cAAF,IAAoB,Y;MACpB,EAAE,UAAF,IAAgB,Q;MACHb,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;gHAYDX,wM;MAEqC,qB;QAAA,QAAc,S;MAAW,sB;QAAA,SAAsB,S;MAAW,2B;QAAA,cAAuB,S;MAAW,yB;QAAA,YAAqB,S;MAAW,0B;QAAA,aAAsB,S;MAAW,0B;QAAA,aAAsB,S;MAAW,sB;QAAA,SAAkB,S;MAAW,0B;QAAA,aAAmB,S;MAAW,0B;QAAA,aAAmB,S;MAAW,gC;QAAA,mBAA6B,S;MAAW,+B;QAAA,kBAA4B,S;MAAW,gC;QAAA,mBAA6B,S;MAAW,uB;QAAA,UAAmB,S;MAAW,4B;QAAA,eAAqB,S;MAAW,wB;QAAA,WAAoB,S;MAAW,uB;QAAA,UAAmB,S;MACxB,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,QAAF,IAAc,M;MACd,EAAE,aAAF,IAAmB,W;MACnB,EAAE,WAAF,IAAiB,S;MACjB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,QAAF,IAAc,M;MACd,EAAE,YAAF,IAAkB,U;MACIB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,SAAF,IAAe,O;MACf,EAAE,cAAF,IAAoB,Y;MACpB,EAAE,UAAF,IAAgB,Q;MACHb,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;8HAqBX,gD;MAEsE,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACHJ,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;sIAoBX,gD;MAEgD,qB;QAAA,QAAiB,I;MAAM,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACjJ,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHb,OAAO,C;K;wHAWCX,wB;

MAEyC,qB;QAAA,QAAiB,K;MAAO,qB;QAAA,QAAiB,K;MAC9E,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MA
Cb,EAAE,OAAF,IAAa,K;MACb,OAAO,C;K;kGAYBX,oB;MAE8B,mB;QAAA,MAAe,S;MAAW,mB;QAAA,MA
Ae,S;MACnE,QAAQ,E;MACR,EAAE,KAAF,IAAW,G;MACX,EAAE,KAAF,IAAW,G;MACX,OAAO,C;K;oHA
YX,kC;MAEuC,qB;QAAA,QAAiB,S;MAAW,qB;QAAA,QAAiB,S;MAAW,mB;QAAA,MAAe,S;MAAW,mB;Q
AAA,MAAe,S;MACpI,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,OAAF,IAAa,K;MACb,EAAE,KAA
F,IAAW,G;MACX,EAAE,KAAF,IAAW,G;MACX,OAAO,C;K;gGAYX,oB;MAE6B,mB;QAAA,MAAY,S;MAA
W,mB;QAAA,MAAY,S;MAC5D,QAAQ,E;MACR,EAAE,KAAF,IAAW,G;MACX,EAAE,KAAF,IAAW,G;MAC
X,OAAO,C;K;kHAYX,kC;MAEsC,qB;QAAA,QAAc,S;MAAW,qB;QAAA,QAAc,S;MAAW,mB;QAAA,MAAY,
S;MAAW,mB;QAAA,MAAY,S;MACvH,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,OAAF,IAAa,K;
MACb,EAAE,KAAF,IAAW,G;MACX,EAAE,KAAF,IAAW,G;MACX,OAAO,C;K;gIAeX,wB;MAE6C,qB;QAA
A,QAakB,S;MAAW,qB;QAAA,QAakB,S;MACxF,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,OAAF
,IAAa,K;MACb,OAAO,C;K;oIAeX,wB;MAE+C,qB;QAAA,QAAiB,S;MAAW,qB;QAAA,QAAiB,S;MACxF,QA
AQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,OAAF,IAAa,K;MACb,OAAO,C;K;4FAKX,Y;MAGI,QAAQ,E
;MACR,OAAO,C;K;oFAKX,Y;MAGI,QAAQ,E;MACR,OAAO,C;K;8FAKX,Y;MAGI,QAAQ,E;MACR,OAAO,C
;K;kGASX,oB;MAE8B,wB;QAAA,WAAkC,S;MAC5D,QAAQ,E;MACR,EAAE,UAAF,IAAgB,Q;MACHB,OAA
O,C;K;4FAUmE,qB;MAAQ,OAAO,M;K;8FAEd,qB;MAAQ,OAAQ,O;K;4FASrB,qB;MAAQ,OAAO,M;K;0GAE
R,qB;MAAQ,OAAc,a;K;8FAE7B,qB;MAAQ,OAAO,M;K;gGAEd,qB;MAAQ,OAAQ,O;K;8FASjB,qB;MAAQ,O
AAO,M;K;gHAEL,qB;MAAQ,OAAiB,gB;K;wGASrC,qB;MAAQ,OAAa,Y;K;0GAEpB,qB;MAAQ,OAAc,a;K;w
GAEvB,qB;MAAQ,OAAa,Y;K;oFCroB7F,4B;MAE6E,iBAAY,KAAZ,C;K;iGASnB,qB;MAAQ,OAAS,Q;K;6FAE
nB,qB;MAAQ,OAAO,M;K;+FAEd,qB;MAAQ,OAAQ,O;K;iGASF,qB;MAAQ,OAAU,S;K;+FAEnB,qB;MAAQ,O
AAS,Q;K;mGAS3B,qB;MAAQ,OAAW,U;K;mGAEnB,qB;MAAQ,OAAW,U;K;6GC1D/E,mb;MAEmC,yB;QAA
A,YAAkB,C;MAAG,qB;QAAA,QAAiB,G;MAAK,sB;QAAA,SAakB,G;MAAK,wB;QAAA,WAAmB,G;MAAI,k
C;QAAA,qBAA6B,G;MAAI,qB;QAAA,QAAc,C;MAAG,qB;QAAA,QAAc,C;MAAG,qB;QAAA,QAAc,C;MAA
G,2B;QAAA,cAAuB,E;MAAI,yB;QAAA,YAAsB,K;MAAO,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,UAAgB,C;
MAAG,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,sB;QAAA,SAAiB,C;MAAG,uB;QAAA,UA
AkB,C;MAAG,6B;QAAA,gBAA8B,I;MAAM,sB;QAAA,SAakB,I;MAAM,uB;QAAA,UAAoB,K;MAAO,wB;QA
AA,WAAqB,K;MAAO,sB;QAAA,SAAmB,K;MAAO,uB;QAAA,UAAoB,K;MAAO,gC;QAAA,mBAA6B,K;MA
AO,gC;QAAA,mBAA6B,K;MAAO,0B;QAAA,aAAuB,K;MAAO,8B;QAAA,iBAA2B,K;MAAO,6B;QAAA,gBA
A0B,K;MAAO,+B;QAAA,kBAA4B,K;MAAO,kC;QAAA,qBAA+B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,8B;
QAAA,iBAA2B,K;MAAO,kC;QAAA,qBAA+B,K;MAAO,oB;QAAA,OAAGB,I;MAAM,sB;QAAA,SAAE,C;MA
AG,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACl/B,QAAQ,E;MACR
,EAAE,WAAF,IAAiB,S;MACjB,EAAE,OAAF,IAAa,K;MACb,EAAE,QAAF,IAAc,M;MACd,EAAE,UAAF,IAAg
B,Q;MACHB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,OAAF,IAAa,K;MACb,EAAE,OAAF,IAAa,K;MACb,EA
AE,OAAF,IAAa,K;MACb,EAAE,aAAF,IAAmB,W;MACnB,EAAE,WAAF,IAAiB,S;MACjB,EAAE,SAAF,IAAe,
O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,QAAF,I
AAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,eAAF,IAAqB,a;MACrB,EAAE,QAAF,IAAc,M;MACd,EAAE
,SAAF,IAAe,O;MACf,EAAE,UAAF,IAAgB,Q;MACHB,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MA
Cf,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,YAAF,IAAkB,U;MACIB,EAA
E,gBAAF,IAAsB,c;MACtB,EAAE,eAAF,IAAqB,a;MACrB,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,oBAAF,IAA
0B,kB;MAC1B,EAAE,eAAF,IAAqB,a;MACrB,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,oBAAF,IAA0B,kB;MAC
1B,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAk
B,U;MACIB,EAAE,UAAF,IAAgB,Q;MACHB,OAAO,C;K;6GC1BX,0C;MAEwC,oB;QAAA,OAAiB,I;MAAM,sB
;QAAA,SAAmB,K;MAAO,uB;QAAA,UAAoB,K;MAAO,uB;QAAA,UAAoB,K;MACpI,QAAQ,E;MACR,EAAE,
MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,
OAAO,C;K;4EAmIX,4B;MAEkE,iBAAY,KAAZ,C;K;4EAEIE,qC;MAE2E,UAAAY,KAAZ,IAAqB,O;K;4EAIbhG,
4B;MAEuE,iBAAY,KAAZ,C;K;4EAEvE,qC;MAE+E,UAAAY,KAAZ,IAAqB,O;K;4EAIbPg,4B;MAEuE,iBAAY,
KAAZ,C;K;4EAEvE,qC;MAE+E,UAAAY,KAAZ,IAAqB,O;K;4EAIgP,4B;MAEoE,iBAAY,KAAZ,C;K;2EAEpE,
qC;MAE4E,UAAAY,KAAZ,IAAqB,O;K;4EAKcJ,4B;MAE6E,iBAAY,KAAZ,C;K;4EAE7E,qC;MAE4F,UAAAY,K

AAZ,IAAqB,O;K;4EAqP1G,4B;MAEqE,iBAAY,KAAZ,C;K;4EAERe,qC;MAE6E,UAAy,KAAZ,IAAqB,O;K;uFJ
57BIG,+H;MAE8B,sB;QAAA,SAakB,S;MAAW,uB;QAAA,UAAmB,S;MAAW,oB;QAAA,OAAgB,S;MAAW,w
B;QAAA,WAAoB,S;MAAW,8B;QAAA,iBAA0B,S;MAAW,oB;QAAA,OAAqB,S;MAAW,2B;QAAA,cAAmC,S;
MAAW,qB;QAAA,QAAuB,S;MAAW,wB;QAAA,WAA6B,S;MAAW,yB;QAAA,YAAqB,S;MAAW,yB;QAAA,
YAAsB,S;MAAW,wB;QAAA,WAAe,S;MAC5Z,QAAQ,E;MACR,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IA
Ae,O;MACf,EAAE,MAAF,IAAY,I;MACZ,EAAE,UAAF,IAAgB,Q;MACHB,EAAE,gBAAF,IAAsB,c;MACtB,EA
AE,MAAF,IAAY,I;MACZ,EAAE,aAAF,IAAmB,W;MACnB,EAAE,OAAF,IAAa,K;MACb,EAAE,UAAF,IAAgB,
Q;MACHB,EAAE,WAAF,IAAiB,S;MACjB,EAAE,WAAF,IAAiB,S;MACjB,EAAE,QAAF,IAAc,Q;MACd,OAAO
,C;K;yFA0CX,uC;MAE+B,sB;QAAA,SAAiB,G;MAAK,0B;QAAA,aAAsB,I;MAAM,uB;QAAA,UAAmB,S;MAC
hG,QAAQ,E;MACR,EAAE,QAAF,IAAc,M;MACd,EAAE,YAAF,IAAkB,U;MACIB,EAAE,SAAF,IAAe,O;MACf
,OAAO,C;K;qFAUgD,qB;MAAQ,OAAQ,E;K;mFAEX,qB;MAAQ,OAAQ,O;K;iFAEjB,qB;MAAQ,OAAO,M;K;
mFAEd,qB;MAAQ,OAAQ,O;K;qFAEf,qB;MAAQ,OAAS,Q;K;mFAElB,qB;MAAQ,OAAQ,O;K;mFAEhB,qB;M
AAQ,OAAQ,O;K;mFAEhB,qB;MAAQ,OAAQ,O;K;qFASF,qB;MAAQ,OAAG,E;K;yFAER,qB;MAAQ,OAAW,U
;K;mFAEtB,qB;MAAQ,OAAQ,O;K;mFAEjB,qB;MAAQ,OAAO,M;K;qFAEd,qB;MAAQ,OAAQ,O;K;yFAEb,qB;
MAAQ,OAAW,U;K;mFAEtB,qB;MAAQ,OAAQ,O;K;qFAEf,qB;MAAQ,OAAS,Q;K;qFAEjB,qB;MAAQ,OAAS,
Q;K;uFAEjB,qB;MAAQ,OAAS,Q;K;mGAEV,qB;MAAQ,OAAgB,e;K;iGAEzB,qB;MAAQ,OAAe,c;K;qFAE9B,q
B;MAAQ,OAAQ,O;K;qFAEf,qB;MAAQ,OAAS,Q;K;iFAEnB,qB;MAAQ,OAAO,M;K;yFASzB,qB;MAAQ,OAA
W,U;K;+FAEhB,qB;MAAQ,OAAc,a;K;uFAE1B,qB;MAAQ,OAAU,S;K;iFAErB,qB;MAAQ,OAAO,M;K;iFASD,
qB;MAAQ,OAAO,M;K;iGAER,qB;MAAQ,OAAc,a;K;uFAE1B,qB;MAAQ,OAAU,S;K;yFAS9B,qB;MAAQ,OA
AU,S;K;yFAEjB,qB;MAAQ,OAAW,U;K;qFAErB,qB;MAAQ,OAAS,Q;K;yFAEf,qB;MAAQ,OAAW,U;K;+FAEh
B,qB;MAAQ,OAAc,a;K;qGAEnB,qB;MAAQ,OAAiB,gB;K;qFAS3B,qB;MAAQ,OAAS,Q;K;mFAElB,qB;MAAQ
,OAAQ,O;K;uFAEf,qB;MAAQ,OAAS,Q;K;mFASxB,qB;MAAQ,OAAQ,O;K;mFAEjB,qB;MAAQ,OAAO,M;K;y
FAEZ,qB;MAAQ,OAAU,S;K;qFAEpB,qB;MAAQ,OAAQ,O;K;qFAEf,qB;MAAQ,OAAS,Q;K;qGAET,qB;MAAQ
,OAAiB,gB;K;+FKnR/F,gB;MAEkC,oB;QAAA,OAAgB,E;MAC9C,QAAQ,E;MACR,EAAE,MAAF,IAAY,I;MA
CZ,OAAO,C;K;+FAiBX,8B;MAEkC,4B;QAAA,eAAqB,S;MAAW,oB;QAAA,OAAgB,E;MAC9E,QAAQ,E;MAC
R,EAAE,cAAF,IAAoB,Y;MACpB,EAAE,MAAF,IAAY,I;MACZ,OAAO,C;K;0EAUX,4B;MAE6D,iBAAY,KAA
Z,C;K;+GC6B7D,sJ;MAEsC,mB;QAAA,MA4GuD,M;MA5GG,oB;QAAA,OAAgB,E;MAAI,oB;QAAA,OAAgB,
E;MAAI,mB;QAAA,MAAe,E;MAAI,qB;QAAA,QAAiB,S;MAAW,oB;QAAA,OAAgB,S;MAAW,qB;QAAA,QA
AiB,S;MAAW,qB;QAAA,QAAiB,S;MAAW,uB;QAAA,UAAmB,S;MAAW,yB;QAAA,YAAqB,S;MAAW,wB;Q
AAA,WAAqB,K;MAAO,sB;QAAA,SAAmB,K;MAAO,wB;QAAA,WAAqB,K;MAAO,kC;QAAA,qBAA+B,K;M
AAO,sB;QAAA,SAAmB,K;MAAO,oB;QAAA,OAAa,I;MAAM,uB;QAAA,UAAc,E;MAC/gB,QAAQ,E;MACR,
EAAE,KAAF,IAAW,G;MACX,EAAE,MAAF,IAAY,I;MACZ,EAAE,MAAF,IAAY,I;MACZ,EAAE,KAAF,IAA
W,G;MACX,EAAE,OAAF,IAAa,K;MACb,EAAE,MAAF,IAAY,I;MACZ,EAAE,OAAF,IAAa,K;MACb,EAAE,O
AAF,IAAa,K;MACb,EAAE,SAAF,IAAe,O;MACf,EAAE,WAAF,IAAiB,S;MACjB,EAAE,UAAF,IAAgB,Q;MAC
hB,EAAE,QAAF,IAAc,M;MACd,EAAE,UAAF,IAAgB,Q;MACHB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,QA
AF,IAAc,M;MACd,EAAE,MAAF,IAAY,I;MACZ,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;6GAWX,+B;MAEsE,
oB;QAAA,OAAgB,S;MACIF,QAAQ,E;MACR,EAAE,QAAF,IAAc,M;MACd,EAAE,OAAF,IAAa,K;MACb,EAA
E,MAAF,IAAY,I;MACZ,OAAO,C;K;qHASX,e;MAEyC,mB;QAAA,MAAe,E;MACpD,QAAQ,E;MACR,EAAE,
KAAF,IAAW,G;MACX,OAAO,C;K;mHAYBX,+D;MAEqE,sB;QAAA,SAakB,E;MAAI,uB;QAAA,UAAoB,K;M
AAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACrK,QAAQ,E;MACR,EAAE,cAAF,IAAoB,Y;MA
CpB,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,I
AAGB,Q;MACHB,OAAO,C;K;iGAUwE,qB;MAAQ,OAAU,S;K;6FAEnB,qB;MAAQ,OAAS,Q;K;+FAEhB,qB;M
AAQ,OAAU,S;K;2FASvB,qB;MAAQ,OAAO,M;K;yFAEhB,qB;MAAQ,OAAM,K;K;yFAEd,qB;MAAQ,OAAM,
K;K;yGCrJ3F,uB;MAEsC,qB;QAAA,QAAiB,S;MAAW,oB;QAAA,ORy9MW,S;MQx9MzE,QAAQ,E;MACR,EA
AE,OAAF,IAAa,K;MACb,EAAE,MAAF,IAAY,I;MACZ,OAAO,C;K;6HAUCX,mF;MAEgD,oB;QAAA,OAAa,S;
MAAW,sB;QAAA,SAakB,S;MAAW,2B;QAAA,cAAuB,S;MAAW,sB;QAAA,SA2C,S;MAAW,qB;QAAA,QA
A6B,S;MAAW,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MAC/S,QAA
Q,E;MACR,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,aAAF,IAAmB,W;MACnB,EAA

E,QAAF,IAAc,M;MACd,EAAE,OAAF,IAAa,K;MACb,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MAChB,OAAO,C;K;uGA2DX,qC;MAEqC,mC;QAAA,sBAAGC,K;MAAO,oB;QAAA,OA4UD,Q;MA3UvE,QAAQ,E;MACR,EAAE,qBAAF,IAA2B,mB;MAC3B,EAAE,MAAF,IAAY,I;MACZ,OA AO,C;K;yGAmBX,yC;MAEsC,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB, K;MACHH,QAAQ,E;MACR,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB, Q;MAChB,OAAO,C;K;yGAsBX,2B;MAGI,QAAQ,E;MACR,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe, O;MACf,OAAO,C;K;+FA8BX,sE;MAEoD,wB;QAAA,WAAoB,I;MAAM,wB;QAAA,WAAqB,K;MAAO,uB;QA AA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACpL,QAAQ,E;MACR,EAAE,SA AAF,IAAe,O;MACf,EAAE,UAAF,IAAgB,Q;MAChB,EAAE,UAAF,IAAgB,Q;MAChB,EAAE,SAAF,IAAe,O;MA Cf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MAChB,OAAO,C;K;6GAuBX,0D;MAE2D,sB;QAA A,SAAkB,M;MAAQ,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MAC/J, QAAQ,E;MACR,EAAE,SAAF,IAAe,O;MACf,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE, YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MAChB,OAAO,C;K;2GAaX,qC;MAE4D,sB;QAAA,SAAkB,S; MAAW,uB;QAAA,UAAoB,S;MAC/G,QAAQ,E;MACR,EAAE,UAAF,IAAgB,Q;MAChB,EAAE,QAAF,IAAc,M; MACd,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;uHAuCX,mF;MAE6C,oB;QAAA,OAAa,S;MAAW,sB;QAAA,S AAkB,S;MAAW,2B;QAAA,cAAuB,S;MAAW,sB;QAAA,SAAmD,S;MAAW,qB;QAAA,QAA6B,S;MAAW,uB;Q AAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACpT,QAAQ,E;MACR,EAAE,M AAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,aAAF,IAAmB,W;MACnB,EAAE,QAAF,IAAc,M;MA Cd,EAAE,OAAF,IAAa,K;MACb,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IA AgB,Q;MAChB,OAAO,C;K;qGA+BX,6D;MAEoC,4B;QAAA,eAAyB,K;MAAO,4B;QAAA,eAAyB,K;MAAO,0B ;QAAA,aAAuB,K;MAAO,yB;QAAA,YAAqB,S;MACnJ,QAAQ,E;MACR,EAAE,cAAF,IAAoB,Y;MACpB,EAA E,cAAF,IAAoB,Y;MACpB,EAAE,YAAF,IAAkB,U;MACIB,EAAE,WAAF,IAAiB,S;MACjB,OAAO,C;K;yGAKB X,4C;MAEsC,oB;QAAA,OAAGB,S;MAAW,uB;QAAA,UAAoB,S;MAAW,wB;QAAA,WAAsB,S;MAAW,uB;QA AA,UAA8B,S;MAC3J,QAAQ,E;MACR,EAAE,MAAF,IAAY,I;MACZ,EAAE,SAAF,IAAe,O;MACf,EAAE,UAA F,IAAgB,Q;MAChB,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;+FAkCmE,qB;MAAQ,OAAa,Y;K;6FAEtB,qB;MA AQ,OAAy,W;K;+FAEnB,qB;MAAQ,OAAa,Y;K;6FAEtB,qB;MAAQ,OAAy,W;K;6FAEpB,qB;MAAQ,OAAy, W;K;6FAStC,qB;MAAQ,OAAy,W;K;6FAEpB,qB;MAAQ,OAAy,W;K;uFAEvB,qB;MAAQ,OAAS,Q;K;qFAEn B,qB;MAAQ,OAAO,M;K;uFASX,qB;MAAQ,OAAS,Q;K;yFAEjB,qB;MAAQ,OAAS,Q;K;qGAEX,qB;MAAQ,O AAe,c;K;iFAEhC,qB;MAAQ,OAAM,K;K;iGCharE,0E;MAEoC,gC;QAAA,mBAA6B,K;MAAO,sB;QAAA,SAAk B,C;MAAG,qB;QAAA,QAAiB,C;MAAG,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA ,WAAqB,K;MAC3L,QAAQ,E;MACR,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,QAAF,IAAc,M;MACd,EAAE, OAAF,IAAa,K;MACb,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACIB,EAAE,UAAF,IAAgB,Q;MA ChB,OAAO,C;K;mFAU8E,qB;MAAQ,OAAG,E;K;+FAEL,qB;MAAQ,OAAC,a;K;iFAE7B,qB;MAAQ,OAAO,M; K;yFAEX,qB;MAAQ,OAAW,U;K;+EAEvB,qB;MAAQ,OAAO,M;K;+EAEf,qB;MAAQ,OAAO,M;K;oEtJlvG,yB ;MAAA,kF;MAAA,0B;MAAA,uB;QAaI,IAAI,OAAO,CAAP,IAA8B,OAAO,KAAzC,C;UACI,MAAM,8BAAYB, wBAAqB,IAA9C,C;;QAEV,OAAy,OAAL,IAAK,C;O;KAhBhB,C;0EAwCiC,qB;MAAQ,OAAA,SAAK,I;K;IuInB V,6B;MAAC,qB;QAAA,8C;MAAA,kB;K;IACjC,2C;MAAA,e;MAAA,iB;MAAA,uB;K;IAAA,yC;MAAA,4C;O; MAKI,0E;MAEA,sE;K;;IAFA,kD;MAAA,+B;MAAA,0C;K;;IAEA,gD;MAAA,+B;MAAA,wC;K;;IAPJ,qC;MAA A,yF;K;;IAAA,0C;MAAA,a;AAAA,S;UAAA,+C;AAAA,O;UAAA,6C;;UAAA,8D;;K;;IA0BmC,sC;MACnC,8B;K ;IAMqC,sC;MACrC,8B;K;;IC5DJ,iC;K;;ICMA,4B;K;;IA6BA,gD;K;;IC5BA,qC;K;;IA0BA,+B;K;;ICnqC,uC;MA CjC,uB;QAAA,UAAsB,E;MACtB,qB;QAAA,+C;MADA,sB;MACA,kB;K;IAEA,4C;MAAA,e;MAAA,iB;MAAA, uB;K;IAAA,0C;MAAA,6C;O;MAKI,4E;MAGA,wE;K;;IAHA,mD;MAAA,gC;MAAA,2C;K;;IAGA,iD;MAAA,gC ;MAAA,yC;K;;IARJ,sC;MAAA,2F;K;;IAAA,2C;MAAA,a;AAAA,S;UAAA,gD;AAAA,O;UAAA,8C;;UAAA,+D;; K;;IAyByB,4B;MACzB,8B;K;;IC/C4C,8B;K;kDAI5C,mB;MAA6D,c;;QrJ2rD7C,Q;QADhB,IAAI,mCAAsB,cAA 1B,C;UAAqC,aAAO,K;UAAP,e;;QACrB,sB;QAaHb,OAAGB,cAAhB,C;UAAgB,2B;UAAM,IqJ3rD6C,OrJ2rD/B, SqJ3rD+B,UrJ2rD7C,C;YAAwB,aAAO,I;YAAP,e;;QAC9C,aAAO,K;;MqJ5rDsD,iB;K;uDAE7D,oB;MACa,c;;Qr JmqDG,Q;QADhB,IAAI,cqJlqDA,QrJkqDA,iBqJlqDA,QrJkqDsB,UAA1B,C;UAAqC,aAAO,I;UAAP,e;;QACrB,O qJnqDZ,QrJmqDY,W;QAaHb,OAAGB,cAAhB,C;UAAgB,yB;UAAM,IAAI,CqJnqDP,oBrJmqDkB,OqJnqDIB,CrJ

mqDG,C;YAAyB,aAAO,K;YAAP,e;;;QAC/C,aAAO,I;;;MqJpqDH,iB;K;2CAEJ,Y;MAAkC,qBAAQ,C;K;IAEqB,q
E;MAAA,qB;QAC3D,OAAI,OAAO,uBAAX,GAAiB,mBAAjB,GAA6C,SAAH,EAAG,C;O;K;4CADjD,Y;MAAk
C,4BAAa,IAAb,EAAMb,GAAAnB,EAawB,GAAxB,kBAA6B,wCAA7B,C;K;2CAIIC,Y;MAI4C,uBAAgB,IAAhB,
C;K;mDAE5C,iB;MAI4D,yBAAgB,IAAhB,EAAsB,KAAtB,C;K;;IC/BhE,8B;MAAA,e;MAAA,iB;MAAA,uB;K;I
AAA,4B;MAAA,+B;O;MACI,4C;MACA,kD;MACA,0C;MACA,8C;K;;IAHA,mC;MAAA,kB;MAAA,2B;K;;IAC
A,sC;MAAA,kB;MAAA,8B;K;;IACA,kC;MAAA,kB;MAAA,0B;K;;IACA,oC;MAAA,kB;MAAA,4B;K;;IAJj,wB;
MAAA,sH;K;;IAAA,6B;MAAA,a;aAAA,O;UAAA,gC;aAAA,U;UAAA,mC;aAAA,M;UAAA,+B;aAAA,Q;UAAA
,iC;;UAAA,6D;;K;;IAOA,4B;MAKI,mD;MACA,2BAA4B,I;K;yCAE5B,Y;MAEiB,IAAN,I;M5JUX,IAAI,E4JXQ,
mD5JWR,CAAJ,C;QACI,cAda,qB;QAeb,MAAM,gCAAyB,OAAQ,WAAjC,C;;M4JZC,QAAM,oBAAN,M;aACH,
M;UAAc,Y;UAAAd,K;aACA,O;UAAe,W;UAAf,K;;UACQ,wC;UAHL,K;;MAAP,W;K;;CAOJ,Y;MAIW,Q;MAHP,
IAAI,CAAC,cAAL,C;QAAGB,MAAM,6B;MACTb,mD;MAEA,OAAO,2F;K;4DAGX,Y;MACI,iD;MACA,kB;MA
CA,OAAO,kD;K;+CAeX,iB;MAII,2BAAY,K;MACZ,gD;K;sCAGJ,Y;MAII,+C;K;;ICjDkC,wB;MAoFtC,oC;MAp
FgE,6B;K;sCAIhE,Y;MAAuC,0C;K;2CAEvC,mB;MAAwD,uB;;QvJkU3C,Q;QADb,YAAY,C;QACC,sB;QAAb,O
AAa,cAAb,C;UAAa,sB;UACT,IuJnUmE,OvJmUrD,IuJnUqD,UvJmUnE,C;YACI,sBAAO,K;YAAP,wB;;UACJ,qB
;;QAEJ,sBAAO,E;;;MuJvUiD,0B;K;+CAExD,mB;MAA4D,sB;;QvJ2V5D,eAAoB,0BAAa,SAAb,C;QACpB,OAA
O,QAAS,cAAhB,C;UACI,IuJ7VsE,OvJ6VxD,QAAS,WuJ7V+C,UvJ6VtE,C;YACI,qBAAO,QAAS,Y;YAAhB,uB;
;;QAGR,qBAAO,E;;;MuJjWqD,yB;K;0CAE5D,Y;MAA+C,+CAAiB,CAAjB,C;K;kDAE/C,iB;MAAyD,+CAAiB,
KAAjB,C;K;6CAEzD,8B;MAA8D,gCAAQ,IAAR,EAAC,SAAd,EAAYB,OAAzB,C;K;IAEIC,wD;MAAGf,uB;MA
A/E,kB;MAAmC,4B;MAC5D,eAAyB,C;MAGrB,+DAAkB,gBAAIB,EAA6B,OAA7B,EAAsC,WAAK,KAA3C,C;
MACA,eAAa,UAAU,gBAAV,I;K;iDAGjB,iB;MACI,+DAAkB,KAAIB,EAAYB,YAAzB,C;MAEA,OAAO,wBAA
K,mBAAY,KAAZ,IAAL,C;K;4FAGY,Y;MAAQ,mB;K;;oCAGnC,iB;MAMI,IAAI,UAAU,IAAd,C;QAAoB,OAA
O,I;MAC3B,IAAI,2BAAJ,C;QAAuB,OAAO,K;MAE9B,OAAO,2DAAC,IAAd,EAAoB,KAApB,C;K;sCAGX,Y;M
AG+B,oEAAgB,IAAhB,C;K;IAE/B,2C;MAAA,oB;MACI,eACsB,C;K;kDAEtB,Y;MAAkC,sBAAQ,gB;K;+CAE1
C,Y;MAEe,gB;MADX,IAAI,CAAC,cAAL,C;QAAGB,MAAM,6B;MACX,iE;MAAX,OAAO,+B;K;;IAO0B,sD;M
AHZC,oB;MAGwD,iD;MAGhD,gEAAmB,KAAAnB,EAA0B,WAAkB,KAA5C,C;MACA,eAAa,K;K;0DAGjB,Y;M
AAsC,sBAAQ,C;K;wDAE9C,Y;MAAGC,mB;K;uDAEhC,Y;MACI,IAAI,CAAC,kBAAL,C;QAAoB,MAAM,6B;
MAC1B,OAAO,yBAAI,mCAAJ,EAAL,YAAJ,E;K;4DAGX,Y;MAAoC,sBAAQ,CAAR,I;K;;IAGxC,kC;MAAA,sC
;K;iEACI,uB;MACI,IAAI,QAAQ,CAAR,IAAa,SAAS,IAA1B,C;QACI,MAAM,8BAA0B,YAAS,KAAT,gBAAuB,
IAAjD,C;;K;kEAIId,uB;MACI,IAAI,QAAQ,CAAR,IAAa,QAAQ,IAAzB,C;QACI,MAAM,8BAA0B,YAAS,KAAT
,gBAAuB,IAAjD,C;;K;iEAIId,oC;MACI,IAAI,YAAY,CAAZ,IAAiB,UAAU,IAA/B,C;QACI,MAAM,8BAA0B,gB
AAa,SAAb,mBAAkC,OAAIC,gBAAkD,IAA5E,C;;MAEV,IAAI,YAAY,OAAhB,C;QACI,MAAM,gCAAYB,gBA
Aa,SAAb,oBAAmC,OAA5D,C;;K;kEAIId,sC;MACI,IAAI,aAAa,CAAb,IAAkB,WAAW,IAAjC,C;QACI,MAAM,8
BAA0B,iBAAC,UAAAd,oBAAqC,QAArC,gBAAsD,IAAhF,C;;MAEV,IAAI,aAAa,QAAjB,C;QACI,MAAM,gCAA
yB,iBAAC,UAAAd,qBAAsC,QAA/D,C;;K;+DAId,a;MAEe,UACsB,M;MAFhC,iBAAE,C;MACL,mB;MAAV,OAA
U,cAAV,C;QAAU,mB;QACN,aAAW,MAAK,UAAAL,SAAiB,6DAAiB,CAAIC,K;;MAEf,OAAO,U;K;6DAGX,oB
;MAIiB,Q;MAHb,IAAI,CAAE,KAAF,KAAU,KAAM,KAApB,C;QAA0B,OAAO,K;MAEjC,oBAAoB,KAAM,W;
MACb,mB;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,gBAAgB,aAAc,O;QAC9B,IAAI,cAAQ,SAAR,CAAJ,C;UAC
I,OAAO,K;;MAGf,OAAO,I;K;;IAjDf,8C;MAAA,6C;QAAA,4B;;MAAA,sC;K;;ICnFwC,uB;MAyHxC,mC;MAZ
CA,uBAC6B,I;MAmC7B,yBACsC,I;K;8CAnHtC,e;MACI,OAAO,6BAAC,GAAAd,S;K;gDAGX,iB;MAAwE,gBAA
R,Y;MAAQ,c;;QxJkrDxD,Q;QADhB,IAAI,wCAAsB,mBAA1B,C;UAAqC,aAAO,K;UAAP,e;;QACrB,2B;QAAhB
,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IwJlrDwD,OxJkrD1C,OwJlrD6C,MAAH,QxJkrDxD,C;YAAwB,aAAO,I;
YAAP,e;;;QAC9C,aAAO,K;;MwJnrDyD,iB;K;kDAEhE,iB;MAEL,IAAI,gCAAJ,C;QAA+B,OAAO,K;MACTC,UA
AU,KAAM,I;MACHb,YAAY,KAAM,M;MrKiNO,Q;MqKhNzB,erKgN4C,CAAnB,mDAAMb,YqKhNzB,GrKgN
yB,C;MqK9M5C,IAAI,eAAS,QAAT,CAAJ,C;QACI,OAAO,K;;MAIP,6B;MAAA,W;QrK0NqB,U;QqK1ND,UrK0
NoB,CAAnB,uDAAMb,oBqK1NP,GrK0NO,C;;MqK1N5C,W;QACI,OAAO,K;;MAGX,OAAO,I;K;mCAIX,iB;M
AMI,IAAI,UAAU,IAAd,C;QAAoB,OAAO,I;MAC3B,IAAI,0BAAJ,C;QAAYB,OAAO,K;MACHc,IAAI,cAAQ,K
AAM,KAAIB,C;QAAwB,OAAO,K;MAEV,gBAAd,KAAM,Q;MAAQ,c;;QxJ6nDT,Q;QADhB,IAAI,wCAAsB,mB
AA1B,C;UAAqC,aAAO,I;UAAP,e;;QACrB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,CwJ7nDK,2

BxJ6nDM,OwJ7nDN,CxJ6nDT,C;YAAyB,aAAO,K;YAAP,e;;;QAC/C,aAAO,I;;;MwJ9nDH,iB;K;sCAGJ,e;MAA
wC,Q;MAAA,4CAAc,GAAd,8B;K;qCAGx,C,Y;MAK+B,OAAQ,SAAR,YAAQ,C;K;oCAEvC,Y;MAAkC,qBAAQ
,C;K;mFACnB,Y;MAAQ,OAAA,YAAQ,K;K;IAWnB,0E;MAAA,wC;MAAS,sB;K;8EACb,mB;MAAsD,+CAAY,
OAAZ,C;K;IAI3C,sG;MAAA,kD;K;8FACH,Y;MAAkC,OAAA,0BAAc,U;K;2FACHd,Y;MAAyB,OAAA,0BAAc,
OAAO,I;K;;wEAJtD,Y;MACI,oBAAoB,6BAAQ,W;MAC5B,+F;K;sHAMmB,Y;MAAQ,OAAA,qBAAiB,K;K;;mF
Ab5D,Y;MACI,IAAI,4BAAJ,C;QACI,+E;;MAcJ,OAAO,mC;K;IAOwD,uD;MAAA,qB;QAAE,2CAAS,EAAT,C;
O;K;qCAAzE,Y;MAAkC,OAAQ,eAAR,YAAQ,EAaA,IAAb,EAAMB,GAAAnB,EAAwB,GAAxB,kBAA6B,iCAA7
B,C;K;+CAE1C,iB;MAAuD,+BAAS,KAAM,IAAf,IAAsB,GAAAtB,GAA4B,wBAAS,KAAM,MAAf,C;K;+CAEnF,
a;MAAwC,OAAI,MAAM,IAAV,GAAgB,YAAhB,GAAoC,SAAF,CAAE,C;K;IAWtD,4E;MAAA,wC;MAAS,6B;
K;gFACf,mB;MAAsE,iDAAc,OAAc,C;K;IAI3D,wG;MAAA,kD;K;gGACH,Y;MAAkC,OAAA,0BAAc,U;K;6FA
ChD,Y;MAAyB,OAAA,0BAAc,OAAO,M;K;;0EAJtD,Y;MACI,oBAAoB,6BAAQ,W;MAC5B,iG;K;wHAMmB,Y
;MAAQ,OAAA,qBAAiB,K;K;;qFAB5D,Y;MACI,IAAI,8BAAJ,C;QACI,mF;;MAcJ,OAAO,qC;K;oDAMf,e;MAA
8D,gBAAR,Y;MAAQ,sB;;QxJiJ9C,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IwJjJsD,OxJiJx
C,OwJjJ2C,IAAH,MxJiJtD,C;YAAwB,qBAAO,O;YAAP,uB;;;QAC9C,qBAAO,I;;;MwJlJ+C,yB;K;IAEtD,iC;MA
AA,qC;K;4DAEI,a;MAAiE,gC;MAAX,OAAU,CAAC,kBAAN,CAAM,0DAAMB,CAApB,KAA4B,oBAAjC,CAA
iC,8DAAqB,CAAjD,C;K;4DACHe,a;MAAyD,OAAU,SAAL,CAAO,IAAF,mBAAL,CAAY,MAAP,C;K;0DACnE,
oB;MACI,IAAI,gCAAJ,C;QAA+B,OAAO,K;MACtC,OAAO,OAAA,CAAE,IAAF,EAAS,KAAM,IAAf,KAAsB,O
AAA,CAAE,MAAF,EAAW,KAAM,MAAjB,C;K;;IANrC,6C;MAAA,4C;QAAA,2B;;MAAA,qC;K;;IChIqC,uB;
MAkBrC,mC;MAIB+D,6B;K;mCAE/D,iB;MAMI,IAAI,UAAU,IAAd,C;QAAoB,OAAO,I;MAC3B,IAAI,0BAAJ,
C;QAAsB,OAAO,K;MAC7B,OAAO,sDAAU,IAAV,EAAGB,KAAhB,C;K;qCAGX,Y;MAG+B,qEAakB,IAAIB,C
;K;IAE/B,iC;MAAA,qC;K;gEACI,a;MAEoB,Q;MADhB,iBA Ae,C;MACC,mB;MAAhB,OAAgB,cAAhB,C;QAAg
B,yB;QACC,U;QAAb,2BAAa,yEAAuB,CAApC,K;;MAEJ,OAAO,U;K;wDAGX,oB;MACI,IAAI,CAAE,KAAF,K
AAU,KAAM,KAAPB,C;QAA0B,OAAO,K;MACjC,OAAO,CvK40sG,qBuK50xF,KvK40wF,C;K;;luKvPrH,6C;
MAAA,4C;QAAA,2B;;MAAA,qC;K;;MCghBA,kC;MA9hBA,cAAwB,C;MACxB,yB;MAEA,sBAAYB,C;;kFAAz
B,Y;MAAA,0B;K,OAAA,gB;MAAA,0B;K;4CA8BA,uB;MAOI,IAAI,cAAc,CAAIB,C;QAAqB,MAAM,6BAAsB,
mBAAtB,C;MAC3B,IAAI,eAAe,kBAAY,OAA/B,C;QAAqC,M;MACrC,IAAI,uBAAGB,qDAAPB,C;QACI,qBAA
c,gBAAyB,gBAAZ,WAAy,EAAC,EAAd,CAAZB,O;QACd,M;;MAGJ,kBAakB,uDAAY,kBAAY,OAAxB,EAAs
B,WAA9B,C;MACIB,oBAAa,WAAb,C;K;0CAGJ,uB;MAII,kBAakB,gBAAMB,WAAAnB,O;M/J20BtB,U+J10BI,k
B/J00BJ,E+J10ByB,W/J00BzB,E+J10BsC,C/J00BtC,E+J10ByC,W/J00BzC,E+J10B+C,kBAAY,O/J00B3D,C;MA
AA,U+Jz0BI,kB/Jy0BJ,E+Jz0ByB,W/Jy0BzB,E+Jz0BsC,kBAAY,OAAZ,GAAmB,WAAAnB,I/Jy0BtC,E+Jz0B+D,
C/Jy0B/D,E+Jz0BkE,W/Jy0BIE,C;M+Jx0BI,cAAO,C;MACP,qBAAc,W;K;yCAGIB,yB;MAGW,Q;MAAP,OAAO,
2BAAY,aAAZ,4D;K;yCAGX,iB;MAA2C,OAAI,SAAS,kBAAY,OAAzB,GAA+B,QAAQ,kBAAY,OAApB,IAA/B
,GAA6D,K;K;yCAExG,iB;MAA2C,OAAI,QAAQ,CAAZ,GAAe,QAAQ,kBAAY,OAApB,IAAf,GAA6C,K;K;2CA
ExF,iB;MACoD,0BAAY,cAAO,KAAP,IAAZ,C;K;yCAEpD,iB;MAA2C,OAAI,UAAqB,cAAZ,kBAAY,CAAzB,G
AAoC,CAApC,GAA2C,QAAQ,CAAR,I;K;yCAEtF,iB;MAA2C,OAAI,UAAS,CAAb,GAA4B,cAAZ,kBAAY,CA
A5B,GAA2C,QAAQ,CAAR,I;K;mCAEtF,Y;MAAkC,qBAAQ,C;K;iCAE1C,Y;MAGwB,IAAI,cAAJ,C;QAAe,MA
AM,2BAAuB,sBAAvB,C;;QAnBIC,Q;QAmBa,OAnBb,2BAmbkG,WAnBIG,4D;;K;uCAqBX,Y;MAG+B,Q;MAA
A,IAAI,cAAJ,C;QAAA,OAAe,I;;QAxBnC,U;QAwBoB,OAxBpB,6BAwByD,WAxBzD,gE;;MAwBoB,W;K;gCA
E/B,Y;MAGuB,IAAI,cAAJ,C;QAAe,MAAM,2BAAuB,sBAAvB,C;;QA7BjC,Q;QA6BY,OA7BZ,2BAQyC,mBAA
Y,cAqB0D,sBArB1D,IAAZ,CARzC,4D;;K;sCA+BX,Y;MAG8B,Q;MAAA,IAAI,cAAJ,C;QAAA,OAAe,I;;QAICl
C,U;QAKcM,B,OAlCnB,6BAQyC,mBAAy,cA0BiB,sBA1BjB,IAAZ,CARzC,gE;;MAkCmB,W;K;0CAE9B,mB;M
AII,sBA Ae,YAAO,CAAP,IAAf,C;MAEA,cAAO,mBAAy,WAAZ,C;MACP,mBAAy,WAAZ,IAAoB,O;MACpB,
wBAAQ,CAAR,I;K;yCAGJ,mB;MAII,sBA Ae,YAAO,CAAP,IAAf,C;MAEA,mBA7CgD,mBAAy,cA6CIC,SA7C
kC,IAAZ,CA6ChD,IAAmC,O;MACnC,wBAAQ,CAAR,I;K;uCAGJ,Y;MAII,IAAI,cAAJ,C;QAAe,MAAM,2BAAu
B,sBAAvB,C;MA7Dd,Q;MA+DP,cA/DO,2BA+DmB,WA/DnB,4D;MAGeP,mBAAy,WAAZ,IAAoB,I;MACpB,c
AAO,mBAAy,WAAZ,C;MACP,wBAAQ,CAAR,I;MACA,OAAO,O;K;6CAGX,Y;MAGqC,OAAI,cAAJ,GAAe,I
AAf,GAAyB,kB;K;sCAE9D,Y;MAII,IAAI,cAAJ,C;QAAe,MAAM,2BAAuB,sBAAvB,C;MAErB,wBAzEgD,mBA
AY,cAyEtB,sBAzEsB,IAAZ,C;MARzC,Q;MAkFP,cAlFO,2BAkFmB,iBAIFnB,4D;MAmFP,mBAAy,iBAAZ,IAA

iC,I;MACjC,wBAAQ,CAAR,I;MACA,OAAO,O;K;4CAGX,Y;MAGoC,OAAI,cAAJ,GAAe,IAAf,GAAYB,iB;K;q
CAE7D,mB;MAEI,mBAAQ,OAAR,C;MACA,OAAO,I;K;uCAGX,0B;MACI,oCAAa,4BAAmB,KAAAnB,EAA0B,
SAA1B,C;MAEb,IAAI,UAAS,SAAb,C;QACI,mBAAQ,OAAR,C;QACA,M;aACG,IAAI,UAAS,CAAb,C;QACH,o
BAAS,OAAT,C;QACA,M;MAGJ,sBA Ae,YAAO,CAAP,IAAf,C;MA2BA,oBAjIgD,mBAAY,cAiI1B,KAjI0B,IA
AZ,C;MAMlhD,IAAI,QAAS,SAAD,GAAQ,CAAR,IAAe,CAA3B,C;QAEI,+BAA+B,mBAAY,aAAZ,C;QAC/B,s
BAAsB,mBAAY,WAAZ,C;QAEtB,IAAI,4BAA4B,WAAhC,C;UACI,mBAAY,eAAZ,IAA+B,mBAAY,WAAZ,C;
U/JgrB3C,U+J/qBY,kB/J+qBZ,E+J/qBiC,kB/J+qBjC,E+J/qB8C,W/J+qB9C,E+J/qBoD,cAAO,CAAP,I/J+qBpD,E+
J/qB8D,2BAA2B,CAA3B,I/J+qB9D,C;;UAAA,U+J7qBY,kB/J6qBZ,E+J7qBiC,kB/J6qBjC,E+J7qB8C,cAAO,CA
AP,I/J6qB9C,E+J7qBwD,W/J6qBxD,E+J7qB8D,kBAAY,O/J6qB1E,C;U+J5qBY,mBAAY,kBAAY,OAAZ,GAAM
B,CAAnB,IAAZ,IAAoC,mBAAY,CAAZ,C;U/J4qBhD,U+J3qBY,kB/J2qBZ,E+J3qBiC,kB/J2qBjC,E+J3qB8C,C/J
2qB9C,E+J3qBiD,C/J2qBjD,E+J3qBoD,2BAA2B,CAA3B,I/J2qBpD,C;;Q+JxqBQ,mBAAY,wBAAZ,IAAwC,O;Q
ACxC,cAAO,e;;QAGP,WArJ4C,mBAAY,cAqJ/B,SArJ+B,IAAZ,C;QAUJ5C,IAAI,gBAAGB,IAApB,C;U/JkqBR,U
+JjqBY,kB/JiqBZ,E+JjqBiC,kB/JiqBjC,E+JjqB8C,gBAAGB,CAAhB,I/JiqB9C,E+JjqBiE,a/JiqBjE,E+JjqBgF,I/JiqB
hF,C;;UAAA,U+J/pBY,kB/J+pBZ,E+J/pBiC,kB/J+pBjC,E+J/pB8C,C/J+pB9C,E+J/pBiD,C/J+pBjD,E+J/pBoD,I/J+
pBpD,C;U+J9pBY,mBAAY,CAAZ,IAAiB,mBAAY,kBAAY,OAAZ,GAAMB,CAAnB,IAAZ,C;U/J8pB7B,U+J7p
BY,kB/J6pBZ,E+J7pBiC,kB/J6pBjC,E+J7pB8C,gBAAGB,CAAhB,I/J6pB9C,E+J7pBiE,a/J6pBjE,E+J7pBgF,kBA
AY,OAAZ,GAAMB,CAAnB,I/J6pBhF,C;;Q+J1pBQ,mBAAY,aAAZ,IAA6B,O;;MAEjC,wBAAQ,CAAR,I;K;oDA
GJ,mC;MAGkD,UAIxB,M;MANtB,eAAe,QAAS,W;MAEsB,OAAZ,kBAAY,O;MAA9C,iBAAc,aAAAd,wB;QACI,
IAAI,CAAC,QAAS,UAAd,C;UAAyB,K;QACzB,mBAAY,KAAZ,IAAqB,QAAS,O;;MAEZ,oB;MAAtB,mBAAc,
CAAd,8B;QACI,IAAI,CAAC,QAAS,UAAd,C;UAAyB,K;QACzB,mBAAY,OAAZ,IAAqB,QAAS,O;;MAGIC,wB
AAQ,QAAS,KAAjB,I;K;0CAGJ,oB;MACI,IAAI,QAAS,UAAb,C;QAAwB,OAAO,K;MAC/B,sBA Ae,IAAK,KAA
L,GAAY,QAAS,KAArB,IAAf,C;MACA,8BAAtLgD,mBAAY,cAsLvB,SAtLuB,IAAZ,CAsLhD,EAA4C,QAA5C,C;
MACA,OAAO,I;K;0CAGX,2B;MACI,oCAAa,4BAAmB,KAAAnB,EAA0B,SAA1B,C;MAEb,IAAI,QAAS,UAAb,
C;QACI,OAAO,K;aACJ,IAAI,UAAS,SAAb,C;QACH,OAAO,oBAAO,QAAP,C;;MAGX,sBA Ae,IAAK,KAAAL,G
AAAY,QAAS,KAArB,IAAf,C;MAEA,WArMgD,mBAAY,cAqMnC,SArMmC,IAAZ,C;MAAsMhD,oBAAtMgD,mBA
AY,cAsM1B,KAtM0B,IAAZ,C;MAuMhD,mBAAmB,QAAS,K;MAE5B,IAAI,QAAS,SAAD,GAAQ,CAAR,IAAe,
CAA3B,C;QAGI,kBAAkB,cAAO,YAAP,I;QAEIb,IAAI,iBAAiB,WAArB,C;UACI,IAAI,eAAe,CAAnB,C;Y/J0m
BZ,U+JzmBgB,kB/JymBhB,E+JzmBqC,kB/JymBrC,E+JzmBkD,W/JymBID,E+JzmB+D,W/JymB/D,E+JzmBqE,a/
JymBrE,C;;Y+JvmBgB,4BA Ae,kBAAY,OAA3B,I;YACA,sBAAsB,gBAAGB,WAAhB,I;YACtB,kBAAkB,kBAA
Y,OAAZ,GAAMB,WAAAnB,I;YAEIb,IAAI,eAAe,eAAAnB,C;c/JmmBhB,U+JlmBoB,kB/JkmBpB,E+JlmByC,kB/Jk
mBzC,E+JlmBsD,W/JkmBtD,E+JlmBmE,W/JkmBnE,E+JlmByE,a/JkmBzE,C;;cAAA,U+JhmBoB,kB/JgmBpB,E+
JhmByC,kB/JgmBzC,E+JhmBsD,W/JgmBtD,E+JhmBmE,W/JgmBnE,E+JhmByE,cAAO,WAAP,I/JgmBzE,C;cAA
A,U+J/lBoB,kB/J+lBpB,E+J/lByC,kB/J+lBzC,E+J/lBsD,C/J+lBtD,E+J/lByD,cAAO,WAAP,I/J+lBzD,E+J/lB6E,a/J
+lB7E,C;;;UAAA,U+J3lBY,kB/J2lBZ,E+J3lBiC,kB/J2lBjC,E+J3lB8C,W/J2lB9C,E+J3lB2D,W/J2lB3D,E+J3lBiE,
kBAAY,O/J2lB7E,C;U+J1lBY,IAAI,gBAAGB,aAApB,C;Y/J0lBZ,U+JzlBgB,kB/JylBhB,E+JzlBqC,kB/JylBrC,E+J
zlbkD,kBAAY,OAAZ,GAAMB,YAAAnB,I/JylBID,E+JzlbmF,C/JylBnF,E+JzlsF,a/JylBtF,C;;YAAA,U+JvlBgB,k
B/JulBhB,E+JvlBqC,kB/JulBrC,E+JvlBkD,kBAAY,OAAZ,GAAMB,YAAAnB,I/JulBID,E+JvlBmF,C/JulBnF,E+Jvl
BsF,Y/JulBtF,C;YAAA,U+JtlBgB,kB/JslBhB,E+JtlBqC,kB/JslBrC,E+JtlBkD,C/JslBID,E+JtlBqD,Y/JslBrD,E+JtlB
mE,a/JslBnE,C;;Q+JnlBQ,cAAO,W;QACP,8BAAuB,mBAAY,gBAAGB,YAAhB,IAAZ,CAAvB,EAAkE,QAAIE,
C;;QAIA,2BAA2B,gBAAGB,YAAhB,I;QAE3B,IAAI,gBAAGB,IAApB,C;UACI,IAAI,QAAO,YAAP,SAAuB,kBA
AY,OAAvC,C;Y/J2kBZ,U+J1kBgB,kB/J0kBhB,E+J1kBqC,kB/J0kBrC,E+J1kBkD,oB/J0kBID,E+J1kBwE,a/J0kBx
E,E+J1kBuF,I/J0kBvF,C;;Y+JxkBgB,IAAI,wBAAwB,kBAAY,OAAxC,C;c/JwkBhB,U+JvkBoB,kB/JukBpB,E+Jv
kByC,kB/JukBzC,E+JvkBsD,uBAAuB,kBAAY,OAAAnC,I/JukBtD,E+JvkB+F,a/JukB/F,E+JvkB8G,I/JukB9G,C;;c+
JrkBoB,mBAAmB,OAAO,YAAP,GAAsB,kBAAY,OAAIC,I;c/JqkBvC,U+JpkBoB,kB/JokBpB,E+JpkByC,kB/Jok
BzC,E+JpkBsD,C/JokBtD,E+JpkByD,OAAO,YAAP,I/JokBzD,E+JpkB8E,I/JokB9E,C;cAAA,U+JnkBoB,kB/JmkB
pB,E+JnkByC,kB/JmkBzC,E+JnkBsD,oB/JmkBtD,E+JnkB4E,a/JmkB5E,E+JnkB2F,OAAO,YAAP,I/JmkB3F,C;;;
UAAA,U+J/jBY,kB/J+jBZ,E+J/jBiC,kB/J+jBjC,E+J/jB8C,Y/J+jB9C,E+J/jB4D,C/J+jB5D,E+J/jB+D,I/J+jB/D,C;U
+J9jBY,IAAI,wBAAwB,kBAAY,OAAxC,C;Y/J8jBZ,U+J7jBgB,kB/J6jBhB,E+J7jBqC,kB/J6jBrC,E+J7jBkD,uBA

AuB,kBAAY,OAAnc,I/J6jBID,E+J7jB2F,a/J6jB3F,E+J7jB0G,kBAAY,O/J6jBtH,C;;YAAA,U+J3jBgB,kB/J2jBhB
,E+J3jBqC,kB/J2jBrC,E+J3jBkd,C/J2jBID,E+J3jBqD,kBAAY,OAAZ,GAAmB,YAAAnB,I/J2jBrD,E+J3jBsF,kBA
AY,O/J2jBIG,C;YAAA,U+J1jBgB,kB/J0jBhB,E+J1jBqC,kB/J0jBrC,E+J1jBkd,oB/J0jBID,E+J1jBwE,a/J0jBxE,E+
J1jBuF,kBAAY,OAAZ,GAAmB,YAAAnB,I/J0jBvF,C;;Q+JvjBQ,8BAAuB,aAAvB,EAAc,QAAtC,C;;MAGJ,OA
AO,I;K;uCAGX,iB;MACI,oCAAa,2BAAkB,KAAIB,EAAyB,SAAzB,C;MAjRN,Q;MAmRP,OAnRO,2BAQyC,m
BAAY,cA2Q3B,KA3Q2B,IAAZ,CARzC,4D;K;uCAsRX,0B;MACI,oCAAa,2BAAkB,KAAIB,EAAyB,SAAzB,C;
MAEb,oBAjRgD,mBAAY,cAiR1B,KAjR0B,IAAZ,C;MARzC,Q;MA0RP,iBA1RO,2BA0RsB,aA1RtB,4D;MA2R
P,mBAAY,aAAZ,IAA6B,O;MAE7B,OAAO,U;K;0CAGX,mB;MAAoD,0BAAQ,OAAAR,MAAoB,E;K;yCAExE,m
B;MAIsB,IAIA,IAJA,EAIuB,M;MAPzC,WA3RgD,mBAAY,cA2RnC,SA3RmC,IAAZ,C;MA6RhD,IAAI,cAAO,I
AAX,C;QACI,iBAAc,WAAAd,UAAyB,IAAzB,U;UACI,IAAI,gBAAW,mBAAY,KAAZ,CAAX,CAAJ,C;YAAmC,
OAAO,QAAQ,WAAAR,I;;aAE3C,IAAI,eAAQ,IAAZ,C;QACW,kB;QAAuB,SAAZ,kBAAY,O;QAARc,qD;UACI,I
AAI,gBAAW,mBAAY,OAAZ,CAAX,CAAJ,C;YAAmC,OAAO,UAAQ,WAAAR,I;;QAE9C,mBAAc,CAAd,YAAs
B,IAAtB,Y;UACI,IAAI,gBAAW,mBAAY,OAAZ,CAAX,CAAJ,C;YAAmC,OAAO,UAAQ,kBAAY,OAAPB,GA
A2B,WAA3B,I;;MAIID,OAAO,E;K;6CAGX,mB;MAIsC,UAOJ,MAPI,EA0a,M;MAV/C,WA9SgD,mBAAY,cA8
SnC,SA9SmC,IAAZ,C;MAgThD,IAAI,cAAO,IAAX,C;QACkC,kB;QAA9B,iBAAc,OAAO,CAAP,IAAd,yB;UAC
I,IAAI,gBAAW,mBAAY,KAAZ,CAAX,CAAJ,C;YAAmC,OAAO,QAAQ,WAAAR,I;;aAE3C,IAAI,cAAO,IAAX,C
;QACH,mBAAc,OAAO,CAAP,IAAd,aAA8B,CAA9B,Y;UACI,IAAI,gBAAW,mBAAY,OAAZ,CAAX,CAAJ,C;Y
AAmC,OAAO,UAAQ,kBAAY,OAAPB,GAA2B,WAA3B,I;;QAEpB,uBAAZ,kBAAY,C;QAAiB,oB;QAA3C,wD;
UACI,IAAI,gBAAW,mBAAY,OAAZ,CAAX,CAAJ,C;YAAmC,OAAO,UAAQ,WAAAR,I;;MAIID,OAAO,E;K;wC
AGX,mB;MACI,YAAY,mBAAQ,OAAAR,C;MACZ,IAAI,UAAS,EAAb,C;QAAiB,OAAO,K;MACxB,sBAAS,KA
AT,C;MACA,OAAO,I;K;4CAGX,iB;MACI,oCAAa,2BAAkB,KAAIB,EAAyB,SAAzB,C;MAEb,IAAI,UAAS,sBA
Ab,C;QACI,OAAO,iB;aACJ,IAAI,UAAS,CAAb,C;QACH,OAAO,kB;;MAGX,oBAhVgD,mBAAY,cAgV1B,KAh
V0B,IAAZ,C;MARzC,Q;MAyVP,cAzVO,2BAyVmB,aAzVnB,4D;MA2VP,IAAI,QAAQ,aAAS,CAArB,C;QAEI,I
AAI,iBAAiB,WAArB,C;U/Joer,U+JneY,kB/JmeZ,E+JneiC,kB/JmejC,E+Jne8C,cAAO,CAAP,I/Jme9C,E+JnewD,
W/JmexD,E+Jne8D,a/Jme9D,C;;UAAA,U+JjeY,kB/JieZ,E+JjeiC,kB/JiejC,E+Jje8C,C/Jie9C,E+JjeiD,C/JiejD,E+Jje
oD,a/Jiepd,C;U+JheY,mBAAY,CAAZ,IAAiB,mBAAY,kBAAY,OAAZ,GAAmB,CAAnB,IAAZ,C;U/Jge7B,U+J/d
Y,kB/J+djC,E+J/diC,kB/J+djC,E+J/d8C,cAAO,CAAP,I/J+d9C,E+J/dwD,W/J+dxD,E+J/d8D,kBAAY,OAAZ,GAA
mB,CAAnB,I/J+d9D,C;;Q+J5dQ,mBAAY,WAAZ,IAAoB,I;QACpB,cAAO,mBAAY,WAAZ,C;;QAGP,wBAjW4C
,mBAAY,cAiWIB,sBAjWkB,IAAZ,C;QAmW5C,IAAI,iBAAiB,iBAArB,C;U/JsdR,U+JrdY,kB/JqdZ,E+JrdiC,kB/J
qdjC,E+Jrd8C,a/Jqd9C,E+Jrd6D,gBAAGB,CAAhB,I/Jqd7D,E+JrdgF,oBAAoB,CAAPB,I/JqdhF,C;;UAAA,U+JndY
,kB/JmdZ,E+JndiC,kB/JmdjC,E+Jnd8C,a/Jmd9C,E+Jnd6D,gBAAGB,CAAhB,I/Jmd7D,E+JndgF,kBAAY,O/Jmd5F
,C;U+JldY,mBAAY,kBAAY,OAAZ,GAAmB,CAAnB,IAAZ,IAAoC,mBAAY,CAAZ,C;U/JkdhD,U+JjdY,kB/JidZ
,E+JjdiC,kB/JidjC,E+Jjd8C,C/Jid9C,E+JjdiD,C/JidjD,E+JjdoD,oBAAoB,CAAPB,I/JidpD,C;;Q+J9cQ,mBAAY,iBA
AZ,IAAiC,I;;MAErC,wBAAQ,CAAR,I;MAEA,OAAO,O;K;6CAGX,oB;MAAke,0B;;QAa5C,wD;QART,aAAL,I
AAK,U;QAAL,Y;UAA8B,SAAZ,kB9K6wOnB,YAAQ,C;;Q8K7wOX,W;UACI,yBAAO,K;UAAP,2B;;QAEJ,WA
1XgD,mBAAY,cA0XnC,SA1XmC,IAAZ,C;QA2XhD,cAAc,W;QACd,eAAe,K;QAEf,IAAI,cAAO,IAAX,C;UACI,
iBAAc,WAAAd,UAAyB,IAAzB,U;YACI,cAAc,mBAAY,KAAZ,C;YAGd,IAjBsE,CAAU,wBAiBIE,0EAjBkE,CAi
BhF,C;cACI,mBAAY,gBAAZ,EAAY,wBAAZ,YAAyB,O;;cAEzB,WAAW,I;;UAGP,OAAZ,kBAAY,EAAK,IAA
L,EAAW,OAAZ,EAAB,IAAPB,C;;UAGE,oB;UAAuB,SAAZ,kBAAY,O;UAArC,uD;YACI,gBAAc,mBAAY,O
AAZ,C;YACd,mBAAY,OAAZ,IAAqB,I;YAGrB,IA/BsE,CAAU,wBA+BIE,kFA/BkE,CA+BhF,C;cACI,mBAAY,g
BAAZ,EAAY,wBAAZ,YAAyB,S;;cAEzB,WAAW,I;;UAGnB,UAAU,mBAAY,OAAZ,C;UAEV,mBAAc,CAAd,Y
AAsB,IAAtB,Y;YACI,gBAAc,mBAAY,OAAZ,C;YACd,mBAAY,OAAZ,IAAqB,I;YAGrB,IA5CsE,CAAU,wBA4
CIE,kFA5CkE,CA4ChF,C;cACI,mBAAY,OAAZ,IAAuB,S;cACvB,UAAU,mBAAY,OAAZ,C;;cAEV,WAAW,I;;;
QAIvB,IAAI,QAAJ,C;UACI,YAAO,mBAAY,UAAU,WAAV,IAAZ,C;QAEX,yBAAO,Q;;MAvDuD,6B;K;6CAEI
E,oB;MAAke,0B;;QAW5C,wD;QART,aAAL,IAAK,U;QAAL,Y;UAA8B,SAAZ,kB9K6wOnB,YAAQ,C;;Q8K7w
OX,W;UACI,yBAAO,K;UAAP,2B;;QAEJ,WA1XgD,mBAAY,cA0XnC,SA1XmC,IAAZ,C;QA2XhD,cAAc,W;QA
Cd,eAAe,K;QAEf,IAAI,cAAO,IAAX,C;UACI,iBAAc,WAAAd,UAAyB,IAAzB,U;YACI,cAAc,mBAAY,KAAZ,C;
YAGd,IAf+E,wBAejE,0EAfiE,CAe/E,C;cACI,mBAAY,gBAAZ,EAAY,wBAAZ,YAAyB,O;;cAEzB,WAAW,I;U

AGP,OAAZ,kBAAY,EAAC,IAAL,EAAW,OAAZ,EAAoB,IAApB,C;;UAGE,oB;UAAuB,SAAZ,kBAAY,O;UAArC,uD;YACI,gBAAC,mBAAY,OAAZ,C;YACd,mBAAY,OAAZ,IAAqB,I;YAGrB,IA7B+E,wBA6BJE,kFA7BiE,CA6B/E,C;cACI,mBAAY,gBAAZ,EAAY,wBAAZ,YAAyB,S;;cAEzB,WAAW,I;;UAGnB,UAAU,mBAAY,OAAZ,C;UAEV,mBAAC,CAAd,YAAsB,IAAtB,Y;YACI,gBAAC,mBAAY,OAAZ,C;YACd,mBAAY,OAAZ,IAAqB,I;YAGrB,IA1C+E,wBA0CjE,kFA1CiE,CA0C/E,C;cACI,mBAAY,OAAZ,IAAuB,S;cACvB,UAAU,mBAAY,OAAZ,C;;cAEV,WAAW,I,;;QAIvB,IAAI,QAAJ,C;UACI,YAAO,mBAAY,UAAU,WAAV,IAAZ,C;QAEV,yBAAO,Q;;MArDuD,6B;K;2CAEIE,qB;MASsB,IAII,IAJJ,EAKM,MALN,EAaA,MAbA,EAauB,MAbvB,EAkBI,MAIBJ,EAmBM,MAAnBN,EA+BI,M;MAvCb,aAAL,IAAK,U;MAAL,Y;QAA8B,SAAZ,kB9K6wOnB,YAAQ,C;;M8K7wOX,W;QACI,OAAO,K;MAEX,WA1XgD,mBAAY,cA0XnC,SA1XmC,IAAZ,C;MA2XhD,cAAc,W;MACd,eAAe,K;MAEf,IAAI,cAAO,IAAX,C;QACI,iBAAC,WAAAd,UAAyB,IAAZB,U;UACI,cAAc,mBAAY,KAAZ,C;UAGd,IAAI,UAAU,0EAAV,CAAJ,C;YACI,mBAAY,gBAAZ,EAAY,wBAAZ,YAAyB,O;;YAEzB,WAAW,I;;QAGP,OAAZ,kBAAY,EA AK,IAAL,EAAW,OAAZ,EAAoB,IAApB,C;;QAGE,oB;QAAuB,SAAZ,kBAAY,O;QAARc,uD;UACI,gBAAC,mBAAY,OAAZ,C;UACd,mBAAY,OAAZ,IAAqB,I;UAGrB,IAAI,UAAU,kFAAV,CAAJ,C;YACI,mBAAY,gBAAZ,EAAY,wBAAZ,YAAyB,S;;YAEzB,WAAW,I;;QAGnB,UAAU,mBAAY,OAAZ,C;QAEV,mBAAC,CAAd,YAAsB,IAAtB,Y;UACI,gBAAC,mBAAY,OAAZ,C;UACd,mBAAY,OAAZ,IAAqB,I;UAGrB,IAAI,UAAU,kFAAV,CAAJ,C;YACI,mBAAY,OAAZ,IAAuB,S;YACvB,UAAU,mBAAY,OAAZ,C;;YAEV,WAAW,I,;;MAIvB,IAAI,QAAJ,C;QACI,YAAO,mBAAY,UAAU,WAAV,IAAZ,C;MAEX,OAAO,Q;K;iCAGX,Y;MACI,WA7agD,mBAAY,cA6anC,SA7amC,IAAZ,C;MA8ahD,IAAI,cAAO,IAAX,C;QACgB,OAAZ,kBAAY,EAAC,IAAL,EAAW,WAAAX,EAaiB,IAAjB,C;;QACT,IxKtS6C,CAAC,cwKsS9C,C;UACS,OAAZ,kBAAY,EAAC,IAAL,EAAW,WAAAX,EAaiB,kBAAY,OAA7B,C;UACA,OAAZ,kBAAY,EAAC,IAAL,EAAW,CAAX,EAAC,IAAd,C;;MAEhB,cAAO,C;MACP,YAAO,C;K;2CAGX,iB;MAGe,IAAC,IAAD,EAAC,M;MAfP,WACW,eAAC,OAAI,KAAM,OAAN,IAAc,SAaIB,GAAwB,KAAxB,GAAmC,aAAa,KAAb,EAAoB,SAApB,CAApC,uB;MAEX,WA7bgD,mBAAY,cA6bnC,SA7bmC,IAAZ,C;MA8bhD,IAAI,cAAO,IAAX,C;Q/J2XJ,U+J1XQ,kB/J0XR,E+J1X6B,I/J0X7B,EAD+F,CAC/F,E+J1XgD,W/J0XhD,E+J1XiE,I/J0XjE,C;;Q+JzXW,IxKtT6C,CAAC,cwKsT9C,C;U/JyXX,U+JxXQ,kB/JwXR,E+JxX6B,I/JwX7B,E+JxXuD,C/JwXvD,E+JxXuE,W/JwXvE,E+JxXwF,kBAAY,O/JwXpG,C;UAAA,U+JvXQ,kB/JuXR,E+JvX6B,I/JuX7B,E+JvXuD,kBAAY,OAAZ,GAAmB,WAAAnB,I/JuXvD,E+JvX6F,C/JuX7F,E+JvX2G,I/JuX3G,C;;M+JrXLI,IAAI,IAAK,OAAL,GAAY,SAAhB,C;QACI,KAAK,SAAL,IAAa,I;;MAIjB,OAAO,qD;K;mCAGX,Y;MAEI,OAAO,qBAAQ,gBAAmB,SAAnB,OAAR,C;K;+CAGX,iB;MAC0D,4BAAQ,KAAR,C;K;+CAC1D,Y;MAA0C,qB;K;IAE1C,gC;MAAA,oC;MACI,0BrHriBuC,E;MqHsiBvC,sBAAiC,U;MACjC,4BAAuC,E;K;yDAEvC,oC;MAEI,kBAAkB,eAAe,eAAGB,CAA/B,K;MACIB,IAAI,eAAc,WAAAd,QAA4B,CAAhC,C;QACI,cAAc,W;MACIB,IAAI,eAAc,UAAAd,QAA6B,CAAjC,C;QACI,cAAkB,cAAc,UAAIB,GAAGC,UAAhC,GAAMd,U;MACrE,OAAO,W;K;;IAZf,4C;MAAA,2C;QAAA,0B;;MAAA,oC;K;qDAGBA,qB;MAEI,WAVEGd,mBAAY,cAuenC,SAvemC,IAAZ,C;MAwehD,WAAe,kBAaA,cAAO,IAAxB,GAA8B,WAA9B,GAAwC,cAAO,kBAAY,OAAnB,I;MACnD,UAAU,IAAV,EAAGB,cAAhB,C;K;;IA5iBJ,iD;MAAA,oD;MAGwC,+B;MApB5C,sB;MAqBsB,Q;MACV,wBAAmB,CAAnB,C;QA AwB,4D;WACxB,sBAaKB,CAaIB,C;QAAuB,uBAaA,eAAb,O;;QACf,MAAM,gCAAYB,uBAaOB,eAA7C,C;MAHIB,0B;MAJJ,Y;K;IAWA,kC;MAAA,oD;MAGoB,+B;MA/BxB,sB;MAGCQ,sBAAC,qD;MAJIB,Y;K;IAOA,4C;MAAAA,oD;MAG2C,+B;MAtC/C,sB;MAuCQ,sBzJrB8D,YyJqBhD,QzJrBgD,C;MyJsB9D,aAAO,mBAAY,O;MACnB,IAAI,mB9K+qPD,YAAQ,C8K/qPX,C;QAA2B,sBAAC,qD;MAN7C,Y;K;IC5BJ,4B;MAMoB,Q;M/KghqBA,U;MADhB,UAAe,C;MACf,uD;QAAgB,cAAhB,iB;QACI,YAAgB,O+KlhqBiB,O/KkhqBjC,I;;M+KlhqBJ,aAAa,iB/KohqBN,G+KphqBM,C;MACb,wBAAGB,SAAhB,gB;QAAgB,gBAAA,SAAhB,M;QACW,SAAP,MAAO,EAAO,S AAP,C;;MAEX,OAAO,M;K;IAGX,0B;MASiB,Q;MAFb,YAAY,iBAAa,gBAAb,C;MACZ,YAAY,iBAAa,gBAAb,C;MACZ,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,KAAM,WAAI,IAAK,MAAT,C;QACN,KAAM,WAAI,IAAK,OAAT,C;;MAEV,OAAO,UAAS,KAAT,C;K;gGAGX,qB;MAWW,4B;MAAA,U;QAAqB,OAAL,S/K0qPhB,YAAQ,C;;M+K1qPf,W;K;oFAGJ,mC;MAUI,O/K6pPO,qBAAQ,C+K7pPf.GAAe,cAAf,GAAMC,S;K;IAGvC,iD;MAMI,IAAI,cAAS,KAAb,C;QAAoB,OAAO,I;MAC3B,IAAI,qBAAGB,aAAhB,IAAiC,SAAK,OAAL,KAAa,KAAM,OAAxD,C;QAA8D,OAAO,K;MAErE,4C;QACI,SAAS,UAAK,CAAL,C;QACT,SAAS,MAAM,CAAN,C;QAEt,IAAI,OAAO,EAAX,C;UACI,Q;eACG,IAAI,cAAc,UAAIB,C;UACH,OAAO,K;;QAIP,0BAAsB,kBAAtB,C;UAA4C,IAAI,CAAI,kBAAH,EAAG,EAaKB,EAaIB,CAAR,C;YAA+B,OAAO,K;eACIF,8BAAsB,sBAAtB,C;UAA

4C,IAAI,CAAI,cAAH,EAAG,EAAC,EAAd,CAAR,C;YAA2B,OAAO,K;eAC9E,+BAAsB,uBAAtB,C;UAA4C,IAAI,CAAI,cAAH,EAAG,EAAC,EAAd,CAAR,C;YAA2B,OAAO,K;eAC9E,6BAAsB,qBAAtB,C;UAA4C,IAAI,CAAI,cAAH,EAAG,EAAC,EAAd,CAAR,C;YAA2B,OAAO,K;eAC9E,+BAAsB,uBAAtB,C;UAA4C,IAAI,CAAI,cAAH,EAAG,EAAC,EAAd,CAAR,C;YAA2B,OAAO,K;eAC9E,gCAAsB,wBAAtB,C;UAA4C,IAAI,CAAI,cAAH,EAAG,EAAC,EAAd,CAAR,C;YAA2B,OAAO,K;eAC9E,8BAAsB,sBAAtB,C;UAA4C,IAAI,CAAI,cAAH,EAAG,EAAC,EAAd,CAAR,C;YAA2B,OAAO,K;eAC9E,iCAAsB,yBAAtB,C;UAA4C,IAAI,CAAI,cAAH,EAAG,EAAC,EAAd,CAAR,C;YAA2B,OAAO,K;eAE9E,qCAAsB,6BAAtB,C;UAA4C,IAAI,CAAI,gBAAH,EAAG,EAAC,EAAd,CAAR,C;YAA2B,OAAO,K;eAC9E,sCAAsB,8BAAtB,C;UAA4C,IAAI,CAAI,gBAAH,EAAG,EAAC,EAAd,CAAR,C;YAA2B,OAAO,K;eAC9E,oCAAsB,4BAAtB,C;UAA4C,IAAI,CAAI,gBAAH,EAAG,EAAC,EAAd,CAAR,C;YAA2B,OAAO,K;eAC9E,qCAAsB,6BAAtB,C;UAA4C,IAAI,CAAI,gBAAH,EAAG,EAAC,EAAd,CAAR,C;YAA2B,OAAO,K;eAEtE,IAAI,YAAM,EAAN,CAAJ,C;UAAc,OAAO,K;;MAIrC,OAAO,I;K;IAGX,4C;MAKI,IAAI,iBAAJ,C;QA AkB,OAAO,M;MACzB,aAAa,CAAK,eAAL,gBAAK,EAAa,SAAb,CAAL,GAA6C,CAA7C,QAAiD,CAAjD,I;Mv C6SkB,kBAAxB,mBuC5SY,MvC4SZ,C;MuC3SH,oDzK5BgD,gByK4BhD,C;MADJ,O9JnCO,WuH+U6C,W;K;Iu CvSxD,mE;MAEI,IAAY,SAAR,0BAAJ,C;QACI,MAAO,gBAAO,OAAP,C;QACP,M;;MAEJ,SAAU,WAAI,SAAJ ,C;MACV,MAAO,gBAAO,EAAP,C;MAEP,4C;QACI,IAAI,MAAK,CAAT,C;UACI,MAAO,gBAAO,IAAP,C;;QA EX,cAAc,UAAK,CAAL,C;QAEV,IADE,OACF,S;UAAmB,MAAO,gBAAO,MAAP,C;aAC1B,mBAFE,OAEF,E;U AA2B,4BAAR,OAAQ,EAA4B,MAA5B,EAAoC,SAApC,C;aAC3B,uBAHE,OAGF,E;UAAmB,MAAO,gBA Ae,g BAAR,OAAQ,CAAF,C;aAC1B,wBAJE,OAIF,E;UAAmB,MAAO,gBA Ae,gBAAR,OAAQ,CAAF,C;aAC1B,sBAL E,OKAF,E;UAAmB,MAAO,gBA Ae,gBAAR,OAAQ,CAAF,C;aAC1B,uBANE,OAMF,E;UAAmB,MAAO,gBA Ae,gBAAR,OAAQ,CAAF,C;aAC1B,wBAPE,OAOF,E;UAAmB,MAAO,gBA Ae,gBAAR,OAAQ,CAAF,C;aAC1B,yB ARE,OAQF,E;UAAmB,MAAO,gBA Ae,gBAAR,OAAQ,CAAF,C;aAC1B,uBATE,OASF,E;UAAmB,MAAO,gBA Ae,gBAAR,OAAQ,CAAF,C;aAC1B,0BAVE,OAUF,E;UAAmB,MAAO,gBA Ae,gBAAR,OAAQ,CAAF,C;aE1B,k BAZE,OA YF,c;UAAmB,MAAO,gBA Ae,kBAAR,OAAQ,CAAF,C;aAC1B,kBA bE,OAaF,e;UAAmB,MAAO,gBA Ae,kBAAR,OAAQ,CAAF,C;aAC1B,kBA dE,OA cF,a;UAAmB,MAAO,gBA Ae,kBAAR,OAAQ,CAAF,C;aAC1B,k BAF E,OA eF,c;UAAmB,MAAO,gBA Ae,kBAAR,OAAQ,CAAF,C;;MAII C,MAAO,gBAAO,EAAP,C;MACP,SAAU,kBAAmB,iBAAV,SAAU,CAAnB,C;K;ICpJd,uC;MAIqD,+CAAwC,iB AAO,CAA/C,IAAoD,mC;K;IAEzG,4D;MAWQ,kBADE,SACF,O;QADJ,OACc,S;WACV,kBAFE,SAEF,c;QAEQ, yCAAwB,MAAO,KAAP,GAAc,CAAtC,C;UAJZ,OAIuD,S;;UAJvD,OAK6B,mBAAL,SAAK,CAAT,GAA+B,sBA A/B,GAAgD,S;;QALpE,OAogB,oCAAJ,GAA0C,sBAA1C,GAA2D,mB;K;IAG3E,gD;MAWQ,kBADE,SACF,O; QADJ,OACc,S;WACV,kBAFE,SAEF,c;QAFJ,OA E8B,mBAAL,SAAK,CAAT,GAA+B,sBAA/B,GAAgD,S;;QAFr E,OAGgB,oCAAJ,GAA0C,sBAA1C,GAA2D,mB;K;IAG3E,kD;MAKI,OA AI,oCAAJ,GAA0C,sBAA1C,GAA2D,o B;K;IAE/D,kD;MAKI,OA AI,oCAAJ,GAA0C,oBAA1C,GAA2D,iB;K;I1KnD/D,yB;MAAA,6B;K;sCACI,Y;MAA kC,Y;K;0CACIC,Y;MAAsC,Y;K;wCACtC,Y;MAAgC,Q;K;4CAC hC,Y;MAAoC,S;K;mCACpC,Y;MAA+B,MAA M,6B;K;uCACrC,Y;MAAmC,MAAM,6B;K;;IAN7C,qC;MAAA,oC;QAAA,mB;;MAAA,6B;K;IASA,qB;MAAA, yB;MACI,+C;K;iCAEA,iB;MAA4C,qCAoB,KAAM,U;K;mCACtE,Y;MAA+B,Q;K;mCAC/B,Y;MAAkC,W;K;i FAEX,Y;MAAQ,Q;K;kCAC/B,Y;MAAkC,W;K;yCACIC,mB;MAAmD,Y;K;8CACnD,oB;MAAmE,OAAA,QAAS ,U;K;sCAE5E,iB;MAAwC,MAAM,8BAA0B,iDAA8C,KAA9C,MAA1B,C;K;wCAC9C,mB;MAA8C,S;K;4CAC9 C,mB;MAAkD,S;K;mCAEID,Y;MAA6C,kC;K;uCAC7C,Y;MAAQD,kC;K;+CACrD,iB;MACI,IAAI,UAAS,CAAb ,C;QAAgB,MAAM,8BAA0B,YAAS,KAA nC,C;MACTb,OAAO,2B;K;0CAGX,8B;MACI,IAAI,cAAa,CAAb,IAA kB,YAAW,CAAjC,C;QAAoC,OAAO,I;MAC3C,MAAM,8BAA0B,gBAAa,SAAb,mBAAkC,OAA5D,C;K;wCAG V,Y;MAAiC,8B;K;;IA5BrC,iC;MAAA,gC;QAAA,e;;MAAA,yB;K;IA+BA,iC;MAA8D,6BAAkB,SAAIB,EAAoC ,KAApC,C;K;IAE5B,8C;MAAC,oB;MAA0B,0B;K;yFACIC,Y;MAAQ,OAAA,WAAO,O;K;0CACtC,Y;MAAkC, OAAA,WNqqP3B,YAAQ,C;K;iDMpqPf,mB;MAA6C,OAAO,SAAP,WAAO,EAAS,OAAT,C;K;sDACpD,oB;MA AsE,c;;Qc4nDtD,Q;QADhB,IAAI,cd3nDyD,Qc2nDzD,iBd3nDyD,Qc2nDnC,UAA1B,C;UAAqC,aAAO,I;UAAP,e; ;QACrB,Od5nD6C,Qc4nD7C,W;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,Cd5nDkD,oBc4nDvC,Od5 nDuC,Cc4nDtD,C;YAAyB,aAAO,K;YAAP,e;;QAC/C,aAAO,I;;Md7nDsD,iB;K;2CAC7D,Y;MAAuC,OAAO,qB AAP,WAAO,C;K;0CAC9C,Y;MAC+C,gBAAP,W;MAAA,OAAwB,cAAxB,GegKpC,SfhKoC,GekKpC,SN83BoB

,Q;K;IT7hC5B,qB;MAIsC,8B;K;IAEtC,4B;MAIqD,OAAI,QAAS,OAAT,GAAgB,CAApB,GAAgC,OAAT,QAAS,CAAhC,GAA8C,W;K;mFAEnG,yB;MAAA,qD;MAAA,mB;QAK0C,kB;O;KAL1C,C;+FAOA,yB;MAAA,+D;MAAA,mB;QAMwD,uB;O;KANxD,C;2FAQA,yB;MAAA,+D;MAAA,mB;QAMoD,uB;O;KANpD,C;IAQA,mC;MAKI,OAAI,QAAS,OAAT,KAAiB,CAArB,GAAwB,gBAAxB,GAAyC,iBAAU,sBAAkB,QAAIB,EAAwC,IAAxC,CAAV,C;K;IAE7C,iC;MAKI,OAAI,QAAS,OAAT,KAAiB,CAArB,GAAwB,gBAAxB,GAAyC,iBAAU,sBAAkB,QAAIB,EAAwC,IAAxC,CAAV,C;K;IAE7C,gC;MAI2D,OAAI,eAAJ,GAAqB,OAAO,OAAP,CAArB,GAA0C,W;K;IAErG,mC;MAImE,OAAAS,cAAT,QAAS,C;K;gFAE5E,yB;MAaA,gE;MAbA,6B;QAyBI,WAAW,eAduE,IAcvE,C;QWCX,iBAAc,CAAd,UXfkF,IWefU;UXA6B,eAf2D,IAevD,CWCtB,KXDdB,CAAJ,C;QAFyC,OAGB/D,I;O;KA3BX,C;8FAaA,yB;MAAA,gE;MAAA,6B;QAYI,WAAW,eAAa,IAAb,C;QWCX,iBAAc,CAAd,UXAO,IWAP,U;UXA6B,eAAI,KWCtB,KXDdB,CAAJ,C;QAC7B,OAAO,I;O;KADX,C;wFAiBA,yB;Me1FA,+D;Mf0FA,gC;QetF0B,gBAAf,gB;QfsGkB,aW3FzB,W;QX2FA,OW1FO,SIZoC,Q;O;KfsF/C,C;yFAwBA,yB;Me3GA,4E;MAAA,gE;Mf2GA,0C;QevGI,qBf2HyB,Qe3HzB,C;QAC8B,gBAAvB,ef0HkB,Qe1HIB,C;Qf0H4B,aWvHnC,W;QXuHA,OWtHO,SIJ4C,Q;O;KfsGvD,C;IAiCI,mC;MAAQ,uBAAG,iBAAO,CAAP,IAAH,C;K;IAQR,qC;MAAQ,OAAA,SAAK,KAAL,GAAy,CAAZ,I;K;4FAEZ,qB;MAK4D,QAAC,mB;K;kGAE7D,qB;MAWI,OAAO,qBAAgB,SAAK,U;K;sFAGhC,yB;MAAA,qD;MAAA,4B;QAKgE,uCAAQ,W;O;KALxE,C;sFAOA,yB;MAAA,qD;MAAA,4B;QAKoD,uCAAQ,W;O;KAL5D,C;sFAOA,mC;MASI,OAAI,mBAAJ,GAAe,cAAf,GAAmC,S;K;4FAGvC,+B;MAQoH,OAAA,SAAK,qBAAy,QAAZ,C;K;IAGzH,uC;MAK+E,kBAAhB,0B;MAAwB,+B;MAAxB,OW5MpD,W;K;IX+MX,yC;MAAkD,QAAM,cAAN,C;aAC9C,C;UAD8C,OACzC,W;aACL,C;UAF8C,OAeZC,OAAO,sBAAK,CAAL,CAAP,C;UAFyC,OAGtC,S;K;IAGZ,8D;MAGBkE,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,c;MACjG,WAAW,cAAX,EAAiB,SAAjB,EAA4B,OAA5B,C;MAEA,UAAU,S;MACV,WAAW,UAAU,CAAV,I;MAEX,OAAO,OAAO,IAAd,C;QACI,UAAW,GAAy,GAAN,IAAM,KAAC,C;QAC5B,aAAa,sBAAI,GAAJ,C;QACb,UAAU,cAAc,MAAd,EAAb,OAAtB,C;QAEV,IAAI,MAAM,CAAV,C;UACI,MAAM,MAAM,CAAN,I;aACL,IAAI,MAAM,CAAV,C;UACD,OAAO,MAAM,CAAN,I;UAEP,OAAO,G;MAEf,OAAO,EAAE,MAAM,CAAN,IAAF,K;K;IAGX,4E;MAe8E,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,c;MAC7G,WAAW,cAAX,EAAiB,SAAjB,EAA4B,OAA5B,C;MAEA,UAAU,S;MACV,WAAW,UAAU,CAAV,I;MAEX,OAAO,OAAO,IAAd,C;QACI,UAAW,GAAy,GAAN,IAAM,KAAC,C;QAC5B,aAAa,sBAAI,GAAJ,C;QACb,UAAU,UAAW,SAAQ,MAAR,EAAgB,OAAhB,C;QAErB,IAAI,MAAM,CAAV,C;UACI,MAAM,MAAM,CAAN,I;aACL,IAAI,MAAM,CAAV,C;UACD,OAAO,MAAM,CAAN,I;UAEP,OAAO,G;MAEf,OAAO,EAAE,MAAM,CAAN,IAAF,K;K;kGAGX,yB;MAAA,8D;MAAA,4D;MAsBqC,8D;QAAA,qB;UAAE,qBAAC,iBAAS,EAAT,CAAd,EAA4B,WAA5B,C;S;O;MATBvC,+D;QAKBI,yB;UAAA,YAAiB,C;QACjB,uB;UAAA,UAAe,c;QAGf,+BAAa,SAAb,EAAwB,OAAxB,EAAiC,oCAAjC,C;O;KATBJ,C;IA6BA,mE;MAmBoC,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,c;MACnE,WAAW,cAAX,EAAiB,SAAjB,EAA4B,OAA5B,C;MAEA,UAAU,S;MACV,WAAW,UAAU,CAAV,I;MAEX,OAAO,OAAO,IAAd,C;QACI,UAAW,GAAy,GAAN,IAAM,KAAC,C;QAC5B,aAAa,sBAAI,GAAJ,C;QACb,UAAU,WAAW,MAAX,C;QAEV,IAAI,MAAM,CAAV,C;UACI,MAAM,MAAM,CAAN,I;aACL,IAAI,MAAM,CAAV,C;UACD,OAAO,MAAM,CAAN,I;UAEP,OAAO,G;MAEf,OAAO,EAAE,MAAM,CAAN,IAAF,K;K;IAGX,8C;MAMQ,gBAAy,OAAZ,C;QAAuB,MAAM,gCAAYB,gBAAa,SAAb,mCAAKD,OAAID,OAAzB,C;WAC7B,gBAAy,CAAZ,C;QAAiB,MAAM,8BAA0B,gBAAa,SAAb,yBAA1B,C;WACvB,cAAU,IAAV,C;QAAkB,MAAM,8BAA0B,cAAW,OAAx,gCAA2C,IAA3C,OAA1B,C;K;IAChC,8B;MAEoC,MAAM,wBAAoB,8BAApB,C;K;IAE1C,8B;MAEoC,MAAM,wBAAoB,8BAApB,C;K;wF2Gjb1C,yB;M1GgCA,wE;M0GhCA,uC;QAmBW,kB1GqBiD,oB;Q0GM9C,Q;QAAA,OAAK,0B;QAAf,OAAU,cAAV,C;UAAU,mB;UACN,UAAU,sBAAM,CAAN,C;UACV,kBAAkB,sBAAy,GAAZ,C;UACIB,W1GuKJ,a0GvKgB,G1GuKhB,E0GrMyC,SA8BIB,CAAU,GAAV,EAAe,WAAf,EAA4B,CAA5B,EAA+B,uBAAuB,CAAC,WAAy,mBAAy,GAAZ,CAAnE,C1GuKvB,C;Q0GrMA,OAGCO,W;O;KANDX,C;4FAsBA,6C;MAwBc,Q;MAAA,OAAA,SAAK,iB;MAAf,OAAU,cAAV,C;QAAU,mB;QACN,UAAU,sBAAM,CAAN,C;QACV,kBAAkB,sBAAy,GAAZ,C;QACIB,W1GuKJ,a0GvKgB,G1GuKhB,E0GvKuB,UAAU,GAAV,EAAe,WAAf,EAA4B,CAA5B,EAA+B,uBAAuB,CAAC,WAAy,mBAAy,GAAZ,CAAnE,C1GuKvB,C;M0GrKA,OAAO,W;K;IFAGX,yB;MAAA,gB;MAAA,8B;M1GtBA,wE;M0GsBA,6D;QAnCW,kB1GqBiD,oB;Q0GM9C,Q;QAAA,OAAK,0B;QAAf,OAAU,cAAV,C;UAAU,mB;UACN,UAAU,sBAAM,CAAN,C;UACV,kBAAkB,sBAAy,GAAZ,C;UA8BwE,U;UA7B1F,W1GuKJ,a0GvKgB,G1GuKhB,E0G1IkC,UA7BD,GA6BC,EA7BoB,uBAAuB,CAAC,WAAy,mBAAy,GAA

Z,CA6BzC,GAAW,qBA7B3B,GA6B2B,EA7BT,CA6BS,CAAX,GAA6C,UA7BxD,WA6BwD,6DAA5D,EA7BiB,CA6BjB,C1G0IIC,C;;Q0G3IA,OA1BO,W;O;KAGX,C;kFA0BA,yB;MAAA,gB;MAAA,8B;MAAA,0E;QAIcC,Q;QAAA,OAAK,0B;QAAf,OAAU,cAAV,C;UAAU,mB;UACN,UAAU,sBAAM,CAAN,C;UACV,kBA6DQ,WA7DU,WAAY,GAAZ,C;UA6DuF,U;UAAjG,W1G2GZ,a0GvKgB,G1GuKhB,E0G3GiD,UA5DhB,GA4DgB,EA5DK,uBAAuB,CA4DjE,WA5D8E,mBAAy,GAAZ,CA4D1B,GAAW,qBA5D1C,GA4D0C,EA5DxB,CA4DwB,CAAX,GAA6C,UA5DvE,WA4DuE,6DAA5D,EA5DE,CA4DF,C1G2GjD,C;;Q0G5GA,OACY,W;O;KA7BhB,C;IFAgCA,yB;MAAA,gB;MAAA,8B;M1GhFA,wE;M0GgFA,qD;QA7FW,kB1GqBiD,oB;Q0GM9C,Q;QAAA,OAAK,0B;QAf,OAAU,cAAV,C;UAAU,mB;UACN,UAAU,sBAAM,CAAN,C;UACV,kBAaKB,sBAAY,GAAZ,C;UAKFiD,U;UAjFnE,W1GuKJ,a0GvKgB,G1GuKhB,E0GtFgC,UAjFsB,uBAAuB,CAAC,WAAY,mBAAy,GAAZ,CAiFhD,kBAA6B,UAjFjC,WaIFiC,6DAAvC,EAjFmB,CAiFnB,C1GsFhC,C;;Q0GvFA,OA9EO,W;O;KA6DX,C;oFAoBA,yB;MAAA,gB;MAAA,8B;MAAA,kE;QAtFc,Q;QAAA,OAAK,0B;QAAf,OAAU,cAAV,C;UAAU,mB;UACN,UAAU,sBAAM,CAAN,C;UACV,kBA2GQ,WA3GU,WAAY,GAAZ,C;UA2GgE,U;UAA1E,W1G6DZ,a0GvKgB,G1GuKhB,E0G7D+C,UA1GO,uBAAuB,CA0GjE,WA1G8E,mBAAy,GAAZ,CA0GjC,kBAA6B,UA1GhD,WA0GgD,6DAvC,EA1GI,CA0GJ,C1G6D/C,C;;Q0G9DA,OACY,W;O;KAvBhB,C;qFA0BA,yB;MAAA,gB;MAAA,8B;M1G9HA,wE;M0G8HA,uC;QA3IW,kB1GqBiD,oB;Q0GM9C,Q;QAAA,OAAK,0B;QAAf,OAAU,cAAV,C;UAAU,mB;UACN,UAAU,sBAAM,CAAN,C;UACV,kBAaKB,sBAAY,GAAZ,C;UACC,oB;UAKIc,U;UAAjC,IAIIkD,uBAAuB,CAAC,WAAY,mBAAy,GAAZ,CAkItF,C;YADA,mBAjI+C,C;;YAiI/C,mBACKB,UAIIW,GAkIX,EAAe,UAIIC,WAKID,6DAAf,EAI16B,CAkI7B,C;;UAIIB,W1GuKJ,a0GvKgB,G1GuKhB,mB;;Q0GvCA,OA9HO,W;O;KA2GX,C;sFAwBA,yB;MAAA,gB;MAAA,8B;MAAA,oD;QAxIc,Q;QAAA,OAAK,0B;QAAf,OAAU,cAAV,C;UAAU,mB;UACN,UAAU,sBAAM,CAAN,C;UACV,kBA6JQ,WA7JU,WAAY,GAAZ,C;UACC,oB;UA8Jc,U;UAAjC,IA9JkD,uBAAuB,CA4JjE,WA5J8E,mBAAy,GAAZ,CA8JtF,C;YADA,mBA7J+C,C;;YA6J/C,mBACKB,UA9JW,GA8JX,EAAe,UA9JC,WA8JD,6DAAf,EA9J6B,CA8J7B,C;;UAFV,W1GWZ,a0GvKgB,G1GuKhB,mB;;Q0GXA,OAAy,W;O;KAvBhB,C;IA6BA,6C;MArKc,Q;MAAA,OAAK,0B;MAAf,OAAU,cAAV,C;QAAU,mB;QACN,UAAU,sBAAM,CAAN,C;QACV,kBA+KG,WA/Ke,WAAY,GAAZ,C;QA2GgE,U;QAoE/E,W1GPP,a0GvKgB,G1GuKhB,E0G0mC,CA9KmB,uBAAuB,CA8KtE,WA9KmF,mBAAy,GAAZ,CA0GjC,GAoErC,CAPeQc,GAA6B,UA1GhD,WA0GgD,6DAoEnD,IAAM,CAAN,I1GPNc,C;;M0GOA,OAAO,W;K;IgeNp0B,oC;MAAC,kB;MAAuB,kB;K;;wCAN7D,Y;MAMsC,iB;K;wCANtC,Y;MAM6D,iB;K;0CAN7D,wB;MAAA,wBAMsC,qCANtC,EAM6D,qCAN7D,C;K;sCAA,Y;MAAA,OAMsC,mDANtC,IAM6D,wCAN7D,O;K;sCAA,Y;MAAA,c;MAMsC,sD;MAAuB,sD;MAN7D,a;K;oCAA,iB;MAAA,4IAMsC,sCANtC,IAM6D,sCAN7D,I;K;wFrKEA,yB;MAAA,kC;MAAA,4C;MAAA,kD;QAMuF,wC;O;MANvF,4CAOI,Y;QAAuC,8B;O;MAP3C,8E;MAAA,2B;QAMuF,2C;O;KANvF,C;IACsC,2C;MAAC,wC;K;0CACnC,Y;MAAqD,4BAAiB,wBAAjB,C;K;;IAIzD,yC;MAI4D,OAAI,oCAAJ,GAA2B,SAAK,KAhC,GAA0C,I;K;IAEtG,uD;MAI0E,OAAI,oCAAJ,GAA2B,SAAK,KAhC,GAA0C,S;K;IAGpH,8B;MAMoB,Q;MADhB,aAAa,gB;MACG,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACL,OAAP,MAAO,EAAO,OAAP,C;;MAEX,OAAO,M;K;IAGX,4B;MAUiB,Q;MAHb,mBAAmB,mCAAwB,EAAxB,C;MACnB,YAAy,iBAAa,YAAb,C;MACZ,YAAy,iBAAa,YAAb,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,KAAM,WAAI,IAAK,MAAT,C;QACN,KAAM,WAAI,IAAK,OAAT,C;;MAEV,OAAO,UAAS,KAAT,C;K;wFUxDX,qB;MAKqE,gB;K;IAErE,iC;MAMoE,4BAAiB,SAAjB,C;K;uFAEpE,gC;MAKI,OAAgB,mBAAhB,C;QAAgB,8B;QAAM,UAAU,OAAV,C;;K;IAMY,oC;MAAC,0B;MACnC,eAAoB,C;K;yCACpB,Y;MAAwC,OAAA,eAAS,U;K;sCACjD,Y;MAA6E,Q;MAAhC,wBAAa,oBAAmB,mBAAnB,EAAmB,2BAAnB,QAAb,EAA0C,eAAS,OAAAnD,C;K;;sF4J5BjD,yB;MAAA,4E;MAAA,gB;MAAA,8B;MAAA,+C;QAUiC,Q;QAA7B,OAA6B,wCAAqB,QAAS,aAA9B,0D;O;KAVjC,C;wFAYA,yB;MAAA,4E;MAAA,gB;MAAA,8B;MAAA,+C;QAWiC,Q;QAA7B,OAA6B,wCAAqB,QAAS,aAA9B,0D;O;KAXjC,C;sFAaA,+C;MAQL,SAAK,aAAI,QAAS,aAAb,EAAmB,KAAAnB,C;K;ICnCT,8C;MAUI,IAAI,wCAAJ,C;QACI,OAAO,SAAK,4BAAqB,GAARb,C;MAET,4B;M5KoTI,Q;MALX,YAAy,oB4K/Sa,G5K+Sb,C;MACZ,IAAI,iBAAiB,CAAC,4B4KhTG,G5KgTH,CAATB,C;Q4KhTgC,MAAM,2BAAuB,wCAAvB,C;;Q5KoTIC,2BAAO,sE;;M4KpTX,+B;K;IAGJ,8C;MAUQ,kBADE,SACF,kB;QADJ,OACkC,YAAT,SAAK,IAAI,EAAy,YAAZ,C;;QADIC,OAEY,uBAAmB,SAAnB,EAAyB,YAAzB,C;K;IAGhB,gD;MAWQ,kBADE,SACF,yB;QADJ,OACyC,cAAT,SAAK,IAAI,EAAy,YAAZ,C;;QADzC,OAEY,8BAA0B,SAA1B,EAAgC,YAAhC,C;K;;IAC0B,4C;MAAC,wB;MAAoC,0B;K;qEAApC,Y;MAAA,yB;K;0CACvC,iB;MAA4C,OAAI,OAAJ,QAAI,EAAO,KAAP,C;K;4CAChD,Y;

MAA+B,OAAI,SAAJ,QAAL,C;K;4CACnC,Y;MAAkC,OAAA,QAAL,W;K;0FACf,Y;MAAQ,OAAA,QAAL,K;K;2
CACnC,Y;MAAkC,OAAA,QAAL,U;K;qDACtC,e;MAA4C,OAAA,QAAL,mBAAY,GAAZ,C;K;uDACHD,iB;MAA
gE,OAAA,QAAL,qBAAc,KAAAd,C;K;6CACpE,e;MAA+B,OAAA,QAAL,WAAI,GAAJ,C;K;0FACT,Y;MAAQ,OA
AA,QAAL,K;K;4FACH,Y;MAAQ,OAAA,QAAL,O;K;6FACJ,Y;MAAQ,OAAA,QAAL,Q;K;8DAEvD,e;MAAmD,g
BAAJ,Q;MAAI,4B;M5K4Px,C,Q;MALX,YAAAY,oB4KvPyD,G5KuPzD,C;MACZ,IAAI,iBAAiB,CAAC,4B4KxP+
C,G5KwP/C,CAAtB,C;QACI,2B4KzPwE,mB;;Q5K4PxE,2BAAO,sE;;M4K5PoC,+B;K;;IAGN,mD;MAAC,wB;M
AA2C,0B;K;4EAA3C,Y;MAAA,yB;K;iDAC1C,iB;MAA4C,OAAI,OAAJ,QAAL,EAAO,KAAP,C;K;mDACHD,Y;
MAA+B,OAAI,SAAJ,QAAL,C;K;mDACnC,Y;MAAkC,OAAA,QAAL,W;K;iGACf,Y;MAAQ,OAAA,QAAL,K;K;k
DACnC,Y;MAAkC,OAAA,QAAL,U;K;4DACtC,e;MAA4C,OAAA,QAAL,mBAAY,GAAZ,C;K;8DACHD,iB;MA
AgE,OAAA,QAAL,qBAAc,KAAAd,C;K;oDACpE,e;MAA+B,OAAA,QAAL,WAAI,GAAJ,C;K;iGACF,Y;MAAQ,O
AAA,QAAL,K;K;mGACH,Y;MAAQ,OAAA,QAAL,O;K;oGACU,Y;MAAQ,OAAA,QAAL,Q;K;sDAE5E,sB;MAA
yC,OAAA,QAAL,aAAI,GAAJ,EAAS,KAAT,C;K;uDAC7C,e;MAAkC,OAAA,QAAL,cAAO,GAAP,C;K;yDACtC,
gB;MAA2C,QAAL,gBAAO,IAAP,C;K;gDAC/C,Y;MAAuB,QAAL,Q;K;qEAE3B,e;MAAmD,gBAAJ,Q;MAAI,4B;
M5KuOx,C,Q;MALX,YAAAY,oB4KIOyD,G5KkOzD,C;MACZ,IAAI,iBAAiB,CAAC,4B4KnO+C,G5KmO/C,CAAt
B,C;QACI,2B4KpOwE,mB;;Q5KuOxE,2BAAO,sE;;M4KvOoC,+B;K;;I5KvFnD,oB;MAAA,wB;MACI,8C;K;gCA
EA,iB;MAA4C,oCAAsB,KAAAM,U;K;kCACxE,Y;MAA+B,Q;K;kCAC/B,Y;MAAkC,W;K;gFAEX,Y;MAAQ,Q;K
;iCAC/B,Y;MAAkC,W;K;2CAEIC,e;MAA+C,Y;K;6CAC/C,iB;MAAsD,Y;K;mCACtD,e;MAAwC,W;K;mFACY,
Y;MAAQ,6B;K;gFAC/B,Y;MAAQ,6B;K;kFACI,Y;MAAQ,8B;K;uCAEjD,Y;MAAiC,6B;K;;IAjBrC,gC;MAAA,+
B;QAAA,c;;MAAA,wB;K;IAoBA,oB;MAMuE,Q;MAA7B,OAA6B,uE;K;IAEvE,wB;MAaI,OAAI,KAAM,OAAN
,GAAa,CAAjB,GAA0B,QAAN,KAAM,EAAM,qBAAc,YAAAY,KAAM,OAAIB,CAAd,CAAN,CAA1B,GAA6E,U;
K;kFAEjF,yB;MAAA,oD;MAAA,mB;QA08C,iB;O;KAP9C,C;8FASA,yB;MAAA,wE;MAAA,mB;QAQ4D,2B;O;
KAR5D,C;IAUA,+B;MAYiD,gBAA7C,qBAAoB,YAAAY,KAAM,OAAIB,CAApB,C;MAAqD,wB;MAArD,OUJO,
S;K;wFVMX,yB;MAAA,4D;MAAA,mB;QA0sD,qB;O;KAPtD,C;IASA,4B;MAM8G,gBAAvC,eAAc,YAAAY,KA
AM,OAAIB,CAAd,C;MAA+C,wB;MAA/C,OUR5D,S;K;4FVuBX,yB;MAAA,wE;MAAA,mB;QAK8D,2B;O;KA
L9D,C;IAOA,8B;MAU+E,OAAM,QAAN,KAAM,EAAM,qBAAc,YAAAY,KAAM,OAAIB,CAAd,CAAN,C;K;sFA
ErF,yB;MchBA,wE;MdgBA,gC;QcZiC,gBAAtB,oB;Qd8BiB,aU7DxB,W;QV6DA,OU5DO,SI8B2C,Q;O;KdYtD,C
;uFA0BA,yB;McnCA,uE;MdmCA,0C;Qc/ByC,gBAA9B,mBdqDiB,QcrDjB,C;QdqD2B,aU3FIC,W;QV2FA,OU1F
O,SIqCmD,Q;O;Kd+B9D,C;4FAoCA,qB;MAK+D,QAAC,mB;K;kGAEhE,qB;MAWI,OAAO,qBAAGB,mB;K;sFA
G3B,yB;MAAA,oD;MAAA,4B;QAM2D,uCAAQ,U;O;KANnE,C;sFAQA,mC;MASI,OAAI,mBAAJ,GAAe,cAAf,
GAAmC,S;K;yFAEvC,yB;MAyBA,kC;MAAA,8B;MAzBA,iC;QAgCiC,Q;QAx2E,OAwBxD,CAAnB,wDAAmB
,oBAxBoE,GAwBpE,C;O;KAhCpD,C;+EAUA,yB;MAAA,kC;MAAA,8B;MAAA,iC;QAKiC,Q;QAA7B,OAAgD,
CAAnB,wDAAmB,YAAI,GAAJ,C;O;KALpD,C;+EAOA,iC;MAKI,sBAAI,GAAJ,EAAS,KAAT,C;K;4FAGJ,yB;
MAAA,kC;MAAA,8B;MAAA,iC;QA0iC,Q;QAA7B,OAAgD,CAAnB,wDAAmB,oBAAY,GAAZ,C;O;KAPpD,C;
gGASA,4B;MASsG,OAAA,SAAK,qBAAc,KAAAd,C;K;kFAG3G,yB;MAAA,gD;MAAA,8B;MAAA,iC;QASiC,Q;
QAA7B,OAAuD,CAA1B,+DAA0B,eAAO,GAAP,C;O;KAT3D,C;6FAWA,qB;MAWoE,oB;K;6FAEpE,qB;MAW
oE,sB;K;kFAEpE,yB;MAAA,6B;MAAA,4B;QAIgE,qBAAK,aAAL,EAAU,eAAV,C;O;KAJhE,C;2FAMA,wC;MA
MiF,Q;MAAA,mCAAI,GAAJ,oBAAY,c;K;uGAG7F,yB;MAAA,gB;MAAA,8B;MAAA,+C;QAMe,Q;QALX,YA
AY,oBAAI,GAAJ,C;QACZ,IAAI,iBAAiB,CAAC,4BAAY,GAAZ,CAAtB,C;UACI,OAAO,c;;UAGP,OAAO,sE;;O
;KANf,C;IAUA,oC;MAUkD,uCAAqB,GAARb,C;K;sFAEID,wC;MAUW,Q;MADP,YAAAY,oBAAI,GAAJ,C;MAC
L,IAAI,aAAJ,C;QACH,aAAa,c;QACb,sBAAI,GAAJ,EAAS,MAAT,C;QACA,a;;QAEA,Y;;MALJ,W;K;wFASJ,qB
;MAMwF,OAAA,iBAAQ,W;K;wFAEHg,qB;MAMgH,OAAA,iBAAQ,W;K;4FAEXH,6C;Maq1BoB,Q;MAAA,Ob
h1BT,iBag1BS,W;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;Qbh1Ba,Wai1Bb,aAAGB,Obj1Be,Iai1B/B,Ebj1BsC,Sai
1BZ,CAAe,OAAf,CAA1B,C;;Mbj1BhB,OAA6B,W;K;wFAGjC,6C;Ma60BoB,Q;MAAA,Obr0BT,iBaq0BS,W;MA
AhB,OAAgB,cAAhB,C;QAAgB,yB;Qbr0Ba,Was0Bb,abt0B0B,Sas0BtB,CAAY,OAAZ,CAAJ,EAAyC,Obt0BC,M
as0B1C,C;;Mbt0BhB,OAA6B,W;K;IAGjC,kC;MAIyB,Q;MAArB,wBAAqB,KAArB,gB;QAAqB,aAAA,KAArB,
M;QAAK,IAAC,yBAAD,EAAM,2B;QACP,sBAAI,GAAJ,EAAS,KAAT,C;;K;IAIR,oC;MAIyB,Q;MAAA,uB;MA
ArB,OAAqB,cAArB,C;QAAqB,wB;QAAhB,IAAC,yBAAD,EAAM,2B;QACP,sBAAI,GAAJ,EAAS,KAAT,C;;K;I
AIR,oC;MAIyB,Q;MAAA,uB;MAArB,OAAqB,cAArB,C;QAAqB,wB;QAAhB,IAAC,yBAAD,EAAM,2B;QACP,

sBAAI,GA AJ,EAAS,KAAT,C;;K;wFAIR,yB;MAAA,0D;MAAA,uE;MAAA,uC;QASW,kBAAY,mBAAoB,YAA
Y,cAAZ,CAApB,C;Qa8xBH,Q;QAAA,Obh1BT,iBag1BS,W;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;Ubh1Ba,Wai
1Bb,aAAGB,Obj1Be,Iai1B/B,Eb/xB2C,Sa+xBjB,CAAe,OAAf,CAA1B,C;;Qb/xBhB,OAlD6B,W;O;KAyCjC,C;oF
AYA,yB;MAAA,0D;MAAA,uE;MAAA,uC;QAYW,kBAAU,mBAAoB,YAA Y,cAAZ,CAApB,C;Qa+wBD,Q;QA
AA,Obr0BT,iBaq0BS,W;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;Ubr0Ba,Was0Bb,abhxBYc,SagxBrC,CAAY,OA
AZ,CAAJ,EAAYC,Obt0BC,Mas0B1C,C;;QbhxBhB,OAtD6B,W;O;KAOCjC,C;0FAeA,yB;MAAA,wE;MAAA,uC;
QAQkB,Q;QADd,aAAa,oB;QACC,OAAA,SA3FsE,QAAQ,W;QA2F5F,OAAc,cAAAd,C;UAAc,uB;UACV,IAAI,U
AAU,KAAM,IAAhB,CAAJ,C;YACI,MAAO,aAAI,KAAM,IAAV,EAAe,KAAM,MAArB,C;;;QAGf,OAAO,M;O;
KAbX,C;8FAGBA,yB;MAAA,wE;MAAA,uC;QAQkB,Q;QADd,aAAa,oB;QACC,OAAA,SA3GsE,QAAQ,W;QA2
G5F,OAAc,cAAAd,C;UAAc,uB;UACV,IAAI,UAAU,KAAM,MAAhB,CAAJ,C;YACI,MAAO,aAAI,KAAM,IAAV,
EAAe,KAAM,MAArB,C;;;QAGf,OAAO,M;O;KAbX,C;yFAiBA,6C;MAOoB,Q;MAAA,OAAA,SA3HoE,QAAQ,
W;MA2H5F,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI,UAAU,OAAV,CAAJ,C;UACI,WAAY,aAAI,OAAQ,IA
AZ,EAAiB,OAAQ,MAAzB,C;;;MAGpB,OAAO,W;K;qFAGX,yB;MAAA,wE;MAAA,uC;QAOW,kBAAS,oB;QA
fA,Q;QAAA,OA3HoE,iBAAQ,W;QA2H5F,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAcmC,Sad/B,CAAU,OAAV,
CAAJ,C;YACI,WAAY,aAAI,OAAQ,IAAZ,EAAiB,OAAQ,MAAzB,C;;;QApB,OAVO,W;O;KAGX,C;+FAUA,6
C;MAOoB,Q;MAAA,OAAA,SAPJoE,QAAQ,W;MAoJ5F,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI,CAAC,UA
AU,OAAV,CAAL,C;UACI,WAAY,aAAI,OAAQ,IAAZ,EAAiB,OAAQ,MAAzB,C;;;MAGpB,OAAO,W;K;2FAG
X,yB;MAAA,wE;MAAA,uC;QAOW,kBAAY,oB;QAFH,Q;QAAA,OApJoE,iBAAQ,W;QAoJ5F,OAAgB,cAAhB,
C;UAAgB,yB;UACZ,IAAI,CackC,SAdjC,CAAU,OAAV,CAAL,C;YACI,WAAY,aAAI,OAAQ,IAAZ,EAAiB,OA
AQ,MAAzB,C;;;QApB,OAVO,W;O;KAGX,C;IAUA,0B;MAQqB,IAAN,I;MADX,IAAI,oCAAJ,C;QACW,QAA
M,cAAN,C;eACH,C;YAAK,iB;YAAL,K;eACA,C;YAAK,aAAU,8BAAJ,GAakB,sBAAK,CAAL,CAAIB,GAA+
B,oBAAW,OAAhD,C;YAAL,K;;YACQ,0BAAM,qBAAoB,YAA Y,cAAZ,CAApB,CAAN,C;YAHL,K;;QAAP,W;;
MAMJ,OAAoC,oBAA7B,mBAAM,oBAAN,CAA6B,C;K;IAGxC,yC;MAIwB,SAApB,WAAoB,Y;MAApB,kB;K;
IAEJ,4B;MAM6D,QAAM,gBAAN,C;aACzD,C;UADyD,OACpD,U;aACL,C;UAFyD,OAEPD,MAAM,UAAK,CA
AL,CAAN,C;;UAFoD,OAGjD,mBAAM,qBAAoB,YAA Y,gBAAZ,CAApB,CAAN,C;;K;IAGZ,yC;MAIwB,OAAP
B,WAAoB,Y;MAApB,kB;K;IAEJ,4B;MAM4D,OAA6B,oBAA7B,mBAAM,oBAAN,CAA6B,C;K;IAEzF,yC;MAI
wB,SAApB,WAAoB,Y;MAApB,kB;K;IAEJ,4B;MAMqD,QAAM,cAAN,C;aACjD,C;UADiD,OAC5C,U;aACL,C;
UAFiD,OC/X8B,uB;;Ud+X9B,OAGzC,uB;;K;IAGZ,iC;MAMmE,4BAAc,SAAd,C;K;IAEnE,yC;MAKI,WAAoB,0
B;MAApB,kB;K;IAEJ,kC;MAOI,Q;MAAA,IAAI,SAAK,UAAT,C;QAAA,OAAoB,MAAM,IAAN,C;;QAAqC,kB
AApB,qBAAc,SAAd,C;QAA4B,wBAAS,UAAT,EAAqB,WAArB,C;QAAjE,OUhiBO,W;;MVgiBP,W;K;IAEJ,mC
;MAOI,Q;MAAA,IAAI,SAAK,UAAT,C;QAAA,OAA0B,MAAN,KAAM,C;;QAAiC,kBAApB,qBAAc,SAAd,C;Q
AA4B,4B;QAAnE,OUziBO,W;;MVyiBP,W;K;IAEJ,mC;MAOI,Q;MAAA,IAAI,SAAK,UAAT,C;QAAA,OAA0B,
QAAN,KAAM,C;;QAAiC,kBAApB,qBAAc,SAAd,C;QAA4B,0B;QAAnE,OUljBO,W;;MVkjBP,W;K;IAEJ,mC;M
AOwB,kBAApB,qBAAc,SAAd,C;MAA4B,4B;MAA5B,OAA4C,oBU3jBrC,WV2jBqC,C;K;IAEhD,iC;MAOwB,k
BAApB,qBAAc,SAAd,C;MAA4B,+B;MAA5B,OUpkBO,W;K;0FVukBX,2B;MAKI,sBAAI,IAAK,MAAT,EAAg
B,IAAK,OAARb,C;K;4FAGJ,yB;MAAA,gD;MAAA,mC;QAKI,kBAAO,KAAP,C;O;KALJ,C;4FAQA,yB;MAAA,
gD;MAAA,mC;QAKI,kBAAO,KAAP,C;O;KALJ,C;4FAQA,yB;MAAA,gD;MAAA,mC;QAKI,kBAAO,KAAP,C;
O;KALJ,C;4FAQA,0B;MAKI,yBAAO,GAAP,C;K;IAGJ,kC;MAOwB,kBAAf,aAAL,SAAK,C;MASCL,6B;MATC
A,OAA+C,oBUtnBxC,WVsnBwC,C;K;IAEnD,mC;MAQwB,kBAAf,aAAL,SAAK,C;MAqCK,YAAL,gBAAK,O;
MARCV,OAAgD,oBUhoBzC,WVgoByC,C;K;IAEPD,mC;MAQwB,kBAAf,aAAL,SAAK,C;MAoCK,YAAL,gBA
AK,O;MApCV,OAAgD,oBU1oBzC,WV0oByC,C;K;IAEPD,mC;MAQwB,kBAAf,aAAL,SAAK,C;MAMCK,YAA
L,gBAAK,O;MANCV,OAAgD,oBUppBzC,WVopByC,C;K;4FAEPD,0B;MAMI,uBAAO,GAAP,C;K;8FAGJ,yB;
MAAA,sD;MAAA,kC;QAMc,UAAV,SAAK,KA AK,EAAU,IAAV,C;O;KANd,C;8FASA,yB;MAAA,sD;MAAA,k
C;QAMc,UAAV,SAAK,KA AK,EAAU,IAAV,C;O;KANd,C;8FASA,yB;MAAA,sD;MAAA,kC;QAMc,UAAV,SA
AK,KA AK,EAAU,IAAV,C;O;KANd,C;IAUA,wC;MACsD,QAAM,cAAN,C;aACID,C;UADkD,OAC7C,U;aACL,
C;UAFkD,gB;;UAAA,OAG1C,S;;K;oF6KtwBZ,yB;MAAA,8D;MAAA,8B;MAAA,qC;QAUiC,Q;QAA7B,OAA2
D,CAA9B,sEAA8B,eAAO,OAAP,C;O;KAV/D,C;wFAYA,yB;MAAA,8D;MAAA,8B;MAAA,sC;QASiC,Q;QAA7
B,OAA2D,CAA9B,sEAA8B,oBAAU,QAAV,C;O;KAT/D,C;wFAWA,yB;MAAA,8D;MAAA,8B;MAAA,sC;QASi

C,Q;QAA7B,OAA2D,CAA9B,sEAA8B,oBAAU,QAAV,C;O;KAT/D,C;4FAWA,8B;MAKI,SAAK,WAAI,OAAJ,C;K;4FAGT,yB;MAAA,gD;MAAA,sC;QAKS,OAAL,SAAK,EAAO,QAAP,C;O;KALT,C;4FAQA,yB;MAAA,gD;MAAA,sC;QAKS,OAAL,SAAK,EAAO,QAAP,C;O;KALT,C;4FAQA,yB;MAAA,gD;MAAA,sC;QAKS,OAAL,SAAK,EAAO,QAAP,C;O;KALT,C;8FAQA,8B;MAKI,SAAK,cAAO,OAAP,C;K;8FAGT,yB;MAAA,sD;MAAA,sC;QAKS,UAAL,SAAK,EAAU,QAAV,C;O;KALT,C;8FAQA,yB;MAAA,sD;MAAA,sC;QAKS,UAAL,SAAK,EAAU,QAAV,C;O;KALT,C;8FAQA,yB;MAAA,sD;MAAA,sC;QAKS,UAAL,SAAK,EAAU,QAAV,C;O;KALT,C;IAQA,qC;MAIU,IAIe,I;MAHjB,kBADE,QACF,c;QAAiB,OAAO,yBAAO,QAAP,C;;QAEpB,aAAsB,K;QACT,0B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,IAAI,oBAAI,IAAJ,CAAJ,C;YAAe,SAAS,I;;QAC5B,OAAO,M;;K;IAKnB,uC;MAKiB,Q;MADb,aAAsB,K;MACT,0B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,IAAI,oBAAI,IAAJ,CAAJ,C;UAAe,SAAS,I;;MAE5B,OAAO,M;K;IAGX,uC;MAII,OAAO,yBAAGB,OAAT,QAAS,CAAhB,C;K;IAGX,0C;MAIW,iBAAmB,gCAAT,QAAS,EAAgC,SAAhC,C;MAIHG,Q;MAkH7B,OAIH2D,CAA9B,sEAA8B,oBAAU,UAAV,C;K;IAqH/D,0C;MAII,UAAmB,8BAAT,QAAS,C;MACnB,09K0EwD,C8K1EjD,G9K0EkD,U8K1EID,IAAoB,4BAAU,GAAV,C;K;IAG/B,0C;MAII,OpLqoPO,EoLroPA,QpL6jPA,YAAQ,CAwER,CoLroPA,IAAyB,4BAAmB,8BAAT,QAAS,CAAnB,C;K;IAGpC,0C;MAIW,iBAAmB,gCAAT,QAAS,EAAgC,SAAhC,C;MA7HG,Q;MA6H7B,O A7H2D,CAA9B,sEAA8B,oBAAU,UAAV,C;K;IAGl/D,0C;MAII,IpLunPO,EoLvnPH,QpL+iPG,YAAQ,CAwER,C oLvnPP,C;QACI,OAAO,4BAAmB,8BAAT,QAAS,CAAnB,C;;QAEp,OAAO,wB;K;IAGf,0C;MAII,UAAmB,8BAAT,QAAS,C;MACnB,I9K0CwD,C8K1CpD,G9K0CqD,U8K1CzD,C;QACI,OAAO,4BAAU,GAAV,C;;QAEp,OA AO,wB;K;IAGf,kC;MACI,a9KmCwD,CAAC,mB;M8KICzD,iB;MACA,OAAO,M;K;IAIX,2C;MAKkF,gCAAc,SAAd,EAAyB,IAAzB,C;K;IAEIF,2C;MAKkF,gCAAc,SAAd,EAAyB,KAAzB,C;K;IAEIF,sE;MACI,iBAAa,KAAb,C;MnKlJgB,kBmKmjX,oB;MACD,OAAO,qBAAP,C;QACI,IAAI,UAAU,kBAAV,6BAAJ,C;UACI,oB;UACA,W AAS,I;;MAGrB,OAAO,Q;K;oFAIX,4B;MAM6D,kCAAS,KAAT,C;K;IAE7D,gC;MAKiD,IAAI,mBAAJ,C;QAAe,MAAM,2BAAuB,gBAAvB,C;;QAArB,OAAMe,2BAAS,CAAT,C;K;IAEpH,sC;MAKwD,OAAI,mBAAJ,GAAe,IAAf,GAAyB,2BAAS,CAAT,C;K;IAEjF,+B;MAKgD,IAAI,mBAAJ,C;QAAe,MAAM,2BAAuB,gBAAvB,C;;QAArB,OAAMe,2BAAS,2BAAT,C;K;IAEnH,qC;MAKuD,OAAI,mBAAJ,GAAe,IAAf,GAAyB,2BAAS,2BAAT,C;K;IAEhF,2C;MAK8E,kCAAc,SAAd,EAAyB,IAAzB,C;K;IAE9E,2C;MAK8E,kCAAc,SAAd,EAAyB,KAAzB,C;K;IAE9E,wE;MAEgB,UAGS,MAHT,EAcY,MAdZ,EAc6B,M;MAfzC,IAAI,uCAAJ,C;QACI,OAAoC,cAA5B,sEAA4B,EAAC,SAAd,EAAyB,uBAAzB,C;MAExC,iBAAsB,C;MACD,oC;MAArB,qBAAkB,CAAIB,mC;QACI,cAAc,sBA AK,SAAL,C;QACd,IAAI,UAAU,OAAV,MAAsB,uBAA1B,C;UACI,Q;QAEJ,IAAI,eAAc,SAAIb,C;UACI,sBAA K,UAAL,EAAmB,OAAnB,C;QAEJ,+B;;MAEJ,IAAI,aAAa,cAAjB,C;QACwB,oC;QAAiB,mB;QAArC,oE;UACI,2BAAS,WAAAT,C;QAEJ,OAAO,I;;QAEp,OAAO,K;;K;IChS+w,C;MAAkC,uB;MAAjC,0B;K;4FACpB,Y;MAAQ,OAAA,eAAS,K;K;iDACxC,iB;MAAkC,mCAAS,0BAAoB,KAApB,CAAT,C;K;;IAGT,gC;MAAyC,8B;MAAx C,0B;K;oFACH,Y;MAAQ,OAAA,eAAS,K;K;yCACxC,iB;MAAkC,mCAAS,0BAAoB,KAApB,CAAT,C;K;mCA EIC,Y;MAAuB,eAAS,Q;K;8CACHC,iB;MAAuC,OAAA,eAAS,kBAAS,0BAAoB,KAApB,CAAT,C;K;yCAEhD,0 B;MAA8C,OAAA,eAAS,aAAI,0BAAoB,KAApB,CAAJ,EAAgC,OAAhC,C;K;yCACvD,0B;MACI,eAAS,aAAI,2 BAAqB,KAArB,CAAJ,EAAiC,OAAjC,C;K;;IAIjB,+C;MACoB,Q;MAAA,kC;MAAhB,IAAa,CAAT,0BAAJ,C;Q AAA,OAA2B,8BAAy,KAAZ,I;;QAAuB,MAAM,8BAA0B,mBAAgB,KAAhB,2BAA0C,gBAAG,2BAAH,CAA1 C,OAA1B,C;K;IAE5D,gD;MACoB,Q;MAAA,qB;MAAhB,IAAa,CAAT,0BAAJ,C;QAAA,OAAsB,iBAAO,KAAP ,I;;QAAkB,MAAM,8BAA0B,oBAAiB,KAAjB,2BAA2C,gBAAG,cAAH,CAA3C,OAA1B,C;K;IAGID,+B;MAK+ C,gCAAqB,SAArB,C;K;IAE/C,iC;MAM6D,wBAAa,SAAb,C;K;;IxKpC7D,oD;MAQuF,wC;K;IARvF,8CASI,Y; MAAuC,8B;K;IAT3C,gF;IyKY8G,wC;MAAA,mB;QAAE,kBAAS,aAAT,C;O;K;IAThY,yB;MASqG,oCAAS,sBA AT,C;K;8FAErG,yB;MAAA,kD;MzKdA,kC;MAAA,0C;MAAA,kD;QAQuF,wC;O;MARvF,4CASI,Y;QAAuC,8B; O;MAT3C,8E;MyKiB2I,qD;QAAA,mB;UAAE,gBAAS,qBAAT,C;S;O;MAH7I,gC;QAGkI,kCAAS,mCAAT,C;O; KAHII,C;IAKA,2B;MAQI,eAAe,6B;MACf,oBAA0B,+BAAN,KAAM,EAAwC,QAAxC,EAA+D,QAA/D,C;MAC 1B,OAAO,Q;K;8FAGX,yB;MAAA,kD;MAAA,gC;QAGkI,gBAAS,aAAT,C;O;KAHII,C;IAGB0C,yB;K;+CAoBtC, kC;MAOI,IAAI,uCAA0B,QAAS,UAAvC,C;QAAkD,M;MACID,OAAO,sBAAS,QAAS,WAAIB,e;K;+CAGX,kC; MAQqD,6BAAS,QAAS,WAAIB,e;K;;I;IAezD,mC;MAA2C,wB;MACvC,eAAoB,C;MACpB,mBAA4B,I;MAC 5B,sBAAyC,I;MACzC,gBAAoC,I;K;gDAEpC,Y;MACI,OAAO,IAAP,C;QACI,QAAM,YAAN,C;eACI,C;YAAA, K;eACA,C;YACI,IAAI,kCAAe,UAAAnB,C;cACI,eAAQ,C;cACR,OAAO,I;;cAEP,sBAAe,I;;YALvB,K;eAOA,C;Y

AAc,OAAO,K;eACrB,C;eAAA,C;YAAgC,OAAO,I;;YAC/B,MAAM,yB;;QAGIB,eAAQ,C;QACR,WAAW,4B;Q
ACX,gBAAW,I;QACX,IzH/FR,oBDgDQ,W0H+CY,kB1H/CZ,CChDR,C;;K;6CyHmGA,Y;MACU,IASe,I;MATrB
,QAAM,YAAN,C;aACI,C;aAAA,C;UAAsC,OAAO,qB;aAC7C,C;UACI,eAAQ,C;UACR,OAAO,kCAAE,O;aAE1
B,C;UACI,eAAQ,C;UACR,aACa,mF;UACb,mBAAY,I;UACZ,OAAO,M;;UAEH,MAAM,yB;;K;uDAItB,Y;MACI
,IAAI,CAAC,cAAL,C;QAAgB,MAAM,6B;;QAA8B,OAAO,W;K;2DAG/D,Y;MAA4C,QAAM,YAAN,C;aACxC,
C;UADwC,OAC1B,6B;aACd,C;UAFwC,OAExB,6BAAsB,sBAAtB,C;;UAFwB,OAGhC,6BAAsB,uCAAoC,YAA
1D,C;;K;IAOqC,4E;MAAA,oB;QACzC,wCAAW,C;QAAX,OACA,yB;O;K;oDALR,+B;MACI,mBAAY,K;MACZ
,eAAQ,C;MACR,OAA6C,0CAAtC,c;K;IAUsC,+E;MAAA,oB;QACzC,wCAAW,C;QAAX,OACA,yB;O;K;yDAN
R,kC;MACI,IAAI,CAAC,QAAS,UAAAd,C;QAAYB,M;MACzB,sBAAE,Q;MACf,eAAQ,C;MACR,OAA6C,6CAAt
C,c;K;2DAMX,kB;M1HjBO,Q;MADP,e0HoBI,M1HpbJ,C;MACO,Q0HmBH,M1HnBG,+D;M0HoBH,eAAQ,C;K
;kGAIR,Y;MAAQ,0C;K;;IzK1LhB,oD;MAQuF,wC;K;IARvF,8CASI,Y;MAAuC,8B;K;IAT3C,gF;sFAAA,yB;MA
AA,kC;MAAA,0C;MAAA,kD;QAQuF,wC;O;MARvF,4CASI,Y;QAAuC,8B;O;MAT3C,8E;MAAA,2B;QAQuF,2
C;O;KARvF,C;IAiBgE,+C;MAAA,mB;QAAE,sB;O;K;IALIE,kC;MAKuD,OAakB,2CAAT,+BAAS,E;K;IAEzE,8
B;MAK6D,OAAI,Qb2rPtD,YAAQ,Ca3rP0C,GAawB,eAAxB,GAAsD,WAAT,QAAS,C;K;IAEnH,yB;MAG8C,kC
;K;IAE9C,yB;MAAA,6B;K;uCACI,Y;MAA6C,kC;K;2CAC7C,a;MAA4B,kC;K;2CAC5B,a;MAA4B,kC;K;;IAHh
C,qC;MAAA,oC;QAAA,mB;;MAAA,6B;K;oFAMA,yB;MAAA,2D;MAAA,4B;QAM4D,uCAAQ,e;O;KANpE,C;I
AgB4F,mH;MAAA,wC;MAAA,6B;MAAA,yB;MAAA,wC;MAAA,wD;MAAA,kC;K;;;kDAAA,Y;;;;cACxF,eA
Ae,uBAaA,W;cAC5B,IAAI,QAAS,UAAb,C;gBACI,gB;gCAAA,sCAAS,QAAT,O;oBAAA,2C;yBAAA,yB;gBAA
A,Q;;gBAEA,gB;gCAAA,sCAAS,iCAAT,O;oBAAA,2C;yBAAA,yB;gBAAA,Q;;;;cAJJ,W;;cAAA,W;;;;;;;
K;IADwF,gE;MAAA,yD;uBAAA,uG;YAAA,S;iBAAA,Q;;iBAAA,uB;O;K;IAP5F,4C;MAOmF,gBAAS,uCAAT,
C;K;IAGBb,4B;MAAE,OAAA,EAAG,W;K;IAP3E,8B;MAO8D,4BAAQ,cAAR,C;K;IAUQ,8B;MAAE,OAAA,EA
AG,W;K;IAR3E,8B;MAQ8D,4BAAQ,gBAAR,C;K;IAM1B,8B;MAAE,S;K;IAJtC,wC;MAEGb,Q;MADZ,IAAI,8
CAAJ,C;QACI,OAA4C,CAApC,2EAAoC,kBAAQ,QAAR,C;;MAEHd,OAAO,uBAAmB,SAAnB,EAAyB,gBAAz
B,EAAiC,QAAjC,C;K;IAGX,4B;MAYiB,Q;MAFb,YAAY,gB;MACZ,YAAY,gB;MACC,2B;MAAb,OAAa,cAAb,
C;QAAa,sB;QACT,KAAM,WAAI,IAAK,MAAT,C;QACN,KAAM,WAAI,IAAK,OAAT,C;;MAEV,OAAO,UAAS
,KAAT,C;K;IAGX,+B;MAQqD,6BAAS,4BAAT,C;K;IAW0B,+G;MAAA,wC;MAAA,6B;MAAA,yB;MAAA,0C;
MAAA,4C;MAAA,0B;MAAA,kC;K;;;mDAAA,Y;;;;kCAC9D,0C;cACb,gB;;;;cAAA,IAAO,iBPYfKd,UOzFzD,
C;gBAAA,gB;;;cACI,QAAQ,yBAAO,iBAAQ,iBAAO,KAAf,C;cACf,WAAkB,WAAP,iBAAO,C;cACIB,YAAgB,
IAAI,iBAAO,KAAf,GAAqB,iBAAO,aAAI,CAAJ,EAAO,IAAP,CAA5B,GAA8C,I;cAC1D,gB;8BAAA,iCAAM,K
AAN,O;kBAAA,2C;uBAAA,yB;cAAA,Q;;cAJJ,gB;;;cAMJ,W;;;;;;K;IAR+E,4D;MAAA,yD;uBAAA,mG;Y
AAA,S;iBAAA,Q;;iBAAA,uB;O;K;IAT/E,uC;MASmE,gBAAY,kCAAZ,C;K;IAkBhC,0D;MAE/B,wB;QAAA,WA
AgC,I;MADhC,0B;MACA,0B;MACA,4B;K;IAGuC,0E;MAAA,oD;MACnC,gBAAE,iCAAS,W;MACxB,iBAAqB,
E;MACrB,gBAAmB,I;K;oEAEnB,Y;MACI,OAAO,aAAS,UAAhB,C;QACI,WAAW,aAAS,O;QACpB,IAAI,wCA
AU,IAAV,MAAmB,sCAAvB,C;UACI,gBAAW,I;UACX,iBAAY,C;UACZ,M;;MAGR,iBAAY,C;K;8DAGhB,Y;
MASW,Q;MARP,IAAI,mBAAa,EAAjB,C;QACI,iB;MACJ,IAAI,mBAAa,CAAjB,C;QACI,MAAM,6B;MACV,aA
Aa,a;MACb,gBAAW,I;MACX,iBAAY,E;MAEZ,OAAO,yE;K;IEAGX,Y;MACI,IAAI,mBAAa,EAAjB,C;QACI,iB
;MACJ,OAAO,mBAAa,C;K;;2CAhC5B,Y;MAAuC,yD;K;;IA2C3C,qD;MAAY,0B;MAAmC,gC;K;IACJ,gF;MAA
A,0D;MACnC,gBAAE,oCAAS,W;K;IEACxB,Y;MACI,OAAO,6CAAY,aAAS,OAAR,B,C;K;oEAGX,Y;MACI,OA
AO,aAAS,U;K;;8CAPxB,Y;MAAuC,4D;K;qDAWvC,oB;MACI,OAAO,uBAA4B,eAA5B,EAAsC,kBAAtC,EAA
mD,QAAnD,C;K;;IAUf,4D;MAAY,0B;MAAmC,gC;K;IACJ,8F;MAAA,wE;MACnC,gBAAE,2CAAS,W;MACxB,
aAAY,C;K;wEACZ,Y;MAC0C,Q;MAAtC,OAAO,oDAAY,oBAAmB,iBAAnB,EAAM,yBAAnB,QAAY,EAAyC,
aAAS,OAALD,C;K;2EAGX,Y;MACI,OAAO,aAAS,U;K;;qDARxB,Y;MAAuC,mE;K;;IAkB3C,oC;MAAY,0B;K;I
AC6C,wE;MACjD,gBAAE,gCAAS,W;MACxB,aAAY,C;K;6DACZ,Y;MAC2C,Q;MAAvC,OAAO,iBAAa,oBAA
mB,iBAAnB,EAAM,yBAAnB,QAAb,EAA0C,aAAS,OAAnD,C;K;gEAGX,Y;MACI,OAAO,aAAS,U;K;;0CARx
B,Y;MAAqD,wD;K;;IAmBzD,0D;MACI,4B;MACA,4B;MACA,4B;K;IAEuC,sE;MAAA,gD;MACnC,iBAAgB,gC
AAU,W;MAC1B,iBAAgB,gCAAU,W;K;4DAC1B,Y;MACI,OAAO,sCAAU,cAAU,OAAPB,EAA4B,cAAU,OAAt
C,C;K;+DAGX,Y;MACI,OAAO,cAAU,UAUV,IAAuB,cAAU,U;K;;yCARhD,Y;MAAuC,uD;K;;IAc3C,6D;MACI,
0B;MACA,gC;MACA,0B;K;IAEuC,4E;MAAA,sD;MACnC,gBAAE,kCAAS,W;MACxB,oBAAiC,I;K;+DAEjC,Y;

MACI,IAAI,CAAC,2BAAL,C;QACI,MAAM,6B;MACV,OAAO,gCAAE,O;K;kEAG1B,Y;MACI,OAAO,2B;K;+E
AGX,Y;MACQ,Q;MAAJ,IAAI,iEAA2B,KAA/B,C;QACI,oBAAe,I;MAEnB,OAAO,yBAAP,C;QACI,IAAI,CAAC
,aAAS,UAAa,C;UACI,OAAO,K;UAEp,cAAc,aAAS,O;UACvB,uBAAuB,wCAAS,2CAAY,OAAZ,CAAT,C;UA
CvB,IAAI,gBAAiB,UAArB,C;YACI,oBAAe,gB;YACf,OAAO,I;;;MAInB,OAAO,I;K;;4CA9Bf,Y;MAAuC,0D;K;
;IAoC9B,6I;MAAA,wC;MAAA,6B;MAAA,yB;MAAA,4C;MAAA,kD;MAAA,gD;MAAA,wB;MAAA,yB;MAAA
,kC;K;;;yDAAA,Y;;;;kBAGyC,I;ICAFIC,C;cACI,sD;cAAhB,gB;;;;cAAA,KAAgB,yBAAhB,C;gBAAA,gB;;;cA
AgB,oC;cACZ,aAAa,6BAAU,oBAAmB,uBAAmB,EAAMb,+BAAmB,QAAV,EAAuC,OAAvC,C;cACb,gB;8BAA
A,sCAAS,4BAAS,MAAT,CAAT,O;kBAAA,2C;uBAAA,yB;cAAA,Q;;cAFJ,gB;;;cAIJ,W;;;;K;IANS,0F;M
AAA,yD;uBAAA,iI;YAAA,S;iBAAA,Q;;iBAAA,uB;O;K;IADb,wD;MACI,gBAAS,kDAAT,C;K;;;IAoByB,qD;M
ACzB,0B;MACA,8B;MACA,0B;MC3TA,IAAI,ED+TQ,qBAAc,CC/TtB,CAAJ,C;QACI,cD8T2B,+CAA4C,iB;QC
7TvE,MAAM,gCAAYB,OAAQ,WAAjC,C;;MAFV,IAAI,EDgUQ,mBAAy,CChUpB,CAAJ,C;QACI,gBD+TyB,6
CAA0C,e;QC9TnE,MAAM,gCAAYB,SAAQ,WAAjC,C;;MAFV,IAAI,EDiUQ,mBAAy,iBCjUpB,CAAJ,C;QACI,
gBDgUkC,0DAAuD,eAAvD,WAAmE,iB;QC/TrG,MAAM,gCAAYB,SAAQ,WAAjC,C;;K;sFDkUa,Y;MAAQ,yB
AAW,iBAAX,I;K;yCAE/B,a;MAAyC,OAAI,KAAK,YAAT,GAAgB,eAAhB,GAAqC,gBAAy,eAAZ,EAASB,oBA
Aa,CAAb,IAAtB,EAASc,eAAtC,C;K;yCAC9E,a;MAAyC,OAAI,KAAK,YAAT,GAAgB,IAAhB,GAA0B,gBAAy
,eAAZ,EAASB,iBAAtB,EAACc,oBAAa,CAAb,IAAI,C;K;IAEzC,8D;MAAA,wC;MAEtB,gBAAe,2BAAS,W;M
ACxB,gBAAe,C;K;0DAEf,Y;MAEI,OAAO,gBAAW,kCAAX,IAAyB,aAAS,UAAzC,C;QACI,aAAS,O;QACT,qC;
;K;2DAIR,Y;MACI,a;MACA,OAAQ,gBAAW,gCAAZ,IAAyB,aAAS,U;K;wDAG7C,Y;MACI,a;MACA,IAAI,iBA
AY,gCAAhB,C;QACI,MAAM,6B;MACV,qC;MACA,OAAO,aAAS,O;K;;qCAvBxB,Y;MAA0B,mD;K;;IAgCA,u
C;MAC1B,0B;MACA,oB;MC3WA,IAAI,ED+WQ,gBAAS,CC/WjB,CAAJ,C;QACI,cD8WsB,yCAASc,YAAtC,M
;QC7WtB,MAAM,gCAAYB,OAAQ,WAAjC,C;;K;0CDgXV,a;MAAyC,OAAI,KAAK,YAAT,GAAgB,eAAhB,GA
AqC,gBAAy,eAAZ,EAASB,CAAtB,EAAYB,YAAzB,C;K;0CAC9E,a;MAAyC,OAAI,KAAK,YAAT,GAAgB,IAA
hB,GAA0B,iBAaA,eAAb,EAuB,CAAvB,C;K;IAE5B,gE;MACnC,YAAW,yB;MACX,gBAAe,4BAAS,W;K;yDA
ExB,Y;MACI,IAAI,cAAQ,CAAZ,C;QACI,MAAM,6B;MACV,6B;MACA,OAAO,aAAS,O;K;4DAGpB,Y;MACI,
OAAO,YAAO,CAAP,IAAY,aAAS,U;K;;sCAZpC,Y;MAAuC,oD;K;;IA5B3C,gD;MACI,0B;MACA,4B;K;IAEuC,
0E;MAAA,oD;MACnC,gBAAe,iCAAS,W;MACxB,iBAAqB,E;MACrB,gBAAmB,I;K;oEAEnB,Y;MACI,IAAI,aA
AS,UAAb,C;QACI,WAAW,aAAS,O;QACpB,IAAI,wCAAU,IAAV,CAAJ,C;UACI,iBAAY,C;UACZ,gBAAW,I;U
ACX,M;;;MAGR,iBAAY,C;K;8DAGhB,Y;MAMiB,Q;MALb,IAAI,mBAAa,EAAjB,C;QACI,iB;MACJ,IAAI,mB
AAa,CAAjB,C;QACI,MAAM,6B;MACV,aACa,gF;MAGb,gBAAW,I;MACX,iBAAY,E;MACZ,OAAO,M;K;iEA
GX,Y;MACI,IAAI,mBAAa,EAAjB,C;QACI,iB;MACJ,OAAO,mBAAa,C;K;;2CAIC5B,Y;MAAuC,yD;K;;IA2Cb,u
C;MAC1B,0B;MACA,oB;MC5bA,IAAI,ED+bQ,gBAAS,CC/bjB,CAAJ,C;QACI,cD8bsB,yCAASc,YAAtC,M;QC
7btB,MAAM,gCAAYB,OAAQ,WAAjC,C;;K;0CDgcV,a;MIvXO,SJsXmC,eAAQ,CAAR,I;MAAD,OAA4B,KAAK,
CAAT,GAAY,yBAAZ,GAAuC,iBAAa,eAAb,EAuB,EAAvB,C;K;0CACxG,a;MIvXO,SJuXmC,eAAQ,CAAR,I;
MAAD,OAA4B,KAAK,CAAT,GAAY,yBAAZ,GAAuC,gBAAy,eAAZ,EAASB,YAAtB,EAA6B,EAA7B,C;K;IA
EjE,gE;MACnC,gBAAe,4BAAS,W;MACxB,YAAW,yB;K;2DAEX,Y;MAEI,OAAO,YAAO,CAAP,IAAY,aAAS,
UAA5B,C;QACI,aAAS,O;QACT,6B;;K;yDAIR,Y;MACI,a;MACA,OAAO,aAAS,O;K;4DAGpB,Y;MACI,a;MAC
A,OAAO,aAAS,U;K;;sCAnBxB,Y;MAAuC,oD;K;;IA6B3C,gD;MACI,0B;MACA,4B;K;IAGuC,0E;MAAA,oD;M
ACnC,gBAAe,iCAAS,W;MACxB,iBAAqB,E;MACrB,gBAAmB,I;K;gEAEnB,Y;MACI,OAAO,aAAS,UAAhB,C;
QACI,WAAW,aAAS,O;QACpB,IAAI,CAAC,wCAAU,IAAV,CAAL,C;UACI,gBAAW,I;UACX,iBAAY,C;UACZ
,M;;;MAGR,iBAAY,C;K;8DAGhB,Y;MAMqB,Q;MALjB,IAAI,mBAAa,EAAjB,C;QACI,a;MAEJ,IAAI,mBAAa,
CAAjB,C;QACI,aACa,gF;QACb,gBAAW,I;QACX,iBAAY,C;QACZ,OAAO,M;;MAEX,OAAO,aAAS,O;K;iEAG
pB,Y;MACI,IAAI,mBAAa,EAAjB,C;QACI,a;MACJ,OAAO,mBAAa,CAAb,IAAkB,aAAS,U;K;;2CAIC1C,Y;MA
AuC,yD;K;;IAuCN,+C;MAAC,sB;MAAiC,gC;K;0CACnE,Y;MAAuC,4BAAiB,aAAO,WAAxB,EAAoC,kBAAPC
,C;K;;IAGP,+C;MAAuE,2B;MAAtE,sB;MAAiC,gC;MACIE,kBAAuB,c;K;6CAEvB,Y;MACI,OAAO,aAAO,UAA
d,C;QACI,WAAW,aAAO,O;QACIB,UAAU,mBAAy,IAAZ,C;QAEV,IAAI,eAAS,WAAI,GAJ,CAAb,C;UACI,
mBAAQ,IAAR,C;UACA,M;;;MAIR,W;K;;IAKgC,0D;MAAC,wC;MAAuC,kC;K;IACrC,0E;MAAA,oD;MACnC,
gBAAmB,I;MACnB,iBAAqB,E;K;oEAERB,Y;MACI,gBAAe,mBAAa,EAAjB,GAAqB,+CAArB,GAA4C,2CAAa,
4BAAb,C;MACvD,iBAAgB,qBAAJ,GAASB,CAAtB,GAA6B,C;K;8DAG7C,Y;MAMiB,Q;MALb,IAAI,iBAAY,C

AAhB,C;QACI,iB;MAEJ,IAAI,mBAAa,CAAjB,C;QACI,MAAM,6B;MACV,aAAa,8D;MAEb,iBAAY,E;MACZ,
OAAO,M;K;iEAGX,Y;MACI,IAAI,iBAAY,CAAhB,C;QACI,iB;MACJ,OAAO,mBAAa,C;K;;2CAxB5B,Y;MAAu
C,yD;K;;IA6B3C,kC;MAWI,OAAW,iDAAJ,GAAwC,SAAxG,GAakD,4BAAwB,SAAxB,C;K;IAeIB,uD;MAAA,
qB;QAAE,6B;O;K;IAX7C,wC;MAWI,OAA2D,cAApD,sBAakB,YAAIB,EAAgC,qCAAhC,CAAOd,C;K;IAqBrC,
iD;MAAA,mB;QAAE,mB;O;K;IAIB5B,gD;MAeI,OAAI,YAAJ,GACI,2BADJ,GAGI,sBAakB,+BAAIB,EAA4B,
YAA5B,C;K;IAER,wD;MAcI,6BAAkB,YAAIB,EAAgC,YAAhC,C;K;ILxpBJ,oB;MAAA,wB;MACI,8C;K;gCAE
A,iB;MAA4C,oCAAmB,KAAM,U;K;kCACrE,Y;MAA+B,Q;K;kCAC/B,Y;MAAkC,W;K;gFAEX,Y;MAAQ,Q;K;i
CAC/B,Y;MAAkC,W;K;wCACIC,mB;MAAMd,Y;K;6CACnD,oB;MAAmE,OAAA,QAAS,U;K;kCAE5E,Y;MAA
6C,kC;K;uCAE7C,Y;MAAiC,6B;K;;;IAdrC,gC;MAAA,+B;QAAA,c;;MAAA,wB;K;IAkBA,oB;MAIoC,6B;K;IAE
pC,2B;MAMmD,OAAI,QAAS,OAAT,GAAgB,CAApB,GAAgC,MAAT,QAAS,CAAhC,GAA6C,U;K;iFAEHg,yB
;MAAA,mD;MAAA,mB;QAKwC,iB;O;KALxC,C;6FAOA,yB;MAAA,uE;MAAA,mB;QAQsD,2B;O;KARtD,C;I
AUA,kC;MAKiE,OAAS,aAAT,QAAS,EAAa,qBAAc,YAAy,QAAS,OAARb,CAAd,CAAb,C;K;uFAE1E,yB;MA
AA,2D;MAAA,mB;QAGgD,qB;O;KAHhD,C;IAKA,+B;MAC2D,OAAS,aAAT,QAAS,EAAa,eAAQ,YAAy,QAAS,
OAARb,CAAR,CAAb,C;K;2FAEpE,yB;MAAA,uE;MAAA,mB;QAMwD,2B;O;KANxD,C;IAQA,iC;MAKmE,O
AAS,aAAT,QAAS,EAAa,qBAAc,YAAy,QAAS,OAARb,CAAd,CAAb,C;K;IAE5E,+B;MAMyD,OAAI,eAAJ,GA
AqB,MAAM,OAAN,CAARb,GAAyC,U;K;IAEIG,kC;MAQI,OAAgB,gBAAT,QAAS,EAAgB,sBAAhB,C;K;sFAG
pB,yB;MavBA,uE;MbuBA,gC;QanB8B,gBAAnB,oB;QbqCiB,aS/CxB,W;QT+CA,OS9CO,SISwC,Q;O;KbmBnD,
C;wFA0BA,yB;Ma1CA,wE;Mb0CA,0C;QatCsC,gBAA3B,mBb4DiB,Qa5DjB,C;Qb4D2B,aS7E1C,W;QT6EA,OS5
EO,SIgBgD,Q;O;KbsC3D,C;sFA+BA,yB;MAAA,mD;MAAA,4B;QAEkD,uCAAQ,U;O;KAF1D,C;IAIA,wC;MA
AgD,QAAM,cAAN,C;aAC5C,C;UAD4C,OACvC,U;aACL,C;UAF4C,OAEvC,MAAM,oBAAW,OAAjB,C;;UAFu
C,OAGpC,S;;K;IKnKZ,oD;MAQuF,wC;K;IARvF,8CASI,Y;MAAuC,8B;K;IAT3C,gF;I0KLA,yC;MzK4BI,IAAI,E
yK3BI,OAAO,CAAP,IAAY,OAAO,CzK2BvB,CAAJ,C;QACI,cyK3BI,aAAJ,GACI,yEADJ,GAGI,8C;QzKyBJ,M
AAM,gCAAyB,OAAQ,WAAjC,C;;K;IyKnBM,mI;MAAA,mB;QAAE,wBAAiB,gCAAjB,EAA6B,YAA7B,EAAM
C,YAANc,EAAyC,sBAAZc,EAAyD,mBAAZD,C;O;K;IAFtB,gF;MACI,oBAAoB,IAApB,EAA0B,IAA1B,C;MAC
A,oCAAgB,6EAAhB,C;K;IAKyB,yL;MAAA,wC;MAAA,6B;MAAA,yB;MAAA,wC;MAAA,wC;MAAA,gD;MA
AA,sD;MAAA,4D;MAAA,wB;MAAA,0B;MAAA,uB;MAAA,0B;MAAA,wB;MAAA,qB;MAAA,4B;MAAA,kC;
K;;;2DAAA,Y;;;cACrB,4BAAiC,eAAL,uBAAK,EAAa,IAAb,C;+BACvB,0BAAO,uBAAP,I;cACV,IAAI,kBAA
O,CAAX,C;oCACiB,iBAaA,qBAAb,C;kCACF,C;gBACD,6C;gBAAV,iB;;;sCAaa,gBAAc,qBAAd,C;gBACH,+C;
gBAAV,gB;;;cAAA,KAAU,2BAAV,C;gBAAA,gB;;;cAAU,kC;cACN,mBAAO,WAAI,GAJ,C;ACP,IAAI,m
BAAO,SAAX,C;gBACI,IAAI,mBAAO,KAAP,GAAc,uBAAlB,C;kBAA0B,sBAAS,mBAAO,kBAAuB,uBAAvB,
C;kBAA8B,gB;;;kBAAxE,gB;;;gBADJ,gB;;;cAGI,gB;8BAAA,iCAAU,8BAAJ,GAAiB,mBAAjB,GAA6B,iBAA
U,mBAAV,CAAnC,O;kBAAA,2C;uBAAA,yB;cAAA,Q;;cACA,mBAAO,qBAAy,uBAAZ,C;cAJX,gB;;;cAFJ,gB;
;cASA,IAAI,iCAAJ,C;gBACI,gB;;;gBADJ,iB;;;cACI,IAAO,mBAAO,KAAd,IAAqB,uBAARb,C;gBAAA,gB;;;c
ACI,gB;8BAAA,iCAAU,8BAAJ,GAAiB,mBAAjB,GAA6B,iBAAU,mBAAV,CAAnC,O;kBAAA,2C;uBAAA,yB;
cAAA,Q;;cACA,mBAAO,qBAAy,uBAAZ,C;cAFX,gB;;;cAIA,IjL4K4C,CiL5KxK,mBjL4KyC,UiL5K7C,C;gBA
AyB,iB;gCAAA,iCAAM,mBAAN,O;oBAAA,2C;yBAAA,yB;gBAAA,Q;;gBAAZB,iB;;;cAjCR,W;;cA4BI,iB;;;c
A1BJ,iB;;;cAGI,KAAU,yBAAV,C;gBAAA,iB;;;6BAAU,sB;cACN,IAAI,kBAAO,CAAX,C;gBAAgB,oCAAQ,CA
AR,I;gBAAW,iB;;;gBAA3B,iB;;;cACA,iBAAO,WAAI,YAAJ,C;cACP,IAAI,iBAAO,KAAP,KAAe,uBAAnB,C;g
BACI,iB;gCAAA,iCAAM,iBAAN,O;oBAAA,2C;yBAAA,yB;gBAAA,Q;;gBADJ,iB;;;cAEI,IAAI,8BAAJ,C;gBA
AiB,iBAAO,Q;;gBAAa,oBAAS,iBAAU,uBAAV,C;cAC9C,kBAAO,c;cAHX,iB;;;cAHJ,iB;;;cASA,IjL+LgD,CiL/L
5C,iBjL+L6C,UiL/LjD,C;gBACI,IAAI,qCAAkB,iBAAO,KAAP,KAAe,uBAARc,C;kBAA2C,iB;kCAAA,iCAAM,i
BAAN,O;sBAAA,2C;2BAAA,yB;kBAAA,Q;kBAA3C,iB;;;gBADJ,iB;;;cAdJ,W;;cAcI,iB;;;cAZJ,iB;;;cAkCJ,W;
;;;K;IArCyB,sI;MAAA,yD;uBAAA,6K;YAAA,S;iBAAA,Q;;iBAAA,uB;O;K;IAF7B,6E;MACI,IAAI,CAA
C,QAAS,UAAAd,C;QAAYB,OAAO,2B;MACHC,OAAO,WAAkB,0EAAIB,C;K;IAwCwB,6B;MAA8B,uB;MAA7B,
kB;MACHC,mBAA6B,C;MAC7B,eAAyB,C;K;2CAEZB,8B;MACI,+DAAkB,SAAlB,EAA6B,OAA7B,EAAc,W
AAK,KAA3C,C;MACA,mBAAiB,S;MACjB,eAAa,UAAU,SAAV,I;K;0CAGjB,iB;MACI,+DAAkB,KAAIB,EAA
yB,YAAzB,C;MAEA,OAAO,wBAAK,mBAAY,KAAZ,IAAL,C;K;qFAGY,Y;MAAQ,mB;K;;IASR,wC;MAAQd,U
B;MAApD,sB;MzKrDxB,IAAI,EyKuDQ,cAAc,CzKvDtB,CAAJ,C;QACI,cyKsD2B,wE;QzKrD3B,MAAM,gCAA

yB,OAAQ,WAAjC,C;;MAFV,IAAI,EyKwDQ,cAAc,aAAO,OzKxD7B,CAAJ,C;QACI,gByKuDqC,wFAA+E,aAA
O,O;QzKiD3H,MAAM,gCAAYB,SAAQ,WAAjC,C;;MyK2DV,kBAAuB,aAAO,O;MAC9B,oBAA8B,C;MAE9B,s
BAAyB,U;K;kFAAZB,Y;MAAA,0B;K,OAAA,gB;MAAA,0B;K;uCAGA,iB;MAGW,Q;MAFP,+DAAkB,KAAIB,
EAAyB,SAAZB,C;MAEA,OAAO,sBAmGmC,CAnG5B,iBAmG6B,GAnGV,KAmGU,IAAD,IAAa,eAnGhD,4D;K
;kCAGX,Y;MAAe,qBAAQ,e;K;IAEgB,4D;MAAA,sC;MAAS,2B;MAC5C,eAAoB,oB;MACpB,eAAoB,4B;K;8D
AEpB,Y;MAKgB,Q;MAJZ,IAAI,iBAAS,CAAb,C;QACI,W;;QAGA,mBAAQ,sCAAQ,YAAP,4DAAR,C;QACA,e
AoFkC,CAPf1B,YAoF2B,GApFb,CAoFa,IAAD,IAAa,+B;QAnF/C,mC;;K;;oCAXZ,Y;MAAuC,kD;K;2CAGvC,i
B;MAGiE,UAQ1C,MAR0C,EAe1C,MAf0C,EAqBtD,M;MatBP,aACQ,KAAM,OAAN,GAAa,IAAK,KAAtB,GA
AkC,UAAN,KAAM,EAAO,IAAK,KAAZ,CAAIc,GAAyD,kD;MAE7D,WAAW,IAAK,K;MAEhB,WAAW,C;MA
CX,UAAU,iB;MAEV,OAAO,OAAO,IAAP,IAAe,MAAM,eAA5B,C;QACI,OAAO,IAAP,IAAe,wBAAO,GAAP,g
E;QACf,mB;QACA,iB;;MAGJ,MAAM,C;MACN,OAAO,OAAO,IAAd,C;QACI,OAAO,IAAP,IAAe,wBAAO,GA
AP,gE;QACf,mB;QACA,iB;;MAEJ,IAAI,MAAO,OAAP,GAAC,IAAK,KAAvB,C;QAA6B,OAAO,IAAK,KAAZ,I
AAoB,I;MAEjD,OAAO,uD;K;mCAGX,Y;MACI,OAAO,qBAAQ,gBAAa,SAAb,OAAR,C;K;4CAGX,uB;MAKI,k
BAAoD,eAAjC,mBAAy,mBAAa,CAAzB,IAA8B,CAA9B,IAAiC,EAAa,WAAb,C;MACpD,gBAAoB,sBAAC,CA
AIB,GAA4B,UAAP,aAAO,EAAO,WAAP,CAA5B,GAAqD,qBAAQ,gBAAa,WAAb,OAAR,C;MACrE,OAAO,eA
AW,SAAX,EAA5B,SAAtB,C;K;qCAGX,mB;MAII,IAAI,aAAJ,C;QACI,MAAM,6BAAsB,qBAAtB,C;;MAGV,cA
6B0C,CA7BnC,iBA6BoC,GA7BjB,SA6BiB,IAAD,IAAa,eA7BvD,IAAmC,O;MACnC,6B;K;+CAGJ,a;MzKhJA,I
AAI,EyKoJQ,KAAK,CzKpJb,CAAJ,C;QACI,cyKmJkB,wC;QzKIJB,MAAM,gCAAYB,OAAQ,WAAjC,C;;MAFV
,IAAI,EyKqJQ,KAAK,SzKrJb,CAAJ,C;QACI,gByKoJqB,wEAA8D,S;QzKnJnF,MAAM,gCAAYB,SAAQ,WAAjC
,C;;MyKqJN,IAAI,IAAI,CAAR,C;QACI,YAAy,iB;QACZ,UAGbS,C,AhB5B,KAgB6B,GhBf,CAGBe,IAAD,IA
Aa,e;QAdnD,IAAI,QAAQ,GAAZ,C;UACW,OAAP,aAAO,EAAK,IAAL,EAAW,KAAx,EAAkB,eAAIB,C;UACA,
OAAP,aAAO,EAAK,IAAL,EAAW,CAAX,EAAc,GAAd,C;;UAEA,OAAP,aAAO,EAAK,IAAL,EAAW,KAAx,E
AAkB,GAAIB,C;;QAGX,oBAAa,G;QACb,wBAAQ,CAAR,I;;K;qCAKR,wB;MAC8C,QAAC,YAAO,CAAP,IAA
D,IAAa,e;K;;IA9G3D,0C;MAAA,oD;MAA6B,uBAAK,gBAAMb,QAAnB,OAAL,EAAmC,CAAnC,C;MAA7B,Y;
K;ICvFJ,0C;MAII,QAAQ,I;MACR,QAAQ,K;MACR,YAAy,kBAAM,CAAC,OAAO,KAAP,IAAD,IAAiB,CAAjB
,IAAN,C;MACZ,OAAO,KAAK,CAAZ,C;QACI,OtL+B4E,0BsL/BrE,kBAAM,CAAN,CtL0Q2B,KAAL,GAAiB,G
A308B,EsL/B1D,KtL0QgB,KAAL,GAAiB,GA308B,CsL/BrE,IAAP,C;UACI,a;;QACJ,OtL6B4E,0BsL7BrE,kBA
AM,CAAN,CtLwQ2B,KAAL,GAAiB,GA308B,EsL7B1D,KtLwQgB,KAAL,GAAiB,GA308B,CsL7BrE,IAAP,C;
UACI,a;;QACJ,IAAI,KAAK,CAAT,C;UACI,UAAU,kBAAM,CAAN,C;UACV,kBAAM,CAAN,EAAW,kBAAM,
CAAN,CAAX,C;UACA,kBAAM,CAAN,EAAW,GAAX,C;UACA,a;UACA,a;;MAGR,OAAO,C;K;IAGX,uC;MA
GI,YAAy,aAAU,KAAV,EAAiB,IAAjB,EAAuB,KAAvB,C;MACZ,IAAI,QAAO,QAAQ,CAAR,IAAP,CAAJ,C;Q
ACI,UAAU,KAAV,EAAiB,IAAjB,EAAuB,QAAQ,CAAR,IAAvB,C;MACJ,IAAI,QAAQ,KAAZ,C;QACI,UAAU,
KAAV,EAAiB,KAAjB,EAAwB,KAAxB,C;K;IAGR,0C;MAII,QAAQ,I;MACR,QAAQ,K;MACR,YAAy,kBAAM
,CAAC,OAAO,KAAP,IAAD,IAAiB,CAAjB,IAAN,C;MACZ,OAAO,KAAK,CAAZ,C;QACI,OpLM6E,0BoLNtE,
kBAAM,CAAN,CpL0O2B,KAAL,GAAiB,KApO+B,EoLN3D,KpL0OgB,KAAL,GAAiB,KApO+B,CoLNtE,IAAP
,C;UACI,a;;QACJ,OpLI6E,0BoLJtE,kBAAM,CAAN,CpLwO2B,KAAL,GAAiB,KApO+B,EoLJ3D,KpLwOgB,KA
AL,GAAiB,KApO+B,CoLJtE,IAAP,C;UACI,a;;QACJ,IAAI,KAAK,CAAT,C;UACI,UAAU,kBAAM,CAAN,C;U
ACV,kBAAM,CAAN,EAAW,kBAAM,CAAN,CAAX,C;UACA,kBAAM,CAAN,EAAW,GAAX,C;UACA,a;UAC
A,a;;MAGR,OAAO,C;K;IAGX,yC;MAGI,YAAy,aAAU,KAAV,EAAiB,IAAjB,EAAuB,KAAvB,C;MACZ,IAAI,
QAAO,QAAQ,CAAR,IAAP,CAAJ,C;QACI,YAAU,KAAV,EAAiB,IAAjB,EAAuB,QAAQ,CAAR,IAAvB,C;MA
CJ,IAAI,QAAQ,KAAZ,C;QACI,YAAU,KAAV,EAAiB,KAAjB,EAAwB,KAAxB,C;K;IAGR,0C;MAII,QAAQ,I;
MACR,QAAQ,K;MACR,YAAy,kBAAM,CAAC,OAAO,KAAP,IAAD,IAAiB,CAAjB,IAAN,C;MACZ,OAAO,K
AAK,CAAZ,C;QACI,OrLnB8D,YqLmBvD,kBAAM,CAAN,CrLnBwE,KAAjB,EqLmB5C,KrLnByE,KAA7B,CqL
mBvD,IAAP,C;UACI,a;;QACJ,OrLrB8D,YqLqBvD,kBAAM,CAAN,CrLrBwE,KAAjB,EqLqB5C,KrLrByE,KAA
7B,CqLqBvD,IAAP,C;UACI,a;;QACJ,IAAI,KAAK,CAAT,C;UACI,UAAU,kBAAM,CAAN,C;UACV,kBAAM,C
AAN,EAAW,kBAAM,CAAN,CAAX,C;UACA,kBAAM,CAAN,EAAW,GAAX,C;UACA,a;UACA,a;;MAGR,OA
AO,C;K;IAGX,yC;MAGI,YAAy,aAAU,KAAV,EAAiB,IAAjB,EAAuB,KAAvB,C;MACZ,IAAI,QAAO,QAAQ,C
AAR,IAAP,CAAJ,C;QACI,YAAU,KAAV,EAAiB,IAAjB,EAAuB,QAAQ,CAAR,IAAvB,C;MACJ,IAAI,QAAQ,K

AAZ,C;QACI,YAAU,KAAV,EAAiB,KAAjB,EAAwB,KAAxB,C;K;IAGR,0C;MAII,QAAQ,I;MACR,QAAQ,K;M
ACR,YAAAY,kBAAM,CAAC,OAAO,KAAP,IAAD,IAAiB,CAAjB,IAAN,C;MACZ,OAAO,KAAK,CAAZ,C;QAC
I,OrK5C+D,aqK4CxD,kBAAM,CAAN,CrK5C0E,KAAiB,EqK4C7C,KrK5C2E,KAA9B,CqK4CxD,IAAP,C;UACI
,a;;QACJ,OrK9C+D,aqK8CxD,kBAAM,CAAN,CrK9C0E,KAAiB,EqK8C7C,KrK9C2E,KAA9B,CqK8CxD,IAAP,
C;UACI,a;;QACJ,IAAI,KAAK,CAAT,C;UACI,UAAU,kBAAM,CAAN,C;UACV,kBAAM,CAAN,EAAW,kBAA
M,CAAN,CAAX,C;UACA,kBAAM,CAAN,EAAW,GAAX,C;UACA,a;UACA,a;;MAGR,OAAO,C;K;IAGX,yC;
MAGI,YAAAY,aAAU,KAAV,EAAiB,IAAjB,EAAuB,KAAvB,C;MACZ,IAAI,QAAO,QAAQ,CAAR,IAAP,CAAJ,
C;QACI,YAAU,KAAV,EAAiB,IAAjB,EAAuB,QAAQ,CAAR,IAAvB,C;MACJ,IAAI,QAAQ,KAAZ,C;QACI,YA
AU,KAAV,EAAiB,KAAjB,EAAwB,KAAxB,C;K;IAKR,gD;MAI6E,UAAU,KAAV,EAAiB,SAAjB,EAA4B,UAA
U,CAAV,IAA5B,C;K;IAC7E,gD;MAC6E,YAAU,KAAV,EAAiB,SAAjB,EAA4B,UAAU,CAAV,IAA5B,C;K;IAC
7E,gD;MAC6E,YAAU,KAAV,EAAiB,SAAjB,EAA4B,UAAU,CAAV,IAA5B,C;K;IAC7E,gD;MAC6E,YAAU,K
AAV,EAAiB,SAAjB,EAA4B,UAAU,CAAV,IAA5B,C;K;IxK9I7E,0C;MF0BI,IAAI,EEjBI,SAAU,OAAV,GAAiB,
CFiBrB,CAAJ,C;QACI,cAda,qB;QAeb,MAAM,gCAAYB,OAAQ,WAAjC,C;;MEIBV,OAAO,oBAAoB,CAApB,E
AAuB,CAAvB,EAA0B,SAA1B,C;K;IAGX,8C;MACe,Q;MAAX,wBAAW,SAAX,gB;QAAW,SAAX,SAAX,M;Q
ACI,SAAS,GAAG,CAAH,C;QACT,SAAS,GAAG,CAAH,C;QACT,WAAW,cAAc,EAAd,EAAkB,EAAiB,C;QAC
X,IAAI,SAAQ,CAAZ,C;UAAe,OAAO,I;;MAE1B,OAAO,C;K;sGAGX,yB;MAAA,8D;MAAA,iC;QASI,OAAO,c
AAc,SAAS,CAAT,CAAd,EAA2B,SAAS,CAAT,CAA3B,C;O;KATX,C;sGAYA,sC;MASI,OAAO,UAAW,SAAQ,
SAAS,CAAT,CAAR,EAAqB,SAAS,CAAT,CAArB,C;K;IAatB,6B;MAWY,Q;MALR,IAAI,MAAM,CAAV,C;QA
Aa,OAAO,C;MACpB,IAAI,SAAJ,C;QAAe,OAAO,E;MAcTb,IAAI,SAAJ,C;QAAe,OAAO,C;MAGtB,OAA8B,iB
AAtB,mDAAsB,EAAU,CAAV,C;K;IAaZ,6C;MAAA,uB;QAAU,2BAAoB,CAApB,EAAuB,CAAvB,EAA0B,iBA
A1B,C;O;K;IAVhC,8B;MF7CI,IAAI,EEsDI,SAAU,OAAV,GAAiB,CFtDrB,CAAJ,C;QACI,cAda,qB;QAeb,MAA
M,gCAAYB,OAAQ,WAAjC,C;;MEqDV,OAAO,eAAW,2BAAAX,C;K;0FAIX,yB;MAAA,sC;MAAA,oC;MAAA,u
BAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,E
AA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MAPf,2B;QAOI,sBAAW,0BAAAX,C;O;KAPJ,C;0FASA,yB;M
AAA,oC;MAQe,gE;QAAA,uB;UAAU,iBAAsB,kB;UAAtB,eAAkC,gB;UAAIC,OA1Dd,UAAW,SAAQ,SA0DW,C
A1DX,CAAR,EAAqB,SA0DC,CA1DD,CAArB,C;S;O;MAkDtB,uC;QAQI,sBAAW,sCAAX,C;O;KARJ,C;4GAU
A,yB;MAAA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA/
Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN,CAA3B,C;W;S;OA+EL,C;MAPf,2B;QAOI,sBAAW,o
CAAX,C;O;KAPJ,C;8GASA,yB;MAAA,oC;MAUe,0E;QAAA,uB;UAAU,iBAAsB,kB;UAAtB,eAAkC,gB;UAAIC
,OA/Ed,UAAW,SAAQ,SA+EW,CA/EX,CAAR,EAAqB,SA+EC,CA/ED,CAArB,C;S;O;MAqEtB,uC;QAUI,sBAA
W,gDAAX,C;O;KAVJ,C;kFAYA,yB;MAAA,sC;MAAA,oC;MAAA,oBAQe,yB;QA9Gf,8D;eA8Ge,yC;UAAA,uB;
YACP,sBAAsB,WAAy,SAAQ,CAAR,EAAW,CAAX,C;YACIC,Q;YAAA,IAAI,oBAAmB,CAAvB,C;cAAA,OA
A0B,e;;cAAqB,eAAsB,gB;cAArE,OAvgG,cAAc,SAuG8C,CAvG9C,CAAd,EAA2B,SAuGoC,CAvGpC,CAA3B,
C;;YAsGH,W;W;S;OADO,C;MARf,sC;QAQI,sBAAW,kCAAX,C;O;KARJ,C;oFAaA,yB;MAAA,oC;MAQe,0E;Q
AAA,uB;UACP,sBAAsB,WAAy,SAAQ,CAAR,EAAW,CAAX,C;UACIC,Q;UAAA,IAAI,oBAAmB,CAAvB,C;Y
AAA,OAA0B,e;;YAAqB,iBAAsB,kB;YAAtB,eAAkC,gB;YAAjF,OAxGG,UAAW,SAAQ,SAwGyC,CAXGzC,CA
AR,EAAqB,SAwG+B,CAXG/B,CAArB,C;;UAuGd,W;S;O;MATR,kD;QAQI,sBAAW,8CAAX,C;O;KARJ,C;sGAa
A,yB;MAAA,sC;MAAA,oC;MAAA,8BAQe,yB;QAxIf,8D;eAwIe,mD;UAAA,uB;YACP,sBAAsB,qBAAsB,SAA
Q,CAAR,EAAW,CAAX,C;YAC5C,Q;YAAA,IAAI,oBAAmB,CAAvB,C;cAAA,OAA0B,e;;cAAqB,eAAsB,gB;cA
ArE,OAjIG,cAAc,SAiI8C,CAjI9C,CAAd,EAA2B,SAiIoC,CAjIpC,CAA3B,C;;YAghI,W;W;S;OADO,C;MARf,sC
;QAQI,sBAAW,4CAAX,C;O;KARJ,C;wGAaA,yB;MAAA,oC;MAQe,8F;QAAA,uB;UACP,sBAAsB,qBAAsB,SA
AQ,CAAR,EAAW,CAAX,C;UAC5C,Q;UAAA,IAAI,oBAAmB,CAAvB,C;YAAA,OAA0B,e;;YAAqB,iBAAsB,k
B;YAAtB,eAAkC,gB;YAAjF,OAII,UAAW,SAAQ,SakIyC,CAIIzC,CAAR,EAAqB,SakI+B,CAII/B,CAArB,C;;
UAIId,W;S;O;MATR,kD;QAQI,sBAAW,wDAAX,C;O;KARJ,C;kGAcA,yB;MAAA,oC;MAOe,wE;QAAA,uB;UA
CP,sBAAsB,mBAAoB,SAAQ,CAAR,EAAW,CAAX,C;UAA1C,OACI,oBAAmB,CAAvB,GAA0B,eAA1B,GAA+
C,mBAAW,CAAX,EAAc,CAAd,C;S;O;MATvD,wC;QAOI,sBAAW,4CAAX,C;O;KAPJ,C;IAMBe,oD;MAAA,uB
;QACP,sBAAsB,SAAU,SAAQ,CAAR,EAAW,CAAX,C;QAAhC,OACI,oBAAmB,CAAvB,GAA0B,eAA1B,GAA
+C,kBAAW,SAAQ,CAAR,EAAW,CAAX,C;O;K;IATIE,uC;MAOI,sBAAW,kCAAX,C;K;IAYc,wE;MAAA,uB;Q

ACV,sBAAsB,mBAAoB,SAAQ,CAAR,EAAW,CAAX,C;QAA1C,OACI,oBAAmB,CAAvB,GAA0B,eAA1B,GA
A+C,kBAAW,SAAQ,CAAR,EAAW,CAAX,C;O;K;IATIE,+C;MAOI,sBAAc,4CAAd,C;K;IAaW,+C;MAAA,uB;Q
AEH,UAAM,CAAN,C;UADJ,OACe,C;aACX,c;UAFJ,OAEiB,E;aACb,c;UAHJ,OAGiB,C;;UAHjB,OAIY,kBAA
W,SAAQ,CAAR,EAAW,CAAX,C;O;K;IAZ/B,gC;MAOI,sBAAW,6BAAX,C;K;4FASJ,yB;MAAA,4D;MAAA,w
D;MAAA,mB;QAOqE,kBAAW,cAAAX,C;O;KAPrE,C;IAgBe,8C;MAAA,uB;QAEH,UAAM,CAAN,C;UADJ,OAC
e,C;aACX,c;UAFJ,OAEiB,C;aACb,c;UAHJ,OAGiB,E;;UAHjB,OAIY,kBAAW,SAAQ,CAAR,EAAW,CAAX,C;O
;K;IAZ/B,+B;MAOI,sBAAW,4BAAX,C;K;0FASJ,yB;MAAA,4D;MAAA,sD;MAAA,mB;QAOoE,iBAAU,cAAV,
C;O;KAPpE,C;IASA,wB;MAK4F,Q;MAA7B,OAA6B,4F;K;IAE5F,wB;MAK4F,Q;MAA7B,OAA6B,4F;K;IAE5F,
gC;MAM+D,IAEJ,IAFI,EAGJ,M;MAFvD,kBAD2D,SAC3D,sB;QADqD,OAC5B,SAAK,W;WAC9B,WAF2D,SA
E3D,wC;QAFqD,OAEe,4F;WACvD,WAH2D,SAG3D,wC;QAHqD,OAGE,gG;;QAHF,OAI7C,uBAAmB,SAAnB,
C;K;IAIuB,wC;MAAC,4B;K;2CACChC,gB;MAAwC,OAAA,eAAW,SAAQ,CAAR,EAAW,CAAX,C;K;4CACnD,Y
;MACgC,sB;K;;IAGpC,kC;MAAA,sC;K;+CACI,gB;MAAoE,OAAE,iBAAF,CAAE,EAAU,CAAV,C;K;gDACtE,
Y;MAC8C,2C;K;;;IAHID,8C;MAAA,6C;QAAA,4B;;MAAA,sC;K;IAMA,kC;MAAA,sC;K;+CACI,gB;MAAoE,O
AAE,iBAAF,CAAE,EAAU,CAAV,C;K;gDACtE,Y;MAC8C,2C;K;;;IAHID,8C;MAAA,6C;QAAA,4B;;MAAA,sC;
K;8EyKjTA,4B;MAUI,OAAK,iBAAL,SAAK,EAAU,KAAV,C;K;ICTT,iC;K;;;oDAyDI,0C;MAiB+D,oB;QAAA,
2C;aAjB/D,kG;K;;IAoBJ,uC;MAAA,e;MAAA,iB;MAAA,uB;K;IAAA,qC;MAAA,wC;O;MASI,4E;MAMA,8E;M
AOA,4E;MAOA,kE;K;;IApBA,mD;MAAA,2B;MAAA,2C;K;;IAMA,oD;MAAA,2B;MAAA,4C;K;;IAOA,mD;M
AAA,2B;MAAA,2C;K;;IAOA,8C;MAAA,2B;MAAA,sC;K;;IA7BJ,iC;MAAA,+K;K;;IAAA,sC;MAAA,a;aAAA,c;
UAAA,gD;aAAA,e;UAAA,iD;aAAA,c;UAAA,gD;aAAA,S;UAAA,2C;;UAAA,oE;;K;;oFAqCA,mB;K;;;;;;;;;
;;I7HmBiD,gD;MAAA,oB;QACzC,WAAW,sBAAmB,YAAF,CAAE,CAAnB,C;QACX,cAAM,IAAN,C;QADA,O
AEA,IAAK,a;O;K;;;IAthb,+B;K;;iFAUA,yB;MAAA,4B;MAAA,mC;QAMI,6BDgDQ,WChDkB,KDgDIB,CChD
R,C;O;KANJ,C;2GAQA,yB;MAAA,4B;MDgDQ,kD;MChDR,uC;QAOI,6BDgDQ,WAAO,cChDW,SDgDX,CAAP
,CChDR,C;O;KAPI,C;+FAUA,yB;MAAA,kC;MAAA,mD;MAAA,yE;QASI,sC;QAAA,4C;O;MATJ,iGAWY,Y;Q
AAQ,2B;OAXpB,E;MAAA,0DAaQ,kB;QACI,wBAAW,MAAX,C;O;MAdZ,sF;MAAA,sC;QASI,0D;O;KATJ,C;I
AiBA,gD;MAaI,4BAA0D,YAAzC,wCAA6B,UAA7B,CAAYC,CAA1D,EAAyE,yBAAzE,C;K;IAEJ,4D;MAcI,4B
AAoE,YAAAnD,0CAA6B,QAA7B,EAAuC,UAAvC,CAAmD,CAApE,EAAmF,yBAAAnF,C;K;IAEJ,+C;MAU6C,Y
AAzC,wCAA6B,UAA7B,CAAYC,CAtEzC,oBDgDQ,WCSBsD,kBDtBtD,CChDR,C;K;IAyEJ,2D;MAWuD,YAAAn
D,0CAA6B,QAA7B,EAAuC,UAAvC,CAAmD,CAPFnD,oBDgDQ,WCoCgE,kBDpChE,CChDR,C;K;IAuFJ,+C;M
AYI,OAA6C,8BAAtC,c;K;8EAZX,yB;MAAA,oE;MAAA,6E;MA YiD,gD;QAAA,oB;UACzC,WAAW,sBAAmB,
YAAF,CAAE,CAAnB,C;UACX,cAAM,IAAN,C;UADA,OAEA,IAAK,a;S;O;MAfb,sC;QAYW,mBAAsC,8BAAt
C,6B;QAAP,OAAO,kD;O;KAZX,C;qGA0BI,yB;MAAA,2D;MAAA,mB;QACI,MAAM,6BAAoB,0BAApB,C;O;
KADV,C;;M8HzIA,yC;;IAAA,uC;MAAA,2C;K;;IAAA,mD;MAAA,kD;QAAA,iC;;MAAA,2C;K;+EakBA,wB;K
;oDAaA,e;MAK2C,IAAI,IAAJ,EAGK,M;MAL5C,IAAI,+CAAJ,C;QAEI,OAAW,GAAl,kBAAS,IAAK,IAAd,CA
AR,GAA4B,cAAI,OAAJ,GAAl,iBAAQ,IAAR,CAAJ,yCAA5B,GAAyD,I;;MAGpE,OAAW,8CAA4B,GAAhC,GA
AqC,8EAARc,GAAoD,I;K;yDAI/D,e;MAGI,IAAI,+CAAJ,C;QACI,OAAW,GAAl,kBAAS,IAAK,IAAd,CAAJ,IA
A0B,GAAl,iBAAQ,IAAR,CAAJ,QAA9B,GAAyD,mCAAZD,GAAoF,I;;MAE/F,OAAW,8CAA4B,GAAhC,GAAq
C,mCAArC,GAAgE,I;K;;;ICtChD,oD;MACf,cAAc,GAAl,kBAAS,OAAQ,IAAjB,C;MACIB,IAAI,YAAY,mCAA
hB,C;QADA,OACuC,O;;QAEEnC,kBAakB,oBAAQ,yCAAR,C;QACIB,IAAI,mBAAJ,C;UAJJ,OAI6B,oBAAgB,O
AAhB,EAAyB,OAAzB,C;;UACrB,WAAW,OAAQ,kBAAS,yCAAT,C;UAL3B,OAMY,SAAS,mCAAb,GAAoC,o
BAAgB,OAAhB,EAAyB,WAAzB,CAApC,GACI,oBAAgB,oBAAgB,IAAhB,EAAsB,OAAtB,CAAhB,EAAgD,W
AAhD,C;;K;8CAdxB,mB;MAKI,OAAI,YAAY,mCAAhB,GAAuC,IAAvC,GACI,OAAQ,cAAK,IAAL,EAAW,4B
AAX,C;K;;qDAiZC,e;MAEyB,Q;MADrB,OACI,OAAA,IAAK,IAAL,EAAy,GAAZ,CAAJ,GAAqB,0EAARb,G
AAoC,I;K;sDAExC,8B;MACI,iBAAU,OAAV,EAAMB,IAAnB,C;K;0DAEJ,e;MACI,OAAI,OAAA,IAAK,IAAL,E
AAy,GAAZ,CAAJ,GAAqB,mCAArB,GAAgD,I;K;;IC1DP,8C;MAAC,wB;K;kFAAA,Y;MAAA,yB;K;;IAiCe,wD
;MAEjE,kC;MAEA,4BAAqC,mDAAJ,GAakD,OAAQ,qBAA1D,GAA0E,O;K;4DAE3G,mB;MAA6C,+BAAS,OA
AT,C;K;6DAC7C,e;MAA8C,eAAQ,IAAR,IAAgB,8BAAE,G;K;;IAGjF,+C;MAW2C,IAAI,IAAJ,EAGV,M;MAL7
B,IAAI,+CAAJ,C;QAEI,OAAW,GAAl,kBAAS,SAAK,IAAd,CAAR,GAA4B,cAAI,OAAJ,GAAl,iBAAQ,SAAR,C
AAJ,yCAA5B,GAAyD,I;;MAGpE,OAAW,SAAK,IAAL,KAAa,GAAjB,GAAsB,mFAAtB,GAAqC,I;K;IAGhD,6C

;MAUI,IAAI,+CAAJ,C;QACI,OAAW,GAAL,kBAAS,SAAK,IAAd,CAAJ,IAA0B,GAAL,iBAAQ,SAAR,CAAJ,QA
A9B,GAAyD,mCAAZD,GAAoF,S;;MAEF,OAAW,SAAK,IAAL,KAAa,GAAjB,GAASB,mCAATB,GAAiD,S;K;I
AG5D,iC;MAAA,qC;MAKI,4B;K;oDACA,Y;MAAiC,0C;K;kDAEjC,e;MAAyD,W;K;mDACzD,8B;MAA4E,c;K;
mDAC5E,mB;MAAwE,c;K;uDACxE,e;MAA8D,W;K;+CAC9D,Y;MAAsC,Q;K;+CACtC,Y;MAAyC,8B;K;;;IAb7
C,6C;MAAA,4C;QAAA,2B;;MAAA,qC;K;IAqB8B,wC;MAC1B,kB;MACA,wB;K;4CAGA,e;MAGQ,Q;MAFJ,U
AAU,I;MACV,OAAO,IAAP,C;QACI,YAAA,GAAL,UAAJ,aAAY,GAAZ,W;UAAwB,W;;QACxB,WAAW,GAAL,
O;QACf,IAAI,oCAAJ,C;UACI,MAAM,I;;UAEN,OAAO,iBAAK,GAAL,C;;;K;6CAKnB,8B;MACI,iBAAU,WAA
K,cAAK,OAAL,EAAC,SAAd,CAAf,EAAYC,cAAzC,C;K;iDAEJ,e;UAGW,I;MAFP,+BAAQ,GAAR,U;QAAoB,O
AAO,W;;MAC3B,cAAc,WAAK,kBAAS,GAAT,C;MAEf,gBAAy,WAAZ,C;QAAoB,W;WACpB,gBAAy,mCAA
Z,C;QAAqC,qB;;QAC7B,2BAAgB,OAAhB,EAAYB,cAAzB,C;MAHZ,W;K;uCAOJ,Y;MAIc,IAAI,IAAJ,Q;MAH
V,UAAU,I;MACV,WAAW,C;MACX,OAAO,IAAP,C;QACU,uBAAI,OAAJ,GAAL,OAAJ,gC;QAAA,mB;UAAgC
,OAAO,I;;QAA7C,MAAM,M;QACN,mB;;K;2CAIR,mB;MACI,+BAAI,OAAQ,IAAZ,GAAoB,OAAPB,C;K;8CA
EJ,mB;MAQ4B,Q;MAPxB,UAAU,O;MACV,OAAO,IAAP,C;QACI,IAAI,CAAC,gBAAS,GAAL,UAAb,CAAL,C;
UAA4B,OAAO,K;QACnC,WAAW,GAAL,O;QACf,IAAI,oCAAJ,C;UACI,MAAM,I;;UAEN,OAAO,gBAAS,0EA
AT,C;;;K;uCAKnB,iB;MACI,gBAAS,KAAT,KAakB,yCAA4B,KAAM,SAAN,KAAgB,aAA5C,IAAsD,KAAM,e
AAY,IAAZ,CAA9E,C;K;yCAEJ,Y;MAA+B,OAkK,SAAL,WAAK,CAAL,GAA0B,SAAR,cAAQ,CAA1B,I;K;IA
GZ,uD;MACX,OAAI,G5JyHoC,YAAU,C4JzHID,GAAMB,OAAQ,WAA3B,GAA6C,GAAF,UAAQ,O;K;yCAF3D
,Y;MACI,aAAM,kBAAK,EAAL,EAAS,+BAAT,CAAN,GAEL,G;K;IAMO,8E;MAAA,6B;QAAyB,Q;QAAT,iBA
AS,sBAAT,EAAS,8BAAT,UAAoB,O;QAAQ,W;O;K;+CAJ3D,Y;MAOsB,Q;MANIB,QAAQ,a;MACR,eAAe,gBA
A+B,CAA/B,O;MACf,gBAAy,CAAZ,C;MACA,kBAAK,kBAAL,EAAW,oDAAX,C;M/KtFJ,IAAI,E+KuFM,YA
AS,C/KvFf,CAAJ,C;QACI,cAdW,e;QAeX,MAAM,6BAAsB,OAAQ,WAA9B,C;;M+KuFN,OAAO,+BAAW,qDA
AX,C;K;IAGa,8C;MACpB,kD;MADqB,wB;K;IACrB,gD;MAAA,oD;MACI,4B;K;;;IADJ,4D;MAAA,2D;QAAA,0
C;;MAAA,oD;K;yDAIA,Y;MAA0C,gBAAT,a;M7Lm9YrB,Q;MADhB,kB6Ll9YmD,mC;M7Lm9YnD,wBAAGB,S
AAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,cAAwB,yBAAa,OAAb,C;;M6Ln9YT,O7Lo9Y9B,W;K;;;I8LtoZX,o
E;MA4BI,MAAM,wBAAoB,sEAApB,C;K;8GA5BV,yB;MAAA,2D;MAAA,sC;QA4BI,MAAM,6BAAoB,sEAAp
B,C;O;KA5BV,C;IA0CoC,mC;MAAQ,4D;K;IAE5C,4C;MAAA,e;MAAA,iB;MAAA,uB;K;IAAA,0C;MAAA,6C;
O;MAK0C,oG;MAAQb,gF;MAAW,4E;K;;IAAhC,+D;MAAA,gC;MAAA,uD;K;;IAAQb,qD;MAAA,gC;MAAA,6
C;K;;IAAW,mD;MAAA,gC;MAAA,2C;K;;IAL1E,sC;MAAA,sJ;K;;IAAA,2C;MAAA,a;aAAA,qB;UAAA,4D;aA
AA,W;UAAA,kD;aAAA,S;UAAA,gD;;UAAA,qF;;K;;6ECnDA,yB;MAAA,0B;MAAA,mC;QAGsD,OAAiC,OAA
3B,SAAL,GAAuB,KAAS,C;O;KAHvF,C;2EAKA,yB;MAAA,0B;MAAA,mC;QAGqD,OAAgC,OAA1B,SAAL,G
AAsB,KAAS,C;O;KAHrF,C;6EAKA,yB;MAAA,0B;MAAA,mC;QAGsD,OAAiC,OAA3B,SAAL,GAAuB,KAAS,
C;O;KAHvF,C;6EAKA,yB;MAAA,0B;MAAA,4B;QAGqC,OAAqB,OAAP,CAAR,SAAE,C;O;KAH1D,C;+EAMA
,yB;MAAA,4B;MAAA,mC;QAGyD,OAAiC,QAA3B,SAAL,GAAuB,KAAS,C;O;KAH1F,C;6EAKA,yB;MAAA,4
B;MAAA,mC;QAGwD,OAAgC,QAA1B,SAAL,GAASB,KAAS,C;O;KAHxF,C;+EAKA,yB;MAAA,4B;MAAA,m
C;QAGyD,OAAiC,QAA3B,SAAL,GAAuB,KAAS,C;O;KAH1F,C;+EAKA,yB;MAAA,4B;MAAA,4B;QAGuC,O
AAqB,QAAP,CAAR,SAAE,C;O;KAH5D,C;ICpCA,qC;K;;ICAA,mB;K;;IAOA,iB;K;;IAOA,2C;K;;IAOA,wB;K;;I
AQA,0B;K;;IAOA,sB;K;;IAOA,4B;K;;IAOA,6C;K;;IA+BuC,wE;MAEnC,uB;QAAA,UAAsB,E;MACtB,qB;QAA
A,8B;MACA,2B;QAAA,qE;MACA,yB;QAAA,YAAqB,E;MAJrB,sB;MACA,sB;MACA,kB;MACA,8B;MACA,0
B;K;;IAGJ,iD;MAAA,e;MAAA,iB;MAAA,uB;K;IAAA,+C;MAAA,kD;O;MAKI,wG;MACA,wG;MACA,8F;K;;I
AFA,iE;MAAA,qC;MAAA,yD;K;;IACA,iE;MAAA,qC;MAAA,yD;K;;IACA,4D;MAAA,qC;MAAA,oD;K;;IAPJ,2
C;MAAA,6K;K;;IAAA,gD;MAAA,a;aAAA,kB;UAAA,8D;aAAA,kB;UAAA,8D;aAAA,a;UAAA,yD;;UAAA,6E;;
K;;IAUA,wB;K;;ICjGA,qB;MAAA,yB;K;0CAII,Y;MAO6D,uB;K;2HAE7D,yB;MAAA,+D;MAAA,kC;MAAA,0F
;MAAA,6F;MAAA,4E;QAUI,wC;QAAS,2C;O;MAVb,mEAWQ,wC;QAA6E,sBAAS,QAAT,EAAMB,QAANB,EA
A6B,QAA7B,C;O;MAXrF,oG;MAAA,yC;QAUI,wDAA+B,YAA/B,C;O;KAVJ,C;uHAcA,yB;MAAA,+D;MAAA,
kC;MAAA,wF;MAAA,yF;MAAA,0E;QAcI,wC;QAAS,2C;O;MAdB,kEAeQ,wC;QAAuF,6BAAS,QAAT,EAAMB,
QAANB,EA6B,QAA7B,C;O;MAf/F,kG;MAAA,yC;QAcI,sDAA+B,YAA/B,C;O;KADJ,C;;IA3BJ,iC;MAAA,gC;
QAAA,e;;MAAA,yB;K;IAGDiC,sB;MAC7B,eAAwB,I;K;4CAExB,6B;MACW,Q;MAAA,mB;MAAA,iB;QAAS,M
AAM,6BAAsB,cAAY,QAAS,aAARb,uCAATB,C;;MAATB,OAAO,I;K;4CAGX,oC;MACI,eAAa,K;K;;;kDC9CjB,

6B;;K;,,,,,;iEA+CA,6B;;K;;ICrDuC,0C;MACvC,uBAAoB,Y;K;wDAEpB,wC;MAM6F,W;K;uDAE7F,wC;K;oDA
MA,6B;MACI,OAAO,oB;K;oDAGX,oC;MACI,eAAe,IAAK,gB;MACpB,IAAI,CAAC,0BAAa,QAAb,EAAuB,QA
AvB,EAAiC,KAAjC,CAAL,C;QACI,M;;MAEJ,uBAAa,K;MACb,yBAAY,QAAZ,EAAzB,QAAtB,EAAgC,KAAh
C,C;K;;4EC9BR,wC;MAqBI,OAAO,e;K;4EAGX,+C;MAuBI,cAAI,KAAJ,C;K;4EAIJ,wC;MAMBI,OAAO,cAAI,
OAAJ,C;K;4EAGX,+C;MAqBI,cAAI,OAAJ,EAAa,KAAb,C;K;IC/FJ,kB;MA6PI,4B;K;+BAtoA,Y;MAOiC,6BAA
S,EAAT,C;K;uCAEjC,iB;MAW2C,4BAAQ,CAAR,EAAW,KAAx,C;K;uCAE3C,uB;MAakB,Q;MAHd,iBAAiB,I
AAjB,EAAuB,KAAvB,C;MACA,QAAQ,QAAQ,IAAR,I;MACR,IAAI,IAAI,CAAJ,IAAS,MAAK,WAAIB,C;QAC
c,IAAI,MAAM,CAAC,CAAD,IAAN,OAAy,CAAhB,C;UACN,eAAe,SAAS,CAAT,C;UACf,6BAAS,QAAT,C;;U
AEA,K;;YAEI,WAAW,cAAU,KAAK,C;YAC1B,IAAI,OAAO,C;;UACN,gBAAO,CAAP,IAAY,CAAZ,GAAgB,C
AAhB,SAAqB,CAArB,C;UACT,Q;;QATJ,c;QAWA,OAAO,OAAO,GAAP,I;;QAEp,OAAO,IAAP,C;UACI,YAA
U,c;UACV,IAAW,IAAP,qBAakB,KAAtB,C;YAA6B,OAAO,K;;K;gCAKhD,Y;MAOmC,OAAU,oBAAV,cAAU,
CAAS,WAAI,EAAJ,CAAnB,yBAA6B,cAA7B,E;K;wCAEnC,iB;MAW8C,iCAAY,KAAZ,C;K;wCAE9C,uB;MAi
BkB,Q;MAPd,mBAAiB,IAAjB,EAAuB,KAAvB,C;MACA,QAAQ,eAAQ,IAAR,C;MACR,IAAI,eAAI,CAAR,C;Q
ACI,O;QACA,IAAI,aAAO,CAAD,aAAN,GAAY,CAAZ,CAAJ,C;UACI,WAAW,CAAe,Q;UACb,YAAa,qBAAO,
EAAP,CAAW,Q;UAEpB,aAAQ,CAAR,C;YACI,eAAe,SAAS,IAAT,C;YAEf,OAAmB,oBAAAnB,sBAAS,QAAT,C
AAmB,CAAnB,iB;iBAEJ,cAAS,CAAT,C;YAEI,OAAU,oBAAV,cAAU,CAAV,iB;;YAEA,iBAae,SAAS,KAAT,
C;YACf,OAAmB,oBAAAnB,sBAAS,UAAT,CAAmB,CAAS,WAAI,EAAJ,CAA5B,KAAiD,oBAAV,cAAU,CAAV,
iBAAvC,C;;UAXR,U;;UAeA,K;;YAEI,WAAW,eAAW,oBAAK,CAAL,C;YACtB,IAAI,YAAO,CAAP,C;;UACC,s
BAAO,CAAP,MAAY,+BAAI,CAAJ,EAAZ,eAAqB,CAArB,C;UACT,MAAM,C;;QAEV,OAAO,SAAO,GAAP,C;
;QAEp,OAAO,IAAP,C;UACI,YAAU,e;UACV,IAAW,IAAP,0CAakB,KAAIB,CAAJ,C;YAA6B,OAAO,K;;K;m
CAKhD,Y;MAKyC,6BAAS,CAAT,MAAe,C;K;kCAExD,Y;MAKuC,uBAAgB,sBAAS,EAAT,CAAhB,EAA8B,sB
AAS,EAAT,CAA9B,C;K;0CAEvC,iB;MASoD,+BAAW,GAAX,EAAgB,KAAhB,C;K;0CAEpD,uB;MAcY,Q;MA
FR,mBAAiB,IAAjB,EAAuB,KAAvB,C;MACA,WAAW,QAAQ,I;MACX,IAAS,WAAI,IAAK,CAAL,IAA0B,SA
AL,IAAK,CAA1B,IAA8C,SAAN,KAAM,CAAI,D,C;QACJ,SAAS,qBAAgB,QAAQ,CAAR,GAAY,OAAO,CAAn
C,C;QACT,cAAO,EAAP,GAAY,E;;QAEZ,cAAO,oBAAe,I;;MAJ1B,Y;MAMA,OAAW,KAAK,KAAT,GAASB,S
AAN,KAAM,CAATB,GAAS,C;K;iCAGjD,Y;MAKqC,6BAAS,EAAT,IAA0B,Q;K;IAWK,oF;MAAA,mB;QAAE,
uBAAa,iBAAb,sBAAqC,eAARc,+BAAqE,aAAM,OAA3E,M;O;K;iDATtE,qC;MxLjLA,IAAI,EwL0LqB,CAAb,8
BAAgB,KAAM,OxL1L9B,GwL0LiD,CAAX,0BAAc,KAAM,OxL1L1D,GwL0LsC,KxL1LtC,CAAJ,C;QACI,cwL
yLgE,kDxLzLID,E;QACd,MAAM,gCAAYB,OAAQ,WAAjC,C;;MAFV,IAAI,EwL2LQ,aAAa,OxL3LrB,CAAJ,C;
QACI,gBwL0LgC,mF;QxLzLhC,MAAM,gCAAYB,SAAQ,WAAjC,C;;MwL2LN,YAAY,CAAC,UAAU,SAAV,IA
AD,IAAwB,CAAxB,I;MAEZ,mBAAe,SAAf,C;MrLzEJ,iBAAc,CAAd,UqL0EW,KrL1EX,U;QqL2EQ,QAAQ,c;Q
ACR,MAAM,UAAN,IAAoB,OAAF,CAAe,C;QACpB,MAAM,aAAW,CAAX,IAAN,IAAgC,OAAV,CAAe,KAA
K,CAAG,C;QACChC,MAAM,aAAW,CAAX,IAAN,IAAiC,OAAx,CAAe,KAAK,EAAL,C;QACjC,MAAM,aAAW,
CAAX,IAAN,IAAiC,OAAx,CAAe,KAAK,EAAL,C;QACjC,0BAAy,CAAZ,I;;MAGJ,gBAAGB,UAAU,UAUV,I;
MACHB,SAAS,sBAAS,YAAY,CAAZ,IAAT,C;MACT,aAAU,CAAV,MAAkB,SAAI,B,M;QACI,MAAM,aAAW,C
AAX,IAAN,IAAqC,OAAf,EAAG,MAAK,IAAI,CAAJ,IAAL,CAAY,C;;MAGzC,OAAO,K;K;yCACX,uD;MAvB4
C,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,KAAM,O;aARrF,0H;K;yCAiCA,iB;MAOyD,8BAAU,KAAV,E
AAiB,CAAjB,EAAoB,KAAM,OAA1B,C;K;yCAEzD,gB;MAKkd,8BAAU,cAAU,IAAV,CAAV,C;K;IAGID,0B;
MAAA,8B;MAO2B,iB;MACvB,uBAAoC,uB;K;IAEpC,qC;MAAA,yC;MACI,4B;K;wDAEA,Y;MAAiC,mC;K;;I
AHrC,iD;MAAA,gD;QAAA,+B;;MAAA,yC;K;8CAMA,Y;MAAkC,8C;K;gDAEIC,oB;MAA4C,OAAA,oBAAc,k
BAAS,QAAT,C;K;uCAC1D,Y;MAA8B,OAAA,oBAAc,U;K;+CAC5C,iB;MAAwC,OAAA,oBAAc,iBAAQ,KAA
R,C;K;+CACtD,uB;MAAmD,OAAA,oBAAc,iBAAQ,IAAR,EAAc,KAAc,C;K;wCAEjE,Y;MAAgC,OAAA,oBAA
c,W;K;gDAC9C,iB;MAA2C,OAAA,oBAAc,kBAAS,KAAT,C;K;gDACzD,uB;MAAuD,OAAA,oBAAc,kBAAS,I
AAT,EAAe,KAAf,C;K;2CAErE,Y;MAAsC,OAAA,oBAAc,c;K;0CAEpD,Y;MAAoC,OAAA,oBAAc,a;K;kDACID
iB;MAAiD,OAAA,oBAAc,oBAAW,KAAx,C;K;kDAC/D,uB;MAA+D,OAAA,oBAAc,oBAAW,IAAX,EAAiB,K
AAjB,C;K;yCAE7E,Y;MAAkC,OAAA,oBAAc,Y;K;iDAeHD,iB;MAAsD,OAAA,oBAAc,mBAAU,KAAV,C;K;iD
ACpE,gB;MAA+C,OAAA,oBAAc,mBAAU,IAAV,C;K;yDAC7D,qC;MACI,OAAA,oBAAc,mBAAU,KAAV,EA
AiB,SAAjB,EAA4B,OAA5B,C;K;;IAtCtB,sC;MAAA,qC;QAAA,oB;;MAAA,8B;K;;IA0CJ,wB;MAAuC,yBAAa,I

AAb,EAAmB,IAAK,IAAI,EAA5B,C;K;IAEvC,wB;MAawC,yBAAa,IAAK,QAAIB,EAA2B,IAAK,YAAI,EAAJ,C
AAQ,QAAxC,C;K;IAGxC,mC;MAUI,IAAA,KAAM,UAAAN,C;QAAmB,MAAM,gCAAyB,uCAAoC,KAA7D,C;
WACzB,IAAA,KAAM,KAAN,GAAa,UAAb,C;QAF8C,OAEhB,0BAAQ,KAAM,MAAd,EAAqB,KAAM,KAAN,
GAAa,CAAb,IAArB,C;WAC9B,IAAA,KAAM,MAAN,GAAC,WAAd,C;QAH8C,OAGf,0BAAQ,KAAM,MAAN,
GAAC,CAAd,IAAR,EAAyB,KAAM,KAA/B,IAAuC,CAAvC,I;QAHe,OAIc,mB;K;IAGZ,oC;MAUI,IAAA,KAA
M,UAAAN,C;QAAmB,MAAM,gCAAyB,uCAAoC,KAA7D,C;WACzB,IAAA,KAAM,KAAN,+C;QAFiD,OAEIB,2
BAAS,KAAM,MAAf,EAA5B,KAAM,KAAN,yBAAa,CAAb,EAAtB,C;WAC/B,IAAA,KAAM,MAAN,+C;QAHi
D,OAGjB,2BAAS,KAAM,MAAN,8BAAC,CAAd,EAAT,EAA0B,KAAM,KAAhC,0BAAwC,CAAxC,E;QAHiB,
OAIzC,oB;K;IAOZ,yB;MAAyC,YjFrTkB,MAAO,OiFqTpB,KjFrToB,CiFqTzB,I;K;IAEzC,4C;MAEI,OAAA,SAA
K,KAAK,EAAL,GAAU,QAAf,GAAyC,CAAX,CAAC,QAAD,IAAW,KAAI,E;K;IAEjD,uC;MxLtVI,IAAI,EwLsV
uD,QAAQ,IxLtV/D,CAAJ,C;QACI,cwLqVuE,+B;QxLpVvE,MAAM,gCAAyB,OAAQ,WAAjC,C;K;IwLqVd,yC;
MxLvVI,IAAI,EwLuVyD,sBAAQ,IAAR,KxLvVzD,CAAJ,C;QACI,cwLsVyE,+B;QxLrVzE,MAAM,gCAAyB,OA
AQ,WAAjC,C;K;IwLsVd,yC;MxLxVI,IAAI,EwLwV6D,QAAQ,IxLxVrE,CAAJ,C;QACI,cwLuV6E,+B;QxLrV7E
,MAAM,gCAAyB,OAAQ,WAAjC,C;K;IwLwVd,yC;MAAYD,oCAA0B,IAA1B,qBAAiC,KAAjC,kB;K;ICrXzD,6
B;MAOqC,OpMmYE,SoMnYF,mBpMmYE,C;K;IoMjYvC,sC;MASgD,6BAAS,WAAT,EAAa,KAAb,C;K;IAEhD
,4C;MAUI,qBAAqB,IAArB,EAA2B,KAA3B,C;MAEA,iBAAiB,IpMqQgB,KoMrQhB,GAAiB,W;MACiC,kBAAk
B,KpMoQe,KoMpQf,GAAkB,W;MAEpC,mBAAmB,0BAAQ,UAAAR,EAAoB,WAApB,IAAqC,W;MACxD,OpMs
WmC,SoMtW5B,YpMsW4B,C;K;IoMnWvC,sC;MAWI,IAAA,KAAM,UAAAN,C;QAAmB,MAAM,gCAAyB,uCA
AoC,KAA7D,C;QACzB,IpMGkE,YoMHIE,KAAM,KpMG6E,KAAjB,EoMHRD,4BAAK,UpMG6E,KAA7B,CoM
HIE,K;UAFiD,OAEIB,sBAAS,KAAM,MAAf,EpMqBsB,SoMrBA,KAAM,KpMqBI,KAAK,GAAW,CoMrBb,Wp
MqBa,MAAX,IAAf,CoMrBtB,C;UAC/B,IpMEkE,YoMFIE,KAAM,MpME6E,KAAjB,EoMFPD,4BAAK,UpME4
E,KAA7B,CoMFIE,K;YAHiD,OpMuBI,SoMpBrB,sBpMiCsB,SoMjCb,KAAM,MpMiCiB,KAAK,GAAy,CoMjC1
B,WpMiC0B,MAAZ,IAAf,CoMjCtB,EAA2B,KAAM,KAAjC,CpMoB+B,KAAK,GAAW,CoMpBN,WpMoBM,M
AAX,IAAf,C;YoMvBJ,OAIzC,mB;K;IAGZ,8B;MAOuC,OpL0VG,UoL1VH,oBpL0VG,C;K;IoLxV1C,uC;MAS
mD,8BAAU,2BAAV,EAAe,KAAf,C;K;IAEnD,6C;MAUI,sBAAsB,IAAtB,EAA4B,KAA5B,C;MAEA,iBAAiB,Ip
LwNkB,KoLxNIB,8B;MACjB,kBAAkB,KpLuNiB,KoLvNjB,8B;MAEIB,mBAAmB,2BAAS,UAAAT,EAAqB,WA
ArB,+B;MACnB,OpL6TsC,UoL7T/B,YpL6T+B,C;K;IoL1T1C,uC;MAWI,IAAA,KAAM,UAAAN,C;QAAmB,MA
AM,gCAAyB,uCAAoC,KAA7D,C;QACzB,IpL7CmE,aoL6CnE,KAAM,KpL7C+E,KAAIB,EoL6CtD,6BAAM,Up
L7C8E,KAA9B,CoL6CnE,K;UAFoD,OAEPB,uBAAU,KAAM,MAAhB,EpLhCuB,UoLgCA,KAAM,KpLhCK,KA
AK,KAAW,ChBsQ7C,UAAW,oBAAL,CoMtOyB,WpMsOzB,MAAK,CAAL,iBAAN,CgBtQ6C,MAAX,CAAhB,
CoLgCvB,C;UACHC,IpL9CmE,aoL8CnE,KAAM,MpL9C+E,KAAIB,EoL8CrD,6BAAM,UpL9C6E,KAA9B,CoL
8CnE,K;YAHoD,OpL9BG,UoLiCtB,uBpLpBuB,UoLoBb,KAAM,MpLpBkB,KAAK,UAAy,ChByP/C,UAAW,oB
AAL,CoMrOc,WpMqOd,MAAK,CAAL,iBAAN,CgBzP+C,MAAZ,CAAhB,CoLoBvB,EAA4B,KAAM,KAAIC,Cp
LjCiC,KAAK,KAAW,ChBsQ7C,UAAW,oBAAL,CoMrOgC,WpMqOhC,MAAK,CAAL,iBAAN,CgBtQ6C,MAA
X,CAAhB,C;YoL8BH,OAI5C,oB;K;IAGZ,sC;MAQI,4BAAU,KjK4+FH,QiK5+FP,C;MACA,OAAO,K;K;IAGX
,uC;MAKsD,OjK2iG3C,eiK3iG2C,4BAAU,IAAV,CjK2iG3C,C;K;IiKziGX,4D;MAOGD,yB;QAAA,YAAiB,C;M
AAG,uB;QAAA,UAAe,KAAM,K;MACrF,4BAAU,KjKy9FH,QiKz9FP,EAA+B,SAA/B,EAA0C,OAA1C,C;MAC
A,OAAO,K;K;IAIX,2C;MzLrHI,IAAI,EX2B8D,YoM0FD,KpM1FkB,KAAjB,EoM0FO,IpM1FsB,KAA7B,CoM0F
D,IzLrH7D,CAAJ,C;QACI,cyLoH6E,+B;QzLnH7E,MAAM,gCAAyB,OAAQ,WAAjC,C;K;IyLoHd,4C;MzLrHI,I
AAI,EKmc+D,aoLmFC,KpLnFiB,KAAIB,EoLmFS,IpLnFqB,KAA9B,CoLmFC,IzLrHhE,CAAJ,C;QACI,cyLqHg
F,+B;QzLpHhF,MAAM,gCAAyB,OAAQ,WAAjC,C;K;IoLpBc,6C;MAScxB,oC;MA/BA,iB;MANA,Y;MACA,Y;
MACA,Y;MACA,Y;MACA,Y;MACA,sB;MILYA,IAAI,E0LLQ,CAAC,WAAK,QAAL,GAAU,QAAV,GAAe,QA
Af,GAAoB,QAARB,MAA2B,C1LKnC,CAAJ,C;QACI,c0LNwC,wD;Q1LOxC,MAAM,gCAAyB,OAAQ,WAAjC,
C;MGoHV,iBAAC,CAAd,UuLxHW,EvLwHX,U;QuLxHiB,c;K;qCAGjB,Y;MAGI,QAAQ,Q;MACR,IAALI,IAAO,
MAAO,C;MACIB,WAAI,Q;MACJ,WAAI,Q;MACJ,WAAI,Q;MACJ,SAAS,Q;MACT,WAAI,E;MACJ,IAAK,IAA
O,KAAM,CAAd,GAA5B,EAAtB,GAA8B,MAAO,C;MACzC,WAAI,C;MACJ,gCAAU,MAAV,I;MACA,OAAO,I
AAI,aAAJ,I;K;8CAGX,oB;MACI,OAAU,cAAV,cAAU,EAAc,QAAc,C;K;IAEd,kC;MAAA,sC;MACI,4B;K;IAD
J,8C;MAAA,6C;QAAA,4B;MAAA,sC;K;IA7BA,gD;MAAA,sD;MACQ,yBAAK,KAAAL,EAAy,KAAZ,EAAmB,

CAAnB,EAAsB,CAAtB,EAA+B,CAAN,KAAzB,EAaUc,SAAU,EAAX,GAAoB,UAAW,CAArE,C;MADR,Y;K;I
CbiD,8C;MACjD,4B;MACA,0C;K;oEADA,Y;MAAA,2B;K;2EACA,Y;MAAA,kC;K;uCAGA,iB;MACI,OAAO,0
CAAgC,kBAaA,KAAM,UAAAnB,KAC/B,mBAAS,KAAM,MAAf,KAawB,0BAAgB,KAAM,aAAtB,CADO,CAA
hC,C;K;yCAIX,Y;MACI,OAAW,cAAJ,GAAe,EAaf,GAAuB,MAAW,SAAN,UAAAM,CAAX,QAAqC,SAAb,iBA
Aa,CAArC,I;K;yCAGlC,Y;MAAkC,OAAE,UAAf,qBAAU,iB;K;;IAGhD,kC;MAM6E,2BAAgB,SAAhB,EAAsB,I
AAtB,C;K;;;0DAYzE,iB;MAA2C,qCAAiB,UAAjB,EAawB,KAAxB,KAAkC,8BAAiB,KAAjB,EAawB,iBAAXB,
C;K;;iDAC7E,Y;MAAkC,QAAC,8BAAiB,UAAjB,EAawB,iBAAXB,C;K;;IAcR,gD;MAI3B,gBAAqB,K;MACrB,u
BAA4B,Y;K;0FACD,Y;MAAQ,oB;K;iGACD,Y;MAAQ,2B;K;2DAE1C,gB;MAA+D,YAAK,C;K;mDAEpE,iB;M
AAgD,gBAAS,aAAT,IAAmB,SAAS,oB;K;0CAC5E,Y;MAAkC,SAAE,iBAAU,oBAAZ,C;K;yCAEIC,iB;MACI,O
AAO,4CAA+B,kBAaA,KAAM,UAAAnB,KAC9B,kBAAU,KAAM,SAAhB,IAA0B,yBAAiB,KAAM,gBADnB,CA
A/B,C;K;2CAIX,Y;MACI,OAAW,cAAJ,GAAe,EAaf,GAAuB,MAAY,SAAP,aAAO,CAAZ,QAAuC,SAAd,oBAA
c,CAAvC,I;K;2CAGlC,Y;MAAkC,OAAE,aAAf,qBAAW,oB;K;;IAGjD,oC;MAOqF,6BAAkB,SAaIB,EAawB,IA
AxB,C;K;IAQvD,+C;MAI1B,gBAAqB,K;MACrB,uBAA4B,Y;K;yFACF,Y;MAAQ,oB;K;gGACD,Y;MAAQ,2B;
K;0DAEzC,gB;MAA6D,YAAK,C;K;kDAEIE,iB;MAA+C,gBAAS,aAAT,IAAmB,SAAS,oB;K;yCAC3E,Y;MAAk
C,SAAE,iBAAU,oBAAZ,C;K;wCAEIC,iB;MACI,OAAO,2CAA8B,kBAaA,KAAM,UAAAnB,KAC7B,kBAAU,KA
AM,SAAhB,IAA0B,yBAAiB,KAAM,gBADpB,CAA9B,C;K;0CAIX,Y;MACI,OAAW,cAAJ,GAAe,EAaf,GAAu
B,MAAY,SAAP,aAAO,CAAZ,QAAuC,SAAd,oBAAc,CAAvC,I;K;0CAGlC,Y;MAAkC,OAAE,aAAf,qBAAW,o
B;K;;IAGjD,oC;MAOKF,4BAAiB,SAAjB,EAauB,IAAvB,C;K;oFAGlF,8B;MAQI,0BAAmB,2BAAS,OAAT,C;K;
IAGvB,+C;MACI,IAAI,CAAC,UAAAL,C;QAAiB,MAAM,gCAAyB,iCAA8B,IAA9B,iBAAzB,C;K;IC5I3B,gC;MA
cW,Q;MADP,IAAI,CAAC,6BAAW,KAAX,CAAL,C;QAAwB,MAAM,uBAAmB,sC/EjBzC,oB+EiByC,CAAnB,C
;;MAC9B,OAAO,sD;K;IAMX,oC;MAakC,Q;MAA9B,OAAW,6BAAW,KAAX,CAAJ,GAAuB,sDAAvB,GAAuC,
I;K;;;;;ICvBhB,yC;MA2B9B,uC;MA1BA,wB;MAIA,gB;M7LQA,IAAI,E6LDS,iBAAY,IAAb,MAAuB,iBAAvB,
C7LCR,CAAJ,C;QACI,c6LDQ,iBAAY,IAAhB,GACI,8CADJ,GAGI,sCAA0B,aAA1B,qC;Q7LDR,MAAM,gCAA
yB,OAAQ,WAAjC,C;;K;yC6LKV,Y;MAAwC,Q;MAAA,oB;MACpC,iB;QAD8B,OActB,G;WACR,oD;QAF8B,O
AEF,SAAL,SAAK,C;WAC5B,6C;QAH8B,OAGd,iBAAK,SAAL,C;WACHB,8C;QAJ8B,OAIb,kBAAM,SAAN,C;;
QAJa,mC;K;IAOIC,qC;MAAA,yC;MACI,YAGqC,oBAAgB,IAAhB,EAAsB,IAAtB,C;K;iGAQJ,Y;MAAQ,gB;K;4
DAEzC,gB;MAOI,8DAAqC,IAArC,C;K;gEAEl,gB;MAMI,uDAA8B,IAA9B,C;K;4DAEJ,gB;MAMI,wDAA+B,I
AA/B,C;K;;;IArCR,iD;MAAA,gD;QAAA,+B;;MAAA,yC;K;;2CArCJ,Y;MAWI,oB;K;2CAXJ,Y;MAeI,gB;K;6CA
fJ,0B;MAAA,2BAWI,8CAXJ,EAeI,kCAfJ,C;K;yCAAA,Y;MAAA,c;MAWI,yD;MAIA,qD;MAfJ,a;K;uCAAA,iB;
MAAA,4IAWI,4CAXJ,IAeI,oCAfJ,I;K;ICLA,kC;MAAA,e;MAAA,iB;MAAA,uB;K;IAAA,gC;MAAA,mC;O;MA
YI,4D;MAKA,8C;MAKA,gD;K;;IAVA,2C;MAAA,sB;MAAA,mC;K;;IAKA,oC;MAAA,sB;MAAA,4B;K;;IAKA,
qC;MAAA,sB;MAAA,6B;K;;IAtBJ,4B;MAAA,mG;K;;IAAA,iC;MAAA,a;aAAA,W;UAAA,wC;aAAA,I;UAAA,i
C;aAAA,K;UAAA,kC;;UAAA,6D;;K;;6ECAa,yB;MAAA,4F;MAAA,2B;QASI,MAAM,mCAA8B,0EAA9B,C;O;
KATV,C;ICkCA,+D;MAaW,Q;MAAP,OAAO,8CAA0,KAAP,EAAC,UAAAd,EAA0B,QAA1B,oC;K;IAGX,kC;MA
IiB,Q;MAAb,wBAaA,KAAb,gB;QAAa,WAAA,KAAb,M;QACI,yBAAO,IAAP,C;;MACJ,OAAO,S;K;mFAGX,qB
;MAGwD,gCAAO,EAAP,C;K;qFAExD,4B;MAG4E,OAAA,yBAAO,KAAP,CALpB,gBAAO,EAAP,C;K;qFAOx
D,4B;MAGmE,OAAA,yBAAO,KAAP,CAVX,gBAAO,EAAP,C;K;IAaxD,wD;MAEQ,sB;QAAqB,yBAAO,UAA
U,OAAV,CAAP,C;WACrB,sD;QAA4B,yBAAO,OAAP,C;WAC5B,2B;QAAmB,yBAAO,kBAAP,C;;QACX,yBA
Ae,SAAR,OAAQ,CAAf,C;K;IIL7EhB,+B;MAY6B,kBAAIB,QAAQ,SAAR,EAAC,EAAd,C;MACH,IX0EE,WWIE
E,GAAK,CAAT,C;QAAY,MAAM,gCAAyB,oEAAzB,C;MADtB,OX4EO,W;K;IWvEX,wC;MAGBW,Q;MAAA,q
CAAiB,KAAjB,C;MAAA,iB;QAA2B,MAAM,gCAAyB,8BAAO,SAAP,4CAA+C,KAAxE,C;;MAAxC,OAAO,I;K
;IAGX,qC;MAY6B,kBAAIB,QAAQ,SAAR,EAAC,EAAd,C;MAAP,OXmEqB,WWnEa,IAAM,CXmEjC,GAAqB,
WAARB,GAA+B,I;K;IWhE1C,8C;MAGBI,WAAW,KAAX,C;MAC4B,kBAArB,QAAQ,SAAR,EAAC,KAAd,C;M
AAP,OX+CqB,WW/CgB,IAAM,CX+CpC,GAAqB,WAArB,GAA+B,I;K;IW5C1C,gC;MAWI,IAAY,CAAR,8BA
AW,CAAf,C;QACI,OAAO,YAAM,SAAN,C;;MAEX,MAAM,gCAAyB,SAAM,SAAN,4BAAzB,C;K;IAGV,yC;M
AkBW,Q;MANP,IAAI,EAAU,CAAV,sBAaA,EAAb,CAAJ,C;QACI,MAAM,gCAAyB,oBAAiB,KAAjB,4CAAzB,
C;;MAEV,IAAI,YAAO,CAAP,IAAY,aAAQ,KAAxB,C;QACI,MAAM,gCAAyB,WAAQ,SAAR,mDAAwD,KAAj
F,C;;MAEH,IAAI,YAAO,EAAX,C;QACH,mBAAM,SAAN,C;;QAEA,0BAAM,SAAN,IAAa,EAAb,C;;MAHJ,W;

K;IAuFJ,8B;MAWSc,+B;K;0EAEtC,4B;MAM8D,OAAK,oBAAL,SAAK,CAAL,GAaKb,K;K;IAEhF,gD;MAQoC,0B;QAAA,aAAsB,K;MAcTD,IAAI,cAAQ,KAAZ,C;QAAmB,OAAO,I;MAC1B,IAAI,CAAC,UAAL,C;QAAiB,OAAO,K;MAExB,gBAAqB,cAAL,SAAK,C;MACrB,iBAAuB,cAAN,KAAM,C;MAEhB,yBAAa,U;MAAb,U;QAA2B,OfFrMyB,oBEqMzB,SFrMyB,CAAY,cAfrB,YAAY,CAAZ,CEoNhB,KFrMyB,oBEqMI,UFrMJ,CAAY,cAfrB,YAAY,CAAZ,C;;MEoNID,W;K;IAGJ,gC;MAGyC,QAAQ,cAAA,sCAAk,cAAL,EAAoB,sCAAk,cAAzB,CAAR,6B;K;ImL3OzC,6C;MAc6B,4B;QAAA,eAAuB,G;MAChD,wCAAsB,EAAtB,EAA0B,YAA1B,C;K;IAEJ,mE;MAKwC,yB;QAAA,YAAoB,E;MAAI,4B;QAAA,eAAuB,G;MjMGnF,IAAI,CmBwR+C,CAAC,Q8K1R5C,Y9K0R4C,CnBxRpD,C;QACI,ciMHIC,wC;QjMIjC,MAAM,gCAAyB,OAAQ,WAAjC,C;;MiMHV,cAAY,gB;MAEC,yBAAS,mBAAS,YAAA,SAAU,OAAV,EAAmB,OAAM,KAAzB,CAAT,I;MAAT,wBAAiD,kBAaKb,SAaIB,C;MA0E9D,gBAAgB,iBA1ET,OA0ES,C;M3Lg7CT,kBAAoB,gB;MAoSd,gB;MADb,YAAY,C;MACC,O2L9xDN,O3L8xDM,W;kBAAb,OAAa,cAAb,C;QAAA,sB;QA1RsB,U;QAAA,cA0RT,oBAAmB,cAAnB,EAAmB,sBAAnB,U;Q2L/sDIB,kB;;YAHA,CAAC,YAAS,CAAT,IAAc,qBAAf,KAA4C,Q3LktDG,I2LltDH,C;UAC5C,a;;UAEA,4B;UA9E+B,uB;;Y/KgHzB,kC;YAAA,wBZ6qDyC,IY7qDzC,C;YAAA,qB;YAAA,oB;YAAA,oB;YAAAd,gE;cACI,I+KjHkD,CAAI,aAAH,U/KiHrC,YZ4qDqC,IY5qDrC,YAAK,OAAL,E+KjHqC,CAAG,C/KiHtD,C;gBACI,sBAAO,O;gBAAP,wB;;YAGR,sBAAO,E;;U+KrHH,iD;UAGI,gCAA2B,EAA3B,C;YAHJ,2BAGqC,I;IBACjC,IAAK,a3LyxD0C,I2LzxD1C,gBAAyB,uBAAzB,CAAL,C;YAJJ,2B3L6xDmD,IOjmDsB,WoLxLI,0BAAuC,mBAAvC,IpLwLJ,C;;YoL5LzE,2BAKY,I;UAYER,iE9LJD,yB8LIC,4B3L+sD+C,I;;QA1RpB,8B;UAA6C,6B;;M2LpgDhF,OAIkF,S3Lo7CE,W2Lp7CF,EAAO,mBAAc,kBAAd,CAAP,EAA0C,IAA1C,CACA,W;K;IAvET,+B;MAeyC,gCAAc,EAAd,C;K;IAEzC,6C;MAGgC,yB;QAAA,YAAoB,E;MAM3C,Q;MALL,cAAY,gB;M3LurBL,kBAAS,gB;MA2FA,U;MAAA,S2LhxBM,O3LgxBN,W;MAAhB,OAAGB,gBAAhB,C;QAAGB,2B;QAAM,Ia3hB6B,CAAC,Qb2hBhB,Oa3hBgB,Cb2hB9B,C;UAAwB,WAAy,WAAI,OAAJ,C;;M2L9wBrD,kB3L+wBE,W;MAMrBA,oBAAM,iBAAa,qCAAwB,EAAXB,CAAb,C;MAuEA,U;MAAA,+B;MAAb,OAAa,gBAAb,C;QAAA,wB;QACT,aAAY,uBAAc,IAAd,E;;M2L5gDhB,sBAAsB,CAGjB,oB3L0gDE,a2L1gDF,CAHiB,mBAGF,C;MAEP,yBAAS,mBAAS,YAAA,SAAU,OAAV,EAAmB,OAAM,KAAzB,CAAT,I;MAAT,wBAAiD,kBAaKb,SAaIB,C;MAMc9D,gBAAgB,iBAnCT,OAmCS,C;M3Lg7CT,oBAAoB,gB;MAoSd,kB;MADb,YAAY,C;MACC,S2LvDN,O3LuvDM,W;MAAb,OAAa,gBAAb,C;QAAA,a,0B;QA1RsB,U;QAAA,cA0RT,oBAAmB,cAAnB,EAAmB,sBAAnB,U;Q2L/sDIB,kB;Q3Lq7C2B,c2Lx7C3B,CAAC,YAAS,CAAT,IAAc,qBAAf,KAA4C,Q3LktDG,M2LltDH,C3Lw7CjB,G2Lv7C3B,I3Lu7C2B,G2Lr7C3B,oBAxcmG,Q3LuvDpD,M2LvDoD,kBAwCnG,Y9LJD,yB8LIC,4B3L+sD+C,MA1RpB,U;UAA6C,+B;;M2L79ChF,OAOCK,S3Lo7CE,a2Lp7CF,EAAO,mBAAc,kBAAd,CAAP,EAA0C,IAA1C,CACA,W;K;IAjCI,8C;MAAA,qB;QAEg,IAAG,QAAG,EAAG,CAAH,C;UAEQ,IAAA,EAAG,OAAH,GAAy,cAAO,OAAAnB,C;YAHZ,OAGyC,c;;YAHZC,OAIoB,E;;UAJpB,OAoY,iBAAS,E;O;K;IAfjC,0C;MAKgC,sB;QAAA,SAAiB,M;MAC7C,OAYK,eAXA,OADL,uBACK,EAAl,4BAAJ,CAWA,EAAa,IAAb,C;K;IAET,gC;MAAwC,uB;;Q/KmDtB,gC;QAAA,gC;QAAA,mB;QAAA,kB;QAAA,kB;QAAd,0D;UACI,I+KpD+C,CAAI,aAAH,U/KoDIC,iCAAk,KAAL,E+KpDkC,CAAG,C/KoDnD,C;YACI,sBAAO,K;YAAP,wB;;QAGR,sBAAO,E;;Mf3CA,4B;M8Lb6B,OAA8C,OAAM,EAAV,GAAc,gBAAd,GAA0B,E;K;IAGpF,wC;MAAkB,W;K;IAC9B,oD;MAAA,uB;QAaKb,wBAAS,I;O;K;IAFvC,mC;MACI,IAAA,M9KkMgD,YAAU,C8KIM1D,C;QAD4C,OACxB,wB;;QADwB,OAEPc,kC;K;mBAGZ,yB;M3L86CA,+D;MAoSA,wE;M2LltDA,sF;QAKI,gBAAGB,2B;Q3Lg7CT,kBAAoB,gB;QAoSd,gB;QADb,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UA1RsB,U;UAAA,cA0RT,oBAAmB,cAAnB,EAAmB,sBAAnB,U;U2L/sDIB,kB;U3Lq7C2B,c2Lx7C3B,CAAC,YAAS,CAAT,IAAc,qBAAf,KAA4C,Q3LktDG,I2LltDH,C3Lw7CjB,G2Lv7C3B,I3Lu7C2B,G2Lr7C3B,sC3L+sD+C,I2L/sD/C,a9LJD,yB8LIC,4B3L+sD+C,IA1RpB,U;YAA6C,6B;;Q2Lz7ChF,OAMK,S3Lo7CE,W2Lp7CF,EAAO,mBAAc,kBAAd,CAAP,EAA0C,IAA1C,CACA,W;O;KAbT,C;6EvEkSA,0B;MAGmE,OAAA,SAAK,gBAAO,GAAP,C;K;qFAExE,yB;MAAA,yD;MAAA,gC;QAO2B,gBAAhB,oB;QAAsB,avHrU7B,W;QuHqUA,OvHpUO,SuHoUqC,W;O;KAPhD,C;uFAUA,yB;MAAA,iE;MAAA,0C;QAQmC,gBAAXB,mBAAc,QAAd,C;QAA8B,avHhVrC,W;QuHgVA,OvH/uo,SuH+U6C,W;O;KARxD,C;IAWA,oC;MAIiB,Q;MAAb,wBAAa,KAAb,gB;QAAA,WAAA,KAAb,M;QACI,yBAAO,IAAP,C;;MACJ,OAAO,S;K;IAGX,oC;MAIiB,Q;MAAb,wBAAa,KAAb,gB;QAAA,WAAA,KAAb,M;QACI,yBAAO,IAAP,C;;MACJ,OAAO,S;K;qFAGX,qB;MAG8D,gCAAO,EAAP,C;K;qFAE9D,4B;MAGkF,OAAA,yBAAO,KAAP,CALpB,gBAAO,EAAP,C;K;qFAO9D,4B;MAG4E,OAAA,yBAAO,KAAP,CAVd,gBAAO,EAAP,C;K;qFAY9D,4B;MAGyE,OAAA,yBAAO,KAAP,CAfX,gBAAO,EAAP,C;

K;qFAiB9D,4B;MAG8E,OAAA,yBAAO,KAAP,CApBhB,gBAAO,EAAP,C;K;qFAsB9D,4B;MAGyE,OAAA,yBAAO,KAAP,CAzBX,gBAAO,EAAP,C;K;qFA2B9D,4B;MAG4E,OAAA,yBAAO,KAAP,CA9Bd,gBAAO,EAAP,C;K;I/H/a9D,iC;MAK0C,iCAAqB,EAARb,C;K;IAE1C,0C;MAQmB,Q;MAAA,qBAAL,SAAK,EAAY,KAAZ,C;MAAL,iB;QAA2B,OAAO,I;MAA5C,UAAU,I;MACV,IAAI,MAAM,sCAAK,UAAAX,IAAwB,MAAM,sCAAK,UAAvC,C;QAAkD,OAAO,I;MACzD,OAAW,OAAJ,GAAL,C;K;IAGf,kC;MAK4C,kCAAsB,EAATB,C;K;IAE5C,2C;MAQmB,Q;MAAA,qBAAL,SAAK,EAAY,KAAZ,C;MAAL,iB;QAA2B,OAAO,I;MAA5C,UAAU,I;MACV,IAAI,MAAM,uCAAM,UAAZ,IAAYB,MAAM,uCAAM,UAAzC,C;QAAoD,OAAO,I;MAC3D,OAAW,QAAJ,GAAL,C;K;IAGf,gC;MAKwC,gCAAoB,EAAPB,C;K;IAExC,yC;MAQI,WAAW,KAAX,C;MAEA,aAAa,SAAK,O;MACIB,IAAI,WAAU,CAAd,C;QAAiB,OAAO,I;MAExB,S;MACA,c;MACA,S;MAEA,gBAAGB,qBAAK,CAAL,C;MACHB,IAAI,YAAY,EAAhB,C;QACI,IAAI,WAAU,CAAd,C;UAAiB,OAAO,I;QAEExB,QAAQ,C;QAER,IAAI,cAAa,EAajB,C;UACI,aAAa,I;UACb,QAAQ,W;eACL,IAAI,cAAa,EAajB,C;UACH,aAAa,K;UACb,QAAQ,W;;UAER,OAAO,I;QAEX,QAAQ,C;QACR,aAAa,K;QACb,QAAQ,W;;MAIZ,uBAAuB,S;MAEvB,qBAAqB,gB;MACrB,aAAa,C;MACb,aAAU,KAAV,MAAsB,MAAtB,M;QACI,YAAY,QAAQ,qBAAK,CAAL,CAAR,EAaiB,KAAjB,C;QAEZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,IAAI,SAAS,cAAb,C;UACI,IAAI,mBAaKB,gBAAtB,C;YACI,iBAAiB,QAAQ,KAAR,I;YAEjB,IAAI,SAAS,cAAb,C;cACI,OAAO,I;;YAGX,OAAO,I;;QAIIf,6BAAU,KAAV,C;QAEA,IAAI,UAAAS,QAAQ,KAAR,IAAT,CAAJ,C;UAA4B,OAAO,I;QAEnC,kBAAU,KAAV,I;;MAGJ,OAAW,UAAJ,GAAgB,MAAhB,GAA4B,CAAC,MAAD,I;K;IAGvC,iC;MAK0C,iCAAqB,EAARb,C;K;IAE1C,0C;MAQI,WAAW,KAAX,C;MAEA,aAAa,SAAK,O;MACIB,IAAI,WAAU,CAAd,C;QAAiB,OAAO,I;MAExB,S;MACA,c;MACA,S;MAEA,gBAAGB,qBAAK,CAAL,C;MACHB,IAAI,YAAY,EAAhB,C;QACI,IAAI,WAAU,CAAd,C;UAAiB,OAAO,I;QAEExB,QAAQ,C;QAER,IAAI,cAAa,EAajB,C;UACI,aAAa,I;UACb,gC;eACG,IAAI,cAAa,EAajB,C;UACH,aAAa,K;UACb,6B;;UAEA,OAAO,I;QAEX,QAAQ,C;QACR,aAAa,K;QACb,6B;;MAIJ,2C;MAEA,qBAAqB,gB;MACrB,e;MACA,aAAU,KAAV,MAAsB,MAAtB,M;QACI,YAAY,QAAQ,qBAAK,CAAL,CAAR,EAaiB,KAAjB,C;QAEZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,IAAI,uBAAS,cAAT,KAAJ,C;UACI,IAAI,uBAaKB,gBAaIB,CAAJ,C;YACI,iBAAiB,8BAAQ,KAAR,E;YAEjB,IAAI,uBAAS,cAAT,KAAJ,C;cACI,OAAO,I;;YAGX,OAAO,I;;QAIIf,6CAAU,KAAV,E;QAEA,IAAI,uBAAS,8BAAQ,KAAR,EAAT,KAAJ,C;UAA4B,OAAO,I;QAEnC,6CAAU,KAAV,E;;MAGJ,OAAW,UAAJ,GAAgB,MAAhB,GAA6B,MAAD,a;K;IAIvC,kC;MAAYD,MAAM,0BAAAsB,6BAA0B,KAA1B,MAAtB,C;K;uEwBhI/D,yB;MAAA,oC;MAAA,uC;QAIiBAAiB,C;QACjB,eAAe,mBAAS,CAAT,I;QACf,iBAAiB,K;QAEjB,OAAO,cAAc,QAArB,C;UACI,YAAGB,CAAC,UAAAL,GAAiB,UAAjB,GAAiC,Q;UAC7C,YAAY,UAAU,iCAAk,KAAL,EAAY,C;UAEZ,IAAI,CAAC,UAAAL,C;YACI,IAAI,CAAC,KAAAL,C;cACI,aAAa,I;;cAEb,0BAAc,CAAd,I;;YAEJ,IAAI,CAAC,KAAL,C;cACI,K;;cAEA,sBAAy,CAAZ,I;;QAIz,OAAO,8BAAy,UAAZ,EAAwB,WAAW,CAAX,IAAxB,C;O;KAZBX,C;yEA4BA,yB;MAAA,8B;MA5BA,oC;MA4BA,uC;QAIK,Q;QAAsB,kBAAtB,2D;QA5BD,iBAAiB,C;QACjB,eAAe,qBAAS,CAAT,I;QACf,iBAAiB,K;QAEjB,OAAO,cAAc,QAArB,C;UACI,YAAGB,CAAC,UAAAL,GAAiB,UAAjB,GAAiC,Q;UAC7C,YAsBwB,SAtBZ,CAAU,mCAAK,KAAL,EAAY,C;UAEZ,IAAI,CAAC,UAAAL,C;YACI,IAAI,CAAC,KAAL,C;cACI,aAAa,I;;cAEb,0BAAc,CAAd,I;;YAEJ,IAAI,CAAC,KAAL,C;cACI,K;;cAEA,sBAAy,CAAZ,I;;QAWZ,OAPo,gCAAY,UAAZ,EAAwB,WAAW,CAAX,IAAxB,CAOGC,W;O;KAJ3C,C;IFAMA,yB;MAAA,mD;MAAA,oC;MAAA,uC;QAIuB,UAAAL,MAAK,EAAL,MAAK,EAAL,M;QAAK,mBAAL,SAAK,C;QAAL,mB;QAAA,kB;QAAA,kB;QAAd,0D;UACI,IAAI,CAAC,UAAU,iCAAk,KAAL,EAAY,CAAL,C;YACI,OAAO,8BAAy,KAAZ,EAAMB,gBAAnB,C;QAEf,OAAO,E;O;KARX,C;mFAWA,yB;MAAA,8B;MAXA,mD;MAAA,oC;MAWA,uC;QAIK,Q;QAAsB,kBAAtB,2D;QAAsB,oB;;UAXJ,kC;UAAA,qBAAL,WAAK,C;UAAAL,qB;UAAA,oB;UAAA,oB;UAAAd,0D;YACI,IAAI,CAUyB,SAVxB,CAAU,mCAAK,KAAL,EAAY,CAAL,C;cACI,mBAAO,gCAAY,KAAZ,EAAMB,kBAAnB,C;cAAP,qB;;UAER,mBAAO,E;;QAOP,OAA4C,2B;O;KAJhD,C;6EAMA,yB;MAAA,mD;MAAA,+C;MAAA,oC;MAAA,uC;QAIkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAAc,cAAAd,C;UAAc,uB;UACV,IAAI,CAAC,UAAU,iCAAk,KAAL,EAAY,CAAL,C;YACI,OAAO,8BAAy,CAAZ,EAAY,QAAQ,CAAR,IAAf,C;;QAEf,OAAO,E;O;KARX,C;+EAWA,yB;MAAA,8B;MAXA,mD;MAAA,+C;MAAA,oC;MAWA,uC;QAIK,Q;QAAsB,kBAAtB,2D;QAAsB,kB;;UAXT,U;UAAA,SAAa,SAAR,YAAL,WAAK,CAAQ,CAAb,W;UAAAd,OAAc,gBAAd,C;YAAc,yB;YACV,IAAI,CAUuB,SAVtB,CAAU,mCAAK,KAAL,EAAY,CAAL,C;cACI,iBAAO,gCAAY,CAAZ,EAAY,QAAQ,CAAR,IAAf,C;cAAP,mB;;UAER,iBAAO,E;;QAOP,OAA0C,yB;O;KAJ9C,C;IAMA,kC;MAhEI,iBAAiB

,C;MACjB,eAAe,mBAAS,CAAT,I;MACf,iBAAiB,K;MAEjB,OAAO,cAAc,QAARb,C;QACI,YAAgB,CAAC,UAAL,GAAiB,UAAjB,GAAiC,Q;QAC7C,YA6DgE,4BA7D1C,iCAAK,KAAL,EA6D0C,E;QA3DhE,IAAI,CAAC,UAAAL,C;UACI,IAAI,CAAC,KAAL,C;YACI,aAAa,I;;YAEb,0BAAc,CAAd,I;;UAEJ,IAAI,CAAC,KAAL,C;YACI,K;;YAEA,sBAAY,CAAZ,I;;MAkDiD,OA9CtD,8BAAY,UAAZ,EAAwB,WAAW,CAAX,IAAxB,C;K;IAGDX,kC;MAzCK,Q;MAAsB,kBAAtB,2D;MA5BD,iBAAiB,C;MACjB,eAAe,qBAAS,CAAT,I;MACf,iBAAiB,K;MAEjB,OAAO,cAAc,QAARb,C;QACI,YAAgB,CAAC,UAAAL,GAAiB,UAAjB,GAAiC,Q;QAC7C,YAkEoD,4BAIE9B,mCAAK,KAAL,EAkE8B,E;QAhEpD,IAAI,CAAC,UAAAL,C;UACI,IAAI,CAAC,KAAL,C;YACI,aAAa,I;;YAEb,0BAAc,CAAd,I;;UAEJ,IAAI,CAAC,KAAL,C;YACI,K;;YAEA,sBAAY,CAAZ,I;;MAuDqC,OAnD1C,gCAAY,UAAZ,EAAwB,WAAW,CAAX,IAAxB,CAOgC,W;K;IA8C3C,uC;MAGsE,oB;;QA3C/C,gC;QAAA,gC;QAAL,mB;QAAA,kB;QAAA,kB;QAAd,0D;UACI,IAAI,CA0CsE,4BA1C3D,iCAAK,KAAL,EA0C2D,EA1C1E,C;YACI,mBAAO,8BAAY,KAaz,EAAmB,gBAAnB,C;YAAP,qB;;QAER,mBAAO,E;;MAuC2D,uB;K;IAEtE,uC;MAICK,Q;MAAsB,kBAAtB,2D;MAAsB,oB;;QAXJ,kC;QAAA,wBAAL,WAAK,C;QAAL,qB;QAAA,oB;QAAA,oB;QAAd,0D;UACI,IAAI,CA+C0D,4BA/C/C,mCAAK,KAAL,EA+C+C,EA/C9D,C;YACI,mBAAO,gCAAY,KAaz,EAAmB,kBAAnB,C;YAAP,qB;;QAER,mBAAO,E;;MA4C+C,OArCV,2B;K;IAuChD,qC;MAGoE,kB;;QApCID,Q;QAAA,OAaA,WAAW,yBAAQ,CAAb,W;QAAd,OAac,cAAd,C;UAAc,uB;UACV,IAAI,CAmCkE,4BAnCvD,iCAAK,KAAL,EAACuD,EAnCtE,C;YACI,iBAAO,8BAAY,CAAZ,EAAe,QAAQ,CAAR,IAAf,C;YAAP,mB;;QAER,iBAAO,E;;MAGCyD,qB;K;IAEpE,qC;MA3BK,Q;MAAsB,kBAAtB,2D;MAAsB,kB;;QAXT,U;QAAA,SAAa,WAAW,eAAL,WAAK,CAAQ,CAAb,W;QAAd,OAac,gBAAd,C;UAAc,yB;UACV,IAAI,CAwCsD,4BAxC3C,mCAAK,KAAL,EAwC2C,EAxC1D,C;YACI,iBAAO,gCAAY,CAAZ,EAAe,QAAQ,CAAR,IAAf,C;YAAP,mB;;QAER,iBAAO,E;;MAqC6C,OA9BV,yB;K;IAGC9C,2B;MA9FI,iBAAiB,C;MACjB,eAAe,mBAAS,CAAT,I;MACf,iBAAiB,K;MAEjB,OAAO,cAAc,QAARb,C;QACI,YAAgB,CAAC,UAAAL,GAAiB,UAAjB,GAAiC,Q;QAC7C,mCAAsB,iCAAK,KAAL,EAAtB,E;QAEA,IAAI,CAAC,UAAAL,C;UACI,IAAI,CAAC,KAAL,C;YACI,aAAa,I;;YAEb,0BAAc,CAAd,I;;UAEJ,IAAI,CAAC,KAAL,C;YACI,K;;YAEA,sBAAY,CAAZ,I;;MAGf+B,OA5EpC,8BAAY,UAAZ,EAAwB,WAAW,CAAX,IAAxB,C;K;yEA8EX,yB;MAAA,8B;MAAA,qC;MAAA,4B;QAI2C,Q;QAAD,OAAuB,KAAtB,2DAAsB,CAAO,W;O;KAJxE,C;IAMA,gC;MAGoD,oB;;QAI7E7B,gC;QAAA,gC;QAAL,mB;QAAA,kB;QAAA,kB;QAAd,0D;UACI,IAAI,wBAAW,iCAAK,KAAL,EAAX,EAAJ,C;YACI,mBAAO,8BAAY,KAaz,EAAmB,gBAAnB,C;YAAP,qB;;QAER,mBAAO,E;;MASEyC,uB;K;mFAEpD,yB;MAAA,8B;MAAA,+C;MAAA,4B;QAIgD,Q;QAAD,OAAuB,UAAAtB,2DAAsB,CAAY,W;O;KAJIF,C;IAMA,8B;MAGkD,kB;;QApEhC,Q;QAAA,OAAa,WAAW,yBAAQ,CAAb,W;QAAd,OAac,cAAd,C;UAAc,uB;UACV,IAAI,wBAAW,iCAAK,KAAL,EAAX,EAAJ,C;YACI,iBAAO,8BAAY,CAAZ,EAAe,QAAQ,CAAR,IAAf,C;YAAP,mB;;QAER,iBAAO,E;;MAGEuC,qB;K;+EAEID,yB;MAAA,8B;MAAA,2C;MAAA,4B;QAI8C,Q;QAAD,OAAuB,QAAtB,2DAAsB,CAAU,W;O;KAJ9E,C;IAMA,8C;MAU8C,uB;QAAA,UAAgB,E;MAO5C,Q;MANd,IAAI,SAAS,CAAb,C;QACI,MAAM,gCAAYB,oBAAiB,MAAjB,wBAAzB,C;MACV,IAAI,UAAU,SAAK,OAAAnB,C;QACI,OAAy,mBAAL,SAAK,EAAY,CAAZ,EAAe,SAAK,OAApB,C;MAEHb,SAAS,mBAAc,MAAd,C;MACK,gBAAS,SAAK,OAAd,I;MAAd,aAAU,CAAV,iB;QACI,EAAG,gBAAO,OAAP,C;MACP,EAAG,gBAAO,SAAP,C;MACH,OAAO,E;K;IAGX,gD;MASwC,uB;QAAA,UAAgB,E;MACnD,Q;MAAD,OAAuB,SAAtB,6DAAsB,EAAS,MAAT,EAaiB,OAAjB,CAA0B,W;K;IAErD,4C;MAU4C,uB;QAAA,UAAgB,E;MAQ1C,Q;MAPd,IAAI,SAAS,CAAb,C;QACI,MAAM,gCAAYB,oBAAiB,MAAjB,wBAAzB,C;MACV,IAAI,UAAU,SAAK,OAAAnB,C;QACI,OAAy,mBAAL,SAAK,EAAY,CAAZ,EAAe,SAAK,OAApB,C;MAEHb,SAAS,mBAAc,MAAd,C;MACT,EAAG,gBAAO,SAAP,C;MACW,gBAAS,SAAK,OAAd,I;MAAd,aAAU,CAAV,iB;QACI,EAAG,gBAAO,OAAP,C;MACP,OAAO,E;K;IAGX,8C;MASsC,uB;QAAA,UAAgB,E;MACjD,Q;MAAD,OAAuB,OAAtB,6DAAsB,EAAO,MAAP,EAAe,OAAf,CAAwB,W;K;2FAEnD,qB;MAWI,OAAO,qBAAgB,SAAK,OAAL,KAAe,C;K;+EAG1C,qB;MAMoD,4BAAU,C;K;sFAE9D,qB;MAMuD,0BAAS,C;K;mFAMhE,yB;MAAA,2C;MAAA,4B;QAMuD,QAAC,kB;O;KANxD,C;yFAQA,yB;MAAA,2C;MAAA,4B;QAWI,OAAO,qBAAqB,QAAL,SAAK,C;O;KAXhC,C;IAiB4D,+C;MAAA,kC;MAAS,uB;MACjE,eAAoB,C;K;gDAEpB,Y;MAA2C,gB;MAAA,iE;MAAJ,4C;K;+CAEvC,Y;MAAyC,sBAAQ,yB;K;;IARrD,+B;MAG4D,4C;K;+EAQ5D,qB;MAE8C,uCAAQ,E;K;+EAEtD,mC;MASI,OA5DgD,qBAAU,CA4D1D,GAAe,cAAf,GAAmC,S;K;6EAEvC,yB;MAAA,2C;MAAA,0C;QASI,OAAI,kBAAJ,GAAe,cAAf,GAAmC,S;O;KATvC,C;IAeI,mC;MAAQ,uBAAg,mBAAS,CAAT,IAAH,C;K;IAMR,qC;MAAQ,OAAA,SAAK,OAAL,GAAc,CAAd,I;K;IAEZ,8C;MAIuB,Q;MAAA,0BAAS,CAAT,

I;MAAnB,OAAgB,CAAT,8BACgB,gBAAZ,qBAAK,KAAL,CAAY,CADhB,IAEoB,eAAhB,qBAAK,QAAQ,CAAR,IAAL,CAAgB,C;K;IAG/B,uC;MAGuD,ONpKyC,oBMoK/B,KAAM,MNpKyB,EMoKIB,KAAM,aAN,GAAqB,CAArB,INpKkB,C;K;IMsKhG,yC;MAGqE,qCAAY,KAAM,MAAIB,EAAYB,KAAM,aAN,GAAqB,CAArB,IAAzB,C;K;uFAErE,iC;MAS2E,2BAAy,KAAZ,EAAMb,GAAnB,C;K;mFAE3E,2C;MAO0D,wB;QAAA,WAAgB,gB;MAAkB,OAAA,8BAAy,UAAZ,EAawB,QAAxB,CAAKC,W;K;IAE9H,uC;MAG6D,OAAA,8BAAy,KAAM,MAAIB,EAAYB,KAAM,aAN,GAAqB,CAArB,IAAzB,CAAiD,W;K;IAE9G,sE;MAImD,qC;QAAA,wBAAgC,S;MAC/E,YAAy,sBAAQ,SAAR,C;MACZ,OAAW,UAAZ,EAAPb,GAAwB,qBAAXB,GN1M4F,oBM0M/B,CN1M+B,EM0M5B,KN1M4B,C;K;IM6MhG,wE;MAIqD,qC;QAAA,wBAAgC,S;MACjF,YAAy,sBAAQ,SAAR,C;MACZ,OAAW,UAAZ,EAAPb,GAAwB,qBAAXB,GNnN4F,oBMmN/B,CNnN+B,EMmN5B,KNnN4B,C;K;IMsNhG,qE;MAIkD,qC;QAAA,wBAAgC,S;MAC9E,YAAy,sBAAQ,SAAR,C;MACZ,OAAW,UAAZ,EAAPb,GAAwB,qBAAXB,GN5N4F,oBM4N/B,QAAQ,CAAR,IN5N+B,EM4NpB,gBN5NoB,C;K;IM+NhG,uE;MAIoD,qC;QAAA,wBAAgC,S;MACHf,YAAy,sBAAQ,SAAR,C;MACZ,OAAW,UAAZ,EAAPb,GAAwB,qBAAXB,GNrO4F,oBMqO/B,QAAQ,SAAU,OAAIB,INrO+B,EMqOL,gBNrOK,C;K;IMwOhG,0E;MAIuD,qC;QAAA,wBAAgC,S;MACnF,YAAy,0BAAy,SAAZ,C;MACZ,OAAW,UAAZ,EAAPb,GAAwB,qBAAXB,GN9O4F,oBM8O/B,CN9O+B,EM8O5B,KN9O4B,C;K;IMiPhG,4E;MAIyD,qC;QAAA,wBAAgC,S;MACrF,YAAy,0BAAy,SAAZ,C;MACZ,OAAW,UAAZ,EAAPb,GAAwB,qBAAXB,GNvP4F,oBMuP/B,CNvP+B,EMuP5B,KNvP4B,C;K;IM0PhG,yE;MAIsD,qC;QAAA,wBAAgC,S;MACIF,YAAy,0BAAy,SAAZ,C;MACZ,OAAW,UAAZ,EAAPb,GAAwB,qBAAXB,GNhQ4F,oBMgQ/B,QAAQ,CAAR,INhQ+B,EMgQpB,gBNhQoB,C;K;IMmQhG,2E;MAIwD,qC;QAAA,wBAAgC,S;MACpF,YAAy,0BAAy,SAAZ,C;MACZ,OAAW,UAAZ,EAAPb,GAAwB,qBAAXB,GNzQ4F,oBMyQ/B,QAAQ,SAAU,OAAIB,INzQ+B,EMyQL,gBNzQK,C;K;IM4QhG,oE;MAOI,IAAI,WAAW,UAAf,C;QACI,MAAM,8BAA0B,gBAAa,QAAb,oCAAKD,UAAID,OAA1B,C;MACV,SAAS,sB;MACT,EAAG,qBAAy,SAAZ,EAakB,CAAIB,EAAqB,UAArB,C;MACH,EAAG,gBAAO,WAAp,C;MACH,EAAG,qBAAy,SAAZ,EAakB,QAAIB,EAA4B,gBAA5B,C;MACH,OAAO,E;K;yFAGX,yB;MAAA,8B;MAAA,qD;MAAA,+D;QAOK,Q;QAAD,OAAuB,aAAtB,2DAAsB,EAAa,UAAb,EAAYB,QAAzB,EAAMc,WAAnc,CAAGD,W;O;KAP3E,C;IASA,uD;MAOI,+BAAa,KAAM,MAAnB,EAA0B,KAAM,aAN,GAAqB,CAArB,IAA1B,EAakD,WAAID,C;K;yFAEJ,yB;MAAA,8B;MAAA,qD;MAAA,gD;QAOK,Q;QAAD,OAAuB,aAAtB,2DAAsB,EAAa,KAAb,EAAoB,WAApB,CAAiC,W;O;KAP5D,C;IASA,sD;MASI,IAAI,WAAW,UAAf,C;QACI,MAAM,8BAA0B,gBAAa,QAAb,oCAAKD,UAAID,OAA1B,C;MAEV,IAAI,aAAY,UAAhB,C;QACI,OAAy,mBAAL,SAAK,EAAY,CAAZ,EAAe,gBAAf,C;MAEhB,SAAS,mBAAc,oBAAU,QAAV,GAAqB,UAArB,KAAd,C;MACT,EAAG,qBAAy,SAAZ,EAakB,CAAIB,EAAqB,UAArB,C;MACH,EAAG,qBAAy,SAAZ,EAakB,QAAIB,EAA4B,gBAA5B,C;MACH,OAAO,E;K;uFAGX,yB;MAAA,8B;MAAA,mD;MAAA,kD;QASK,Q;QAAD,OAAuB,YAAtB,2DAAsB,EAAY,UAAZ,EAawB,QAAxB,CAAKC,W;O;KAT7D,C;IAWA,yC;MAKqE,8BAAy,KAAM,MAAIB,EAAYB,KAAM,aAN,GAAqB,CAArB,IAAzB,C;K;uFAErE,yB;MAAA,8B;MAAA,mD;MAAA,mC;QAOK,Q;QAAD,OAAuB,YAAtB,2DAAsB,EAAY,KAAZ,CAAMb,W;O;KAP9C,C;IASA,yC;MAKI,IAAI,wBAAW,MAAX,CAAJ,C;QACI,OAAO,8BAAy,MAAO,OAAAnB,EAA2B,gBAA3B,C;;MAEX,OAAO,8BAAy,CAAZ,EAAe,gBAAf,C;K;IAGX,2C;MAKI,IAAI,wBAAW,MAAX,CAAJ,C;QACI,ON3XyE,oBM2XxD,MAAO,ON3XiD,C;;MM6X7E,OAAO,S;K;IAGX,yC;MAKI,IAAI,sBAAS,MAAT,CAAJ,C;QACI,OAAO,8BAAy,CAAZ,EAAe,mBAAS,MAAO,OAAhB,IAAf,C;;MAEX,OAAO,8BAAy,CAAZ,EAAe,gBAAf,C;K;IAGX,2C;MAKI,IAAI,sBAAS,MAAT,CAAJ,C;QACI,ON9YwF,oBM8YvE,CN9YuE,EM8YpE,mBAAS,MAAO,OAAhB,IN9YoE,C;;MMgZ5F,OAAO,S;K;IAGX,sD;MAMI,IAAK,qBAAU,MAAO,OAAP,GAAgB,MAAO,OAAvB,IAAV,CAAD,IAA6C,wBAAW,MAAX,CAA7C,IAAMe,sBAAS,MAAT,CAAvE,C;QACI,ONTawF,oBMsavE,MAAO,ONTagE,EMsaxD,mBAAS,MAAO,OAAhB,INtawD,C;;MMwa5F,OAAO,S;K;IAGX,mD;MAKmf,oCAAKB,SAAIB,EAA6B,SAA7B,C;K;IAEnF,mD;MAKuE,sCAAKB,SAAIB,EAA6B,SAA7B,C;K;IAEvE,iF;MAIsE,qC;QAAA,wBAAgC,S;MACIG,YAAy,sBAAQ,SAAR,C;MACL,Q;MAAA,IAAI,UAAZ,EAAb,C;QAAA,OAAiB,qB;;QA5JvB,U;QA4JM,OA5JgB,aAAtB,+DAAsB,EA4JyC,CA5JzC,EA4J4C,KA5J5C,EA4JmD,WA5JnD,CAAGD,W;;MA4JvE,W;K;IAGJ,mF;MAIwE,qC;QAAA,wBAAgC,S;MACpG,YAAy,sBAAQ,SAAR,C;MACL,Q;MAAA,IAAI,UAAZ,EAAb,C;QAAA,OAAiB,qB;;QArKvB,U;QAq

KM,OArKgB,aAAtB,+DAAsB,EAqKyC,CArKzC,EAqK4C,KArK5C,EAqKmD,WArKnD,CAAgD,W;;MAqKvE,W;K;IAGJ,gF;MAIqE,qC;QAAA,wBAAgC,S;MACjG,YAAy,sBAAQ,SAAR,C;MACL,Q;MAAA,IAAI,UAAS,EAAb,C;QAAA,OAAiB,qB;;QAA2B,iBAAa,QAAQ,CAAR,I;QAAb,eAAwB,gB;QA9K1E,U;QA8KM,OA9KgB,aAAtB,+DAAsB,EAAa,UAAb,EAAYB,QAazB,EA8K4D,WA9K5D,CAAgD,W;;MA8KvE,W;K;IAGJ,kF;MAIuE,qC;QAAA,wBAAgC,S;MACnG,YAAy,sBAAQ,SAAR,C;MACL,Q;MAAA,IAAI,UAAS,EAAb,C;QAAA,OAAiB,qB;;QAA2B,iBAAa,QAAQ,SAAU,OAAIB,I;QAAb,eAAuC,gB;QAvLzF,U;QAUlM,OAuLgB,aAAtB,+DAAsB,EAAa,UAAb,EAAYB,QAazB,EAuL2E,WAvL3E,CAAgD,W;;MAuLvE,W;K;IAGJ,oF;MAI2E,qC;QAAA,wBAAgC,S;MACvG,YAAy,0BAAy,SAAZ,C;MACL,Q;MAAA,IAAI,UAAS,EAAb,C;QAAA,OAAiB,qB;;QAA2B,iBAAa,QAAQ,SAAU,OAAIB,I;QAAb,eAAuC,gB;QAhMzF,U;QAgMM,OAHMgB,aAAtB,+DAAsB,EAAa,UAAb,EAAYB,QAazB,EAgM2E,WAhM3E,CAAgD,W;;MAGMvE,W;K;IAGJ,sF;MAIyE,qC;QAAA,wBAAgC,S;MACrG,YAAy,0BAAy,SAAZ,C;MACL,Q;MAAA,IAAI,UAAS,EAAb,C;QAAA,OAAiB,qB;;QAA2B,iBAAa,QAAQ,CAAR,I;QAAb,eAAwB,gB;QazM1E,U;QAYMM,OAzMgB,aAAtB,+DAAsB,EAAa,UAAb,EAAYB,QAazB,EAyM4D,WAzM5D,CAAgD,W;;MAyMvE,W;K;IAGJ,qF;MAI0E,qC;QAAA,wBAAgC,S;MACtG,YAAy,0BAAy,SAAZ,C;MACL,Q;MAAA,IAAI,UAAS,EAAb,C;QAAA,OAAiB,qB;;QAlNvB,U;QAKNM,OAInGB,aAAtB,+DAAsB,EAKNyC,CAINzC,EAKN4C,KAIN5C,EAKNmD,WAlNnD,CAAgD,W;;MAkNvE,W;K;IAGJ,uF;MAI4E,qC;QAAA,wBAAgC,S;MACxG,YAAy,0BAAy,SAAZ,C;MACL,Q;MAAA,IAAI,UAAS,EAAb,C;QAAA,OAAiB,qB;;QA3NvB,U;QA2NM,OA3NgB,aAAtB,+DAAsB,EA2NyC,CA3NzC,EA2N4C,KA3N5C,EA2NmD,WA3NnD,CAAgD,W;;MA2NvE,W;K;+EAOJ,yC;MAQoF,OAAA,KAAM,iBAAQ,SAAR,EAAC,WAAd,C;K;+EAE1F,uC;MAOI,OAAA,KAAM,iBAAQ,SAAR,EAAC,SAAd,C;K;yFAEV,yC;MAMyF,OAAA,KAAM,sBAAa,SAAb,EAAMb,WAAAnB,C;K;+FAE/F,yB;MAAA,oC;MAAA,gC;MAAA,uC;QAEW,Q;QAAA,IApe4C,mBAAS,CAoerD,C;uBAAkB,oBAAU,iCAAK,CAAL,EAAV,E;UAAA,YNljBoD,oBMkjBrB,CNljBqB,C;UMkjBtE,OLrjBwD,2BAAL,GAakB,K;;UKqjBrE,OAAyD,S;QAAhE,W;O;KafJ,C;iGakBA,yB;MAAA,oC;MAAA,uC;QAEI,OAtfmD,mBAAS,CAsf5D,GAAyB,UAAU,iCAAK,CAAL,EAAV,CAAMb,WAAAnB,GNpkBoD,oBMokBV,CNpkBU,CMokB7E,GAA2E,S;O;Kaf/E,C;+EAmBA,4B;MAIsE,OAAA,KAAM,iBAAQ,SAAR,C;K;IAE5E,0F;MAKI,IAAK,cAAc,CAAf,IAASb,aAAa,CAAnC,IAA0C,cAAa,SAAK,OAAL,GAAC,MAAd,IAAb,CAA1C,IAAiF,eAAc,KAAM,OAAN,GAAe,MAAf,IAAd,CAArF,C;QACI,OAAO,K;;MAGX,iBAAC,CAAd,UAAASB,MAAtB,U;QACI,IAAI,CAA0B,SAAZB,qBAAK,aAAa,KAAb,IAAL,CAAYB,EAAO,iBAAM,cAAc,KAAd,IAAN,CAAP,EAAMC,UAAAnC,CAA9B,C;UACI,OAAO,K;;MAEf,OAAO,I;K;IAGX,mD;MAG+C,0B;QAAA,aAASB,K;MACjE,OAAA,SAAK,OAAL,GAAC,CAAd,IAA2B,SAAR,qBAAK,CAAL,CAAQ,EAAO,IAAP,EAAa,UAAb,C;K;IAE/B,iD;MAG6C,0B;QAAA,aAASB,K;MAC/D,OAAA,SAAK,OAAL,GAAC,CAAd,IAAMC,SAAhB,qBAAK,2BAAL,CAAgB,EAAO,IAAP,EAAa,UAAb,C;K;IAEvC,qD;MAGyD,0B;QAAA,aAASB,K;MAC3E,IAAI,CAAC,UAAD,IAAe,6BAAf,IAAiC,0BAArC,C;QACI,OAAy,WAAAL,SAAK,EAAW,MAAX,C;;QAEZ,OAAO,6BAakB,CAAIB,EAAqB,MAArB,EAA6B,CAA7B,EAAgC,MAAO,OAAvC,EAA+C,UAA/C,C;K;IAGf,iE;MAG0E,0B;QAAA,aAASB,K;MAC5F,IAAI,CAAC,UAAD,IAAe,6BAAf,IAAiC,0BAArC,C;QACI,OAAy,aAAL,SAAK,EAAW,MAAX,EAAMb,UAAAnB,C;;QAEZ,OAAO,6BAakB,UAAIB,EAA8B,MAA9B,EAAc,CAAtC,EAAYC,MAAO,OAahD,EAawD,UAAxD,C;K;IAGf,mD;MAGuD,0B;QAAA,aAASB,K;MACzE,IAAI,CAAC,UAAD,IAAe,6BAAf,IAAiC,0BAArC,C;QACI,OAAy,SAAL,SAAK,EAAS,MAAT,C;;QAEZ,OAAO,6BAakB,mBAAS,MAAO,OAahB,IAAIB,EAA0C,MAA1C,EAakD,CAAID,EAAqD,MAAO,OAA5D,EAAoE,UAApE,C;K;IAMf,wD;MAQ8D,0B;QAAA,aAASB,K;MACHf,qBfjnBO,MAAO,KeinBa,SAAK,OfjnBIB,EinB0B,KAAM,OfjnBhC,C;MemnBd,QAAQ,C;MACR,OAAO,IAAI,cAAJ,IAA8B,SAAR,qBAAK,CAAL,CAAQ,EAAO,iBAAM,CAAN,CAAP,EAA8B,UAA9B,CAArC,C;QACI,a;;MAEJ,IAAS,mBAAL,SAAK,EAAMb,IAAI,CAAJ,IAAnB,CAAL,IAAwC,mBAAN,KAAM,EAAMb,IAAI,CAAJ,IAAnB,CAA5C,C;QACI,a;;MAEJ,OAAO,8BAAY,CAAZ,EAAe,CAAf,CAakB,W;K;IAG7B,wD;MAQ8D,0B;QAAA,aAASB,K;MACHf,iBAAiB,SAAK,O;MACtB,kBAakB,KAAM,O;MACxB,qBfxoBO,MAAO,KewoBa,UfxoBb,EewoByB,WfxoBzB,C;Me0oBd,QAAQ,C;MACR,OAAO,IAAI,cAAJ,IAA+C,SAAZB,qBAAK,aAAa,CAAb,GAAiB,CAAjB,IAAL,CAAYB,EAAO,iBAAM,cAAc,CAAd,GAakB,CAAIB,IAAN,CAAP,EAAGD,UAahD,CAAtD,C;QACI,a;;MAEJ,IAAS,mBAAL,SAAK,EAAMb,aAAa,CAAb,GAAiB,CAAjB,IAAnB,CAAL,IAAQD,mBAAN,KAAM,EAAMb,cAAc,CAAd,GAakB,CAAIB,IAAnB,CAAzD,C;QACI,a;;MAEJ,OAAO,8BAAY,aAAa,CAAb,IAAZ,EAA4B,UAA5B,CAAwC,W;K;IAMnD,8D;MAQqD,0B;QAAA,aAakB,C;MAAG,0B;QAAA,aAASB,K;MAMnE,UAAkB,M;MAL3C,IAAI,

CAAC,UAAD,IAAe,KAAM,OAAN,KAAc,CAA7B,IAAkC,6BAAtC,C;QACI,WAAiB,SAAN,KAAM,C;QACjB, ONjtBwF,kB8G3ME,oBxG45BrE,IwG55BqE,C9G2MF,EMitB7D,UNjtB6D,C;;MMotBnE,uBAAX,UAAW,EAAC ,CAAd,C;MAAkB,oC;kBAA3C,gD;QACI,kBAAkB,qBAAI,KAAJ,C;QACR,c;;UjCikXE,U;UAAhB,4BiCjkXQ,Kj CikXR,kB;YAAgB,cAAhB,UiCjkXQ,KjCikXR,S;YAAAsB,IiCjkXC,SAAH,UjCikXgB,oBiCjkXhB,CAAG,0BjCik XD,C;cAAwB,aAAO,I;cAAP,e;;;UAC9C,aAAO,K;;;QiClkXH,e;UACI,OAAO,K;;MAEf,OAAO,E;K;IAGX,ke;M ASyD,0B;QAAA,aAAkB,2B;MAAW,0B;QAAA,aAAsB,K;MACxG,IAAI,CAAC,UAAD,IAAe,KAAM,OAAN,K AAc,CAA7B,IAAkC,6BAAtC,C;QACI,WAAiB,SAAN,KAAM,C;QACjB,ONruB4F,sB8G3MM,oBxGg7BzE,IwG h7ByE,C9G2MN,EMquB7D,UNruB6D,C;;kBMyuBhG,iBAAYB,eAAX,UAAW,EAaA,2BAAb,CAAZB,WAAwD, CAAXD,U;QACI,kBAAkB,qBAAI,KAAJ,C;QACR,c;;UjCyiXE,Q;UAAhB,wBiCziXQ,KjCyiXR,gB;YAAgB,cAA hB,UiCziXQ,KjCyiXR,O;YAAAsB,IiCziXC,SAAH,UjCyiXgB,oBiCziXhB,CAAG,0BjCyiXD,C;cAAwB,aAAO,I;c AAP,e;;;UAC9C,aAAO,K;;;QiC1iXH,e;UACI,OAAO,K;;MAGf,OAAO,E;K;IAIX,8E;MAA2G,oB;QAAA,OAAg B,K;MAOrG,UAKA,M;MAXIB,cAAkB,CAAC,IAAL,GACV,aAAW,gBAAX,UAAW,EAAC,CAAd,CAAX,EAAs C,eAAT,QAAS,EAaA,gBAAb,CAATC,CADU,GAGV,SAAW,eAAX,UAAW,EAaA,2BAAb,CAAX,EAAMD,gBA AT,QAAS,EAAC,CAAd,CAAnD,C;MAEJ,IAAI,iCAAkB,yBAATB,C;QACkB,yB;QAAd,OAAc,cAAAd,C;UAAc,u B;UACV,IAAU,cAAN,KAAM,EAAC,CAAd,EAaiB,SAAJB,EAaUB,KAAvB,EAa8B,KAAM,OAApC,EAa4C,U AA5C,CAAV,C;YACI,OAAO,K;;;QAGD,2B;QAAd,OAAc,gBAAd,C;UAAc,2B;UACV,IAAU,kBAAN,KAAM,E AAkB,CAAIb,EAaqB,SAARb,EAa2B,OAA3B,EAakC,KAAM,OAAxC,EAAGD,UAAhD,CAAV,C;YACI,OAA O,O;;;MAGnB,OAAO,E;K;IAGX,qE;MAUsB,UAMA,M;MAfIB,IAAI,CAAC,UAAD,IAAe,OAAQ,KAAR,KAAG B,CAAnC,C;QACI,aAAqB,UAAR,OAAQ,C;QACrB,YAAgB,CAAC,IAAL,GAAW,sBAAQ,MAAR,EAAGB,UA AhB,CAAX,GAA4C,0BAAy,MAAZ,EAaOB,UAApB,C;QACxD,OAAW,QAAQ,CAAZ,GAAe,IAAf,GAAyB,U AAS,MAAT,C;;MAGpC,cAAkB,CAAC,IAAL,GAAW,aAAW,gBAAX,UAAW,EAAC,CAAd,CAAX,EAa6B,gB AA7B,CAAX,GAAoD,SAAW,eAAX,UAAW,EAaA,2BAAb,CAAX,EAa0C,CAA1C,C;MAEIE,IAAI,6BAAJ,C;Q ACkB,yB;oBAAd,OAAc,cAAAd,C;UAAc,yB;UACmB,sB;;Yb7sBrB,U;YAAA,Sa6sBa,Ob7sBb,W;YAAhB,OAAg B,gBAAhB,C;cAAgB,2B;cAAM,IA6sBgC,cb7sBIB,Oa6sBkB,EAAC,CAAd,sBb7sBIB,Oa6sBmD,OAAjC,ab7sBhC ,C;gBAAwB,qBAAO,O;gBAAP,uB;;;YAC9C,qBAAO,I;;;Ua4sBC,uC;UACA,IAAI,sBAAJ,C;YACI,OAAO,YAA S,cAAT,C;;;QAGD,2B;oBAAd,OAAc,gBAAd,C;UAAc,2B;UACmB,wB;;YbntBrB,U;YAAA,SamtBa,ObntBb,W; YAAhB,OAAgB,gBAAhB,C;cAAgB,6B;cAAM,IamtBgC,kBbntBIB,SamtBkB,EAakB,CAAIb,sBbntBIB,SamtBu D,OAArC,abntBhC,C;gBAAwB,uBAAO,S;gBAAP,uB;;YAC9C,uBAAO,I;;;UaktBC,2C;UACA,IAAI,wBAAJ,C; YACI,OAAO,YAAS,gBAAT,C;;;MAInB,OAAO,I;K;IAGX,iE;MAY+D,0B;QAAA,aAAkB,C;MAAG,0B;QAAA, aAAsB,K;MACtG,4BAAU,OAAV,EAAMb,UAAAnB,EAA+B,UAA/B,EAakD,KAAID,C;K;IAEJ,mE;MAYmE,0B ;QAAA,aAAkB,2B;MAAW,0B;QAAA,aAAsB,K;MACIH,4BAAU,OAAV,EAAMb,UAAAnB,EAA+B,UAA/B,EA AkD,IAAID,C;K;IAEJ,ke;MAWgE,0B;QAAA,aAAkB,C;MAAG,0B;QAAA,aAAsB,K;MACvG,gB;MAAA,8CA AU,OAAV,EAAMb,UAAAnB,EAA+B,UAA/B,EAakD,KAAID,mDAAMe,E;K;IAEvE,sE;MAYoE,0B;QAAA,aA AkB,2B;MAAW,0B;QAAA,aAAsB,K;MACnH,gB;MAAA,8CAAU,OAAV,EAAMb,UAAAnB,EAA+B,UAA/B,EA AkD,IAAID,mDAaKe,E;K;IAKtE,6D;MAM4C,0B;QAAA,aAAkB,C;MAAG,0B;QAAA,aAAsB,K;MACnF,OAA W,cAAc,gCAAZB,GACI,sBAAW,mBAAY,IAAZ,CAAX,EAa8B,UAA9B,EAa0C,UAA1C,CADJ,GNz2B4F,kB8 G3ME,oBxGujC5E,IwGvjC4E,C9G2MF,EM42BpE,UN52BoE,C;K;IM+2BhG,+D;MAQgD,0B;QAAA,aAAkB,C; MAAG,0B;QAAA,aAAsB,K;MACvF,OAAW,cAAc,gCAAZB,GACI,sBAAQ,MAAR,EAAGB,UAAhB,EAa4B,gB AA5B,EAaOC,UAApC,CADJ,GNx3B4F,kBM23B1E,MN33B0E,EM23BIE,UN33BkE,C;K;IM83BhG,iE;MAQgD ,0B;QAAA,aAAkB,2B;MAAW,0B;QAAA,aAAsB,K;MAC/F,OAAW,cAAc,gCAAZB,GACI,0BAaE,mBAAY,IA AZ,CAAf,EAakC,UAAIC,EAa8C,UAA9C,CADJ,GNp4BgG,sB8G3MM,oBxGklChF,IwGllCgF,C9G2MN,EMu4 BpE,UNv4BoE,C;K;IM04BpG,mE;MAQoD,0B;QAAA,aAAkB,2B;MAAW,0B;QAAA,aAAsB,K;MACnG,OAAW ,cAAc,gCAAZB,GACI,sBAAQ,MAAR,EAAGB,UAAhB,EAa4B,CAA5B,EAa+B,UAA/B,EAakD,IAAID,CADJ, GNn5BgG,sBMs5B1E,MNt5B0E,EMs5BIE,UNt5BkE,C;K;IMy5BpG,mD;MAM+D,0B;QAAA,aAAsB,K;MACjF, OAAI,yBAAJ,GACI,sBAAQ,KAAR,UAA4B,UAA5B,KAA2C,CAD/C,GAGI,sBAAQ,KAAR,EAaE,CAAf,EAak B,gBAAIb,EAa0B,UAA1B,KAAyC,C;K;IAIjD,kD;MAMsD,0B;QAAA,aAAsB,K;MACxE,6BAAQ,IAAR,UAA2 B,UAA3B,KAA0C,C;K;kFAE9C,4B;MAI0E,OAAA,KAAM,yBAAgB,SAAhB,C;K;IAM3C,yE;MACjC,oB;MAC A,8B;MACA,oB;MACA,kC;K;IAG8C,sF;MAAA,gE;MAC1C,iBAAqB,E;MACrB,yBAAwC,WAAX,yCAAW,EA

AS,CAAT,EAAY,oCAAM,OAAIB,C;MACxC,uBAA2B,sB;MAC3B,gBAA0B,I;MAC1B,eAAmB,C;K;0EAEnB,Y
;MACI,IAAI,uBAAkB,CAAtB,C;QACI,iBAAY,C;QACZ,gBAAW,I;;QAEX,IAAI,4CAAQ,CAAR,IAAa,uDAaA,y
CAA1B,IAAmC,uBAAkB,yCAAM,OAA/D,C;UACI,gBAAW,qCAAyB,iBAAN,yCAAM,CAAzB,C;UACX,uBA
AkB,E;;UAEIB,YAAkB,iDAAN,yCAAM,EAAa,oBAAb,C;UACIB,IAAI,SAAS,IAAb,C;YACI,gBAAW,qCAAyB
,iBAAN,yCAAM,CAAzB,C;YACX,uBAAkB,E;;YAEIB,IAAK,QAAiB,KAAjB,aAAL,EAAY,SAAU,KAAV,a;Y
ACZ,gBAAW,gCAAwB,KAAxB,C;YACX,yBAaOB,QAAQ,MAAR,I;YACpB,uBAAkB,0BAAwB,WAAU,CAAd
,GAAiB,CAAjB,GAAwB,CAA5C,K;;QAG1B,iBAAY,C;;K;oEAIpB,Y;MAKiB,Q;MAJb,IAAI,mBAAa,EAAjB,C
;QACI,iB;MACJ,IAAI,mBAAa,CAAjB,C;QACI,MAAM,6B;MACV,aAAa,mE;MAEb,gBAAW,I;MACX,iBAAY,
E;MACZ,OAAO,M;K;uEAGX,Y;MACI,IAAI,mBAAa,EAAjB,C;QACI,iB;MACJ,OAAO,mBAAa,C;K;;iDA9C5B
,Y;MAA8C,+D;K;;IAgEU,0E;MAAA,0C;QhB1mCjD,SgB2mCH,sBAAW,kBAAX,EAAuB,YAAvB,EAakD,kBA
AID,C;QAAA,OAAwE,KAAK,CAAT,GAAY,IAAZ,GAAsB,OAAM,CAAN,C;O;K;IAdIG,iF;MAUkE,0B;QAAA,
aAAkB,C;MAAG,0B;QAAA,aAAsB,K;MAAO,qB;QAAA,QAAa,C;MAC7H,wBAAwB,KAAxB,C;MAEA,OAA
O,4BAAwB,SAAXB,EAA8B,UAA9B,EAA0C,KAA1C,EAAiD,gDAAjD,C;K;IAwBiD,gF;MAAA,0C;QAAkB,Q;
QAAA,oCAAU,sBAAV,EAA0B,YAA1B,EAAqD,kBAArD,EAAwE,KAAxE,aAAsF,GAAG,UAAH,EAAe,WAA
O,OAAtB,CAAtF,O;O;K;IAIB9E,mF;MAc0E,0B;QAAA,aAAkB,C;MAAG,0B;QAAA,aAAsB,K;MAAO,qB;QAA
A,QAAa,C;MACrI,wBAAwB,KAAxB,C;MACA,qBAAGC,OAAX,UAAW,C;MAEHc,OAAO,4BAAwB,SAAXB,E
AA8B,UAA9B,EAA0C,KAA1C,EAAiD,sDAAjD,C;K;IAIX,wC;MnBltCI,IAAI,EmBmtCI,SAAS,CnBntCb,CAAJ,
C;QACI,cmBktCkB,8C;QnBjtCIB,MAAM,gCAAyB,OAAQ,WAAjC,C;;K;ImBkuCgE,sD;MAAA,qB;QAAE,yCA
AU,EAAV,C;O;K;IAZhf,mE;MAWmE,0B;QAAA,aAAsB,K;MAAO,qB;QAAA,QAAa,C;MACzG,OAASe,OAAt
E,+BAAkB,UAAIB,UAA2C,UAA3C,EAA+D,KAA/D,CAAsE,EAAI,iCAAJ,C;K;IAE1E,yD;MAWyD,0B;QAAA,
aAAsB,K;MAAO,qB;QAAA,QAAa,C;MAC/F,IAAI,UAAW,OAAX,KAAmB,CAAvB,C;QACI,gBAAgB,WAAW,
CAAX,C;QAChB,IAAI,EAAC,SAh/BuC,YAAU,CAG/BID,CAAJ,C;UACI,OAAO,mBAAM,SAAN,EAAiB,UAAj
B,EAA6B,KAA7B,C;;MAI2E,kBAAb,cAAtE,+BAAkB,UAAIB,UAA2C,UAA3C,EAA+D,KAA/D,CAAsE,C;Mb
8OtE,kBAAM,iBAAa,qCAAwB,EAAxB,CAAb,C;MAuEA,Q;MAAA,6B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT
,WAAy,WatTgF,uBbsTIE,IatTkE,CbsThF,C;;MatThB,ObuTO,W;K;Ia5SmE,wD;MAAA,qB;QAAE,yCAAU,EAA
V,C;O;K;IARhf,qE;MAOiE,0B;QAAA,aAAsB,K;MAAO,qB;QAAA,QAAa,C;MACvG,OAASe,OAAtE,6BAAkB,
UAAIB,UAA2C,UAA3C,EAA+D,KAA/D,CAAsE,EAAI,mCAAJ,C;K;IAE1E,2D;MAOuD,0B;QAAA,aAAsB,K;
MAAO,qB;QAAA,QAAa,C;MAC7F,IAAI,UAAW,OAAX,KAAmB,CAAvB,C;QACI,OAAO,mBAAoB,oBAAd,
WAAW,CAAX,CAAc,CAApB,EAAgC,UAAhC,EAA4C,KAA5C,C;;MAG+E,kBAAb,cAAtE,6BAAkB,UAAIB,U
AA2C,UAA3C,EAA+D,KAA/D,CAAsE,C;MbqNtE,kBAAM,iBAAa,qCAAwB,EAAxB,CAAb,C;MAuEA,Q;MA
AA,6B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,WAAy,Wa7RgF,uBb6RIE,Ia7RkE,Cb6RhF,C;;Ma7RhB,Ob8RO,
W;K;Ia3RX,0D;MASI,wBAAwB,KAAxB,C;MAEA,oBAAoB,C;MACpB,gBAAgB,sBAAQ,SAAR,EAAmB,aAA
nB,EAAkC,UAAIC,C;MACHB,IAAI,cAAa,EAAb,IAAmB,UAAS,CAAhC,C;QACI,OAAO,OAAO,SAAK,WAAZ,
C;;MAGX,gBAAgB,QAAQ,C;MACxB,aAAa,iBAAsB,SAAJ,GAAqB,eAAN,KAAM,EAAa,EAAb,CAArB,GAA2
C,EAA7D,C;;QAET,MAAO,WA36B6E,8BA26B/D,aA36B+D,EA26BhD,SA36BgD,CAAkC,WA26B/G,C;QACP,
gBAAgB,YAAy,SAAU,OAAtB,I;QAEhB,IAAI,aAAa,MAAO,KAAP,MAAe,QAAQ,CAAR,IAAf,CAAjB,C;UAA
2C,K;QAC3C,YAAy,sBAAQ,SAAR,EAAmB,aAAnB,EAAkC,UAAIC,C;;MACP,sBAAa,EAAb,C;MAET,MAAO
,WAI7BiF,8BAk7BnE,aAl7BmE,EAk7BpD,gBAI7BoD,CAAkC,Wak7BnH,C;MACP,OAAO,M;K;2EAGX,mC;M
AOmD,qB;QAAA,QAAa,C;MAAmB,OAAA,KAAM,eAAM,SAAN,EAAY,KAAZ,C;K;+FAEzF,mC;MAU6D,qB;
QAAA,QAAa,C;MAAuB,OAAA,KAAM,yBAAGB,SAAhB,EAA8B,KAAtB,C;K;IAEvG,iC;MAK2D,mCAAGB,M
AAhB,EAAwB,IAAxB,EAA8B,IAA9B,E;K;IAE3D,0B;MAKgD,OAAe,UAAf,uBAAe,C;K;IAqB/D,uD;MAQsB,
Q;MAPIB,IAAI,iCAAkB,yBAAtB,C;QACI,OAAy,SAAL,SAAK,EAAO,KAAP,EAA2B,IAA3B,C;;MAGhB,IAAI
,cAAS,KAAb,C;QAAoB,OAAO,I;MAC3B,IAAI,qBAAGB,aAAhB,IAAiC,SAAK,OAAL,KAAe,KAAM,OAAID,
C;QAAkE,OAAO,K;MAEvD,uB;MAAIB,aAAU,CAAV,gB;QACI,IAAI,CAAS,SAAR,qBAAK,CAAL,CAAQ,EA
AO,iBAAM,CAAN,CAAP,EAA8B,IAA9B,CAAb,C;UACI,OAAO,K;;MAIf,OAAO,I;K;IAGX,6C;MAQsB,Q;M
APIB,IAAI,iCAAkB,yBAAtB,C;QACI,OAAO,kBAAQ,KAAR,C;;MAGX,IAAI,cAAS,KAAb,C;QAAoB,OAAO,I;
MAC3B,IAAI,qBAAGB,aAAhB,IAAiC,SAAK,OAAL,KAAe,KAAM,OAAID,C;QAAkE,OAAO,K;MAEvD,uB;
MAAIB,aAAU,CAAV,gB;QACI,IAAI,qBAAK,CAAL,MAAW,iBAAM,CAAN,CAAF,C;UACI,OAAO,K;;MAIf,

WAAN,KAAM,mC;K;iDAE/C,iB;MAK0C,OAAM,aAAN,KAAM,mC;K;iDAEhD,iB;MAS4C,OAAM,aAAN,KAAM,mC;K;gDAGlD,iB;MAKwC,OAAM,WAAN,KAAM,kC;K;gDAE9C,iB;MAKyC,OAAM,aAAN,KAAM,kC;K;gDAE/C,iB;MAS2C,OAAM,aAAN,KAAM,kC;K;iDAEjD,iB;;QAY4C,OACxC,cAAc,KAAAd,EAAiC,KAAjC,C;;QACF,+C;UACE,MAAM,6BAAyB,sCAAmC,KAAAnC,OAAzB,EAAsE,CAAtE,C;;UAHkC,O;;K;0DAM5C,iB;;QAMqD,OACjD,cAAc,KAAAd,EAAiC,IAAjC,C;;QACF,+C;UACE,MAAM,6BAAyB,0CAAuC,KAAvC,OAAzB,EA0E,CAA1E,C;;UAH2C,O;;K;uDAMrD,iB;;QAWmD,OAC/C,cAAc,KAAAd,EAAiC,KAAjC,C;;QACF,+C;UAFiD,OAG/C,I;;UAH+C,O;;K;gEAMnD,iB;;QAK4D,OACxD,cAAc,KAAAd,EAAiC,IAAjC,C;;QACF,+C;UAF0D,OAGxD,I;;UAHwD,O;;K;;IA/XhE,0C;MAAA,yC;QAAA,wB;;MAAA,kC;K;oCAwYA,Y;MAC6C,kBAAy,YAAD,aAAx,EAzZK,eAAS,QAAT,GAAqB,CAyZ1B,C;K;qCAE7C,iB;MAiBW,Q;MATH,IAAA,IAAK,aAAL,C;QACI,IAAI,KAAM,WAAN,IAAqB,IAAK,WAAL,KAakB,KAAM,WAAXB,gBAAoC,CAA7D,C;UACI,OAAO,I;;UAEP,MAAM,gCAAyB,2EAAzB,C;WAEd,IAAA,KAAM,aAAN,C;QAASB,OAAO,K;MAI7B,KA7a0C,eAAS,QAAT,GAAqB,CA6a/D,OAA0B,KA7agB,WAAS,QAAT,GAAqB,CA6a/D,E;QACI,aAAa,IAAK,QAAL,KAAa,KAAM,QAAnB,C;QAET,uB;UACI,iCAA0B,MAA1B,C;;UAEA,kCAA2B,MAA3B,C;aAGZ,IAAA,IAAK,eAAL,C;QACI,mCAAqB,IAAK,QAA1B,EAAiC,KAAM,QAAvC,C;;QAEA,mCAAqB,KAAM,QAA3B,EAakC,IAAK,QAAvC,C;MABR,W;K;gDAiBJ,kC;MAGW,Q;MAFP,kBAAkB,cAAc,UAAAd,C;MACIB,mBAAmB,eAAa,WAAb,C;MACZ,IAAI,8EAAsC,mBAAtC,CAAJ,C;QACH,yBAAyB,oBAAa,cAAc,WAAAd,CAAAb,C;QACzB,uBAAgB,cAAc,YAAAd,MAA8B,kBAA9B,CAAhB,C;;QAEA,wBAA8B,WAAb,YAAa,yBAAsB,UAAtB,CAA9B,C;;MAJJ,W;K;sCAQJ,iB;MAMuD,wBAAS,KAAD,aAAR,C;K;uCAEvD,iB;MAQe,UAAUJ,M;MAXP,IAAI,iBAAJ,C;QAEQ,cAAS,CAAT,C;UAAc,MAAM,gCAAyB,mEAAzB,C;aACpB,YAAQ,CAAR,C;UAAa,W;;UACL,OAAC,IAAD,a;QAHZ,W;;MAMJ,IAAI,UAAS,CAAAb,C;QAAGB,OAAO,qC;MAEvB,YAAy,Y;MACZ,aAAa,mCAAQ,KAAR,E;MACN,IAAI,kBAAJ,C;QACH,IAAI,yEAAJ,C;UAEI,yBAAgB,MAAhB,C;;UAEA,IAAI,sCAAS,KAAT,IAAkB,KAAIB,CAAJ,C;YACI,mCAA0B,MAA1B,C;;YAEA,aAAa,cAAc,KAAAd,C;YACb,eAAe,eAAQ,cAAc,MAAd,CAAR,C;YACf,mBAAmB,oCAAS,KAAT,E;YACnB,kBAAkB,iBAAe,cAAc,sCAAW,KAAX,EAAd,CAAF,C;YACIB,IAAI,4CAAe,KAf,IAAwB,MAAxB,KAakC,gBAAgB,YAAhB,gBAAgC,CAAtE,C;cACI,0BAA6B,WAAZ,WAAy,EAAS,8BAAa,UAAb,CAAT,CAA7B,C;;cAEA,SAAI,YAAM,WAAN,KAAM,CAAN,EAAMB,WAAN,KAAM,CAAnB,IAA0B,CAA9B,GAAiC,yCAAjC,GAA+C,qD;;;QAK3D,IAAI,sCAAS,KAAT,IAAkB,KAAIB,CAAJ,C;UACI,0BAAwB,WAAP,MAAO,EAAS,8BAAa,UAAb,CAAT,CAAxB,C;;UAEA,SAAI,YAAM,WAAN,KAAM,CAAN,EAAMB,WAAN,KAAM,CAAnB,IAA0B,CAA9B,GAAiC,yCAAjC,GAA+C,qD;;;MAvBvD,a;K;uCA4BJ,iB;MASI,eAAqB,WAAAN,KAAM,C;MACrB,IAAa,QAAT,KAAuB,KAA3B,C;QACI,OAAO,mBAAM,QAAN,C;;MAGX,WAAW,kB;MACX,aAAa,sBAAS,IAAT,IAAiB,K;MAC9B,OAAc,aAAP,MAAO,EAAW,IAAX,C;K;qCAGIB,iB;MAQe,Q;MADX,IAAI,UAAS,CAAAb,C;QAEQ,sB;UAAgB,gD;aAchB,sB;UAAgB,4D;;UACR,MAAM,gCAAyB,4DAAzB,C;QAHIB,W;;MAMJ,IAAI,kBAAJ,C;QACI,OAAO,gBAAgB,qCAAQ,KAAR,EAAhB,C;;QAEp,IAAI,iBAAJ,C;UACI,OAAO,mBAAa,WAAN,KAAM,CAAAb,C;QAEX,aAAa,qCAAQ,KAAR,E;QAEb,IAAI,kEAAgC,mBAAhC,CAAJ,C;UACI,UAAU,cAAc,sBAAS,oCAAS,KAAT,EAAT,CAAd,0BAA0C,KAA1C,E;UACV,OAAO,gBAAgB,cAAc,MAAd,MAAwB,GAAxB,CAAhB,C;;QAEX,OAAO,iBAAiB,MAAjB,C;;K;qCAIf,iB;MAOI,eAAqB,WAAN,KAAAM,C;MACrB,IAAa,QAAT,KAAuB,KAAvB,IAAgC,aAAY,CAAhD,C;QACI,OAAO,iBAAI,QA AJ,C;;MAGX,WAAW,kB;MACX,aAAa,sBAAS,IAAT,IAAiB,K;MAC9B,OAAc,aAAP,MAAO,EAAW,IAAX,C;K;oCAGIB,iB;MAEI,kBAAkB,SAAM,IAAK,cAAX,EAAwB,KAAM,cAA9B,C;MACIB,OAAO,IAAK,kBAAS,WAAT,CAAL,GAA6B,KAAM,kBAAS,WAAT,C;K;oCAG9C,Y;MACmC,oCAAW,C;K;oCAE9C,Y;MACmC,oCAAW,C;K;oCAE9C,Y;MACmC,+BAAy,yCAAS,WAArB,KAAiC,wBAAy,qDAAa,WAAzB,C;K;kCAEpE,Y;MACiC,QAAC,iB;K;yFAGC,Y;MAAQ,OAAI,iBAAJ,GAAmB,IAAD,aAAIB,GAA6B,I;K;yCAExE,iB;MACI,kBAAkB,IAAK,WAAL,KAakB,KAAM,WAAXB,C;MACIB,IAAI,yBAAc,CAAd,IAAMB,CAAA,WAAy,QA AZ,GAAwB,CAAxB,MAA6B,CAApD,C;QACI,OAAO,IAAK,WAAS,iBAAU,KAAM,WAAhB,C;MAEZB,QAAQ,CA11BsC,eAAS,QAAT,GAAqB,CA01B3D,KAAyB,KA11Ba,WAAS,QAAT,GAAqB,CA01B3D,K;MACR,OAAW,iBAAJ,GAakB,CAAC,CAAD,IAAIB,GAA0B,C;K;uHAMrC,kB;MAeI,OAAO,OAAO,gBAAP,EAAoB,mBAAPB,EAAoC,qBAAPC,EAAsD,qBAAtD,EAAwE,yBAAxE,C;K;uHAGX,kB;MAcI,OAAO,OAAO,iBAAP,EAAqB,qBAArB,EAAuC,qBAAvC,EAAYd,yBAAzD,C;K;uHAGX,kB;MAaI,OAAO,OAAO,mBAAP,EAAuB,qBAAvB,EAAYC,yBAAzC,C;K;uHAGX,kB;MAYI,OAAO,OAAO,mBAAP,EAAuB,yBAAvB,C;K;0FAKP,Y;MAAQ,OAAI,iBAAJ,GAakB,CAAIB,GAA0

B,6CAAE,EAaf,EAAMb,Q;K;4FAIrD,Y;MAAQ,OAAI,iBAAJ,GAakB,CAAIB,GAA0B,+CAAiB,EAajB,EAaq
B,Q;K;4FAIvD,Y;MAAQ,OAAI,iBAAJ,GAakB,CAAIB,GAA0B,+CAAiB,EAajB,EAaqB,Q;K;gGAIvD,Y;MAC
LsB;QADI,OACY,C;WACHb,wB;QAFI,OAey,cAAc,wCAAQ,IAAR,EAAd,CAA6B,Q;;QAFzC,OAGK,wCAAQ,
UAAR,EAauB,Q;K;0CAMxC,gB;MAQiB,UAN,M;MAAM,sB;MACT,iBAAA,yCAAS,WAAT,E;QAA4B,SAA
P,wCAAO,kB;WAC5B,iBAAA,qDAAa,WAAb,E;QAAgC,SAAP,wCAAO,kB;;QAG5B,6BAAoB,YAAM,WAA1
B,EAAsC,kBAAtC,EAAMd,IAAnD,C;;MALR,a;K;wCAUJ,gB;MAUiB,UAN,M;MAAM,sB;MACT,iBAAA,yC
AAS,WAAT,E;;WACA,iBAAA,qDAAa,WAAb,E;;;QACQ,+BAAoB,YAApB,EAA2B,kBAA3B,EAAwC,IAAxC,
C;MAHZ,a;K;uCAOJ,gB;MAUI,OAAa,WAAb,oBAAO,IAAP,CAAA,4BAAyD,Q;K;kFAKhD,Y;MAAQ,6D;K;mF
AKP,Y;MAAQ,8D;K;qFAKN,Y;MAAQ,gE;K;qFAKR,Y;MAAQ,gE;K;0FAKH,Y;MAAQ,qE;K;0FAKR,Y;MAA
Q,qE;K;yFAKT,Y;MAAQ,oE;K;uFASrC,Y;MAAQ,2D;K;wFAQR,Y;MAAQ,4D;K;0FAQR,Y;MAAQ,8D;K;0FA
QR,Y;MAAQ,8D;K;+FAQR,Y;MACI,OAAW,uBAAgB,eAApB,GAAgC,YAAhC,GAA2C,4D;K;+FAAtD,Y;MAA
Q,mE;K;8FAyR,Y;MAEW,Q;MADP,YAAy,Y;MAER,uB;QAae,Y;WACf,8C;;WACA,+C;;QACQ,qBAAc,KAA
d,C;MAJZ,W;K;2CAUR,Y;MASuC,8B;K;4CAEvC,Y;MASwC,+B;K;kCAExC,Y;MAuBwC,Q;MAAA,sB;MACp
C,qB;QAD8B,OACxB,I;WACN,iBAAA,yCAAS,WAAT,E;QAF8B,OAET,U;WACrB,iBAAA,qDAAa,WAAb,E;Q
AH8B,OAGL,W;;QAErB,iBAAiB,iB;Q8HzhBF,gBAAhB,sB;Q9H2hBK,e;UAAgB,yBAAO,EAAP,C;QACF,YAA
d,kB;QA9RD,WAAO,iB;QAAP,YAAoB,oB;QAApB,cAAoC,sB;QAApC,cAAsD,sB;QAAtD,kBAAwE,0B;QAsS/
D,0B;QAPI,cAAc,iB;QACd,eAAe,UAAS,C;QACxB,iBAAiB,YAAW,C;QAC5B,iBAAiB,YAAW,CAAX,IAAgB,
gBAAe,C;QACHd,iBAAiB,C;QACjB,IAAI,OAAJ,C;UACI,yBAAO,IAAP,CAAA,gBAAO,GAAP,C;UACb,+B;;Q
AEJ,IAAI,aAAa,YAAy,cAAc,UAA1B,CAAb,CAAJ,C;UACI,IAAI,6DAAe,CAAnB,C;YAAsB,yBAAO,EAAP,C;
UACtB,yBAAO,KAAP,CAAc,gBAAO,GAAP,C;;QAEIB,IAAI,eAAe,eAAe,YAAy,OAA3B,CAAf,CAAJ,C;UACI
,IAAI,6DAAe,CAAnB,C;YAAsB,yBAAO,EAAP,C;UACtB,yBAAO,OAAP,CAAgB,gBAAO,GAAP,C;;QAEpB,I
AAI,UAAJ,C;UACI,IAAI,6DAAe,CAAnB,C;YAAsB,yBAAO,EAAP,C;UAEIB,gBAAW,CAAX,IAAgB,OAAhB,I
AA2B,QAA3B,IAAuC,UAAvC,C;YACI,mCAAiB,OAAjB,EAA0B,WAA1B,EAuA,C,CAAvC,EAA0C,GAA1C,E
AA2D,KAA3D,C;eACJ,mBAAe,OAAf,C;YACI,mCAAiB,cAAc,OAAd,IAAjB,EAA0C,cAAc,OAAxD,EAAMe,C
AAAnE,EAAsE,IAAtE,EAaWf,KAAxF,C;eACJ,mBAAe,IAAf,C;YACI,mCAAiB,cAAc,IAAd,IAAjB,EAAsC,cAA
c,IAApD,EAA2D,CAA3D,EAA8D,IAA9D,EAAGf,KAAhF,C;;YAEA,yBAAO,WAAP,CAAoB,gBAAO,IAAP,C;;
QAGhC,IAAI,cAAc,aAAa,CAA/B,C;UAAkC,yBAAO,CAAP,EAAU,EAAV,CAAE,gBAAO,EAAP,C;QAvC/B,O
Ox1B3B,SuHoUqC,W;;K;4C9HikB5C,yE;MACI,yBAAO,KAAP,C;MACA,IAAI,eAAc,CAAIB,C;QACI,yBAAO,
EAAP,C;QACA,iBAAuC,WAAtB,UAAW,WAAW,EAAS,cAAT,EAAYB,EAazB,C;QACR,sB;;UsB5zBzB,Q;UA
AA,OAAQ,WAAR,etB4zBc,UsB5zBd,CAAQ,CAAR,W;UAAAd,OAAc,cAAAd,C;YAAc,uB;YACV,ItB2zBiD,UsB3
zBnC,YtB2zBU,UsB3zBV,YAAK,KAAL,EtB2zBmC,MAAM,EsB3zBvD,C;cACI,qBAAO,K;cAAP,uB;;UAGR,q
BAAO,E;;QtBuzBC,oBAAoB,qBAAuC,CAAvC,I;QAEhB,KAAC,SAAD,IAAc,gBAAgB,CAA9B,C;UAAmC,8B
AAy,UAAZ,EAawB,CAAxB,EAA2B,aaa3B,C;;UAC3B,8BAAy,UAAZ,EAawB,CAAxB,EAA2B,CAAC,CAA
C,gBAAgB,CAAhB,IAAD,IAAsB,CAAtB,IAAD,IAA4B,CAA5B,IAA3B,C;;MAGhB,yBAAO,IAAP,C;K;0CAGJ,
0B;MAGBwC,wB;QAAA,WAAgB,C;MIn9BxD,IAAI,EJ09BQ,YAAy,CIP9BpB,CAAJ,C;QACI,cJm9ByB,oD;QII
9BzB,MAAM,gCAAyB,OAAQ,WAAjC,C;;MJm9BN,aAAa,sBAAS,IAAT,C;MACb,IAAW,WAAP,MAAO,CAA
X,C;QAAyB,OAAO,MAAO,W;MACvC,OAAO,sBAAsB,MAAtB,EAuA,C,eAAT,QAAS,EAaA,EAAb,CAAvC,IA
AgE,UAAAL,IAAK,C;K;qCAI3E,Y;M8HvmBuB,gBAAhB,sB;M9HqnBH,IAAI,iBAAJ,C;QAAkB,yBAAO,EAAP,
C;MACIB,yBAAO,IAAP,C;MAC4B,YAAAd,kB;MAxWP,YAAO,kB;MAAP,cAAqB,sB;MAArB,cAAuC,sB;MAA
vC,kBAAyD,0B;MAyW5D,cACY,K;MACZ,IAAI,iBAAJ,C;QAEI,wB;;MAEJ,eAAe,oB;MACf,iBAAiB,YAAW,C
AAX,IAAgB,gBAAe,C;MACHd,iBAAiB,YAAW,CAAX,KAAiB,cAAc,QAA/B,C;MACjB,IAAI,QAAJ,C;QACI,y
BAAO,OAAP,CAAc,gBAAO,EAAP,C;;MAEIB,IAAI,UAAJ,C;QACI,yBAAO,OAAP,CAAgB,gBAAO,EAAP,C;;
MAEpB,IAAI,eAAe,CAAC,QAAD,IAAa,CAAC,UAA7B,CAAJ,C;QACI,mCAAiB,OAAjB,EAA0B,WAA1B,EA
AuC,CAAvC,EAA0C,GAA1C,EAA2D,IAA3D,C;;MApBuB,00x7B5B,SuHoUqC,W;K;;;kC9H5YhD,Y;MAAA,
c;MAuBiD,2D;MAvBjD,a;K;gCAAA,iB;MAAA,2IAuBiD,gDAvBjD,G;K;IA8hCA,qC;MAIW,Q;MAAA,IAAI,6D
AAJ,C;QACH,uBAAgB,4BAAiC,oBAAL,SAAK,CAAJ,C,EAA2C,IAA3C,yCAAhB,C;;QAES,oBAAT,8BAAS,EA
AW,IAAX,C;MAHb,W;K;IAMJ,uC;MAII,kBAakB,4BAA4B,SAA5B,0CAAiE,IAAjE,C;MACIB,IAAa,WAAD,a
AAR,yDAAsB,WAAtB,CAAJ,C;QACI,OAAO,gBAAgB,4BAA4B,SAA5B,EAAkC,IAAIC,yCAAhB,C;;QAEp,aA

Aa,sBAaOb,SAApB,EAA0B,IAA1B,0C;QACb,OAAO,iBAAwB,WAAP,MAAO,yBAAsB,UAAtB,CAAxB,C;;K;I
Alf,uC;MAaW,Q;MAHP,gBAAgB,oBAaOb,SAApB,EAA0B,IAA1B,yC;MIviChB,IAAI,CJwiCI,CAAW,QAAV,
SAAU,CiXiCnB,C;QACI,cJuiC0B,+B;QItiC1B,MAAM,gCAAYB,OAAQ,WAAjC,C;;MJuiCV,YAAsB,YAAV,SA
AU,C;MACf,IAAI,sEAAqB,SAArB,CAAJ,C;QACH,uBAAgB,KAAhB,C;;QAEA,aAAwE,YAA3D,oBAaOb,SAA
pB,EAA0B,IAA1B,0CAA2D,C;QACxE,kCAA2B,MAA3B,C;;MAJJ,W;K;IAgBuB,oC;MAAQ,oE;K;IAOP,sC;MA
AQ,sE;K;IAWN,sC;MAAQ,sE;K;IAQV,qC;MAAQ,qE;K;IAOP,uC;MAAQ,uE;K;IAWN,uC;MAAQ,uE;K;IAQX,
qC;MAAQ,qE;K;IAOP,uC;MAAQ,uE;K;IAWN,uC;MAAQ,uE;K;IAQhB,gC;MAAQ,gE;K;IAOP,kC;MAAQ,kE;K
;IAWN,kC;MAAQ,kE;K;IAQX,gC;MAAQ,gE;K;IAOP,kC;MAAQ,kE;K;IAWN,kC;MAAQ,kE;K;IAQb,8B;MAA
Q,8D;K;IAOP,gC;MAAQ,gE;K;IAWN,gC;MAAQ,gE;K;IAQZ,6B;MAAQ,6D;K;IAOP,+B;MAAQ,+D;K;IAWN,+
B;MAAQ,+D;K;yEAG/B,+B;MAIqE,8BAAW,SAAX,C;K;2EAERe,+B;MAUwE,8BAAW,SAAX,C;K;IAIxE,yC;
MACI,aAAa,KAAM,O;MACnB,IAAI,WAAU,CAAd,C;QAAiB,MAAM,gCAAYB,qBAAZB,C;MACvB,YAAY,C;
MACZ,aAAa,gCAAS,K;MACTb,qBAaQb,U;MACrB,QAAM,iBAAM,KAAN,CAAN,C;aACI,E;aAAa,E;UAAy,
qB;UAAZ,K;;MAEJ,cAAc,QAAQ,C;MACTb,iBAaiB,WAAiB,aAN,KAAM,EAAW,EAAX,C;MAE9B,cAAU,K
AAV,C;QACI,MAAM,gCAAYB,eAAzB,C;WACV,qBAAM,KAAN,MAAgB,EAhB,C;QACI,IAAI,mCAAW,M
AAf,C;UAAuB,MAAM,+B;QAC7B,sBAAsB,K;QACtB,sBAAsB,K;QACtB,eAA8B,I;QAC9B,OAAO,QAAQ,MA
Af,C;UACI,IAAI,iBAAM,KAAN,MAAgB,EAAPB,C;YACI,IAAI,mBAAM,mCAAW,MAALC,C;cAA0C,MAA
M,+B;YACHD,kBAakB,I;YACIB,Q;;UAekB,iBAae,K;UA+EjD,QAHgC,U;UAIhC,Y;YAAO,eAhFqB,KAgFjB,
O;YAAJ,S;cAAc,SAAU,YAhFH,KAgFG,YAAK,CAAL,E;cAAV,OAhFqC,CAAM,kBAAK,EAAL,CAAN,qCAA
kB,2C;;;YAgFnC,a;;UAhF7B,gBAAgB,KiBv1CgE,WjBmqClF,UiBnqCkF,EjBwqCrF,CiBxqCqF,C;UjBwlChF,IA
AI,SuBrhCgC,YAAU,CvBqhC9C,C;YAAyB,MAAM,+B;UAC/B,gBAAS,SAAU,OAAAnB,I;UACqB,cAAU,K;UsB
zrCpC,U;UAAA,IAAI,WAAS,CAAT,IAAc,WAAS,iBtByrCP,KsBzrCO,CAA3B,C;YAAA,StByrCoB,KsBzrCkB,
YAAI,OAAJ,C;;YtByrCO,MAAM,gCAAYB,qCAAzB,C;;UAA9C,qB;UACA,qB;UACA,WAAW,sBAAsB,QAAtB
,EAAgC,eAAhC,C;UACX,IAAI,YAAY,IAAZ,IAAoB,yBAAY,IAAZ,MAAxB,C;YAA0C,MAAM,gCAAYB,yCA
AzB,C;UACHD,WAAW,I;UACX,eAAyB,WAAV,SAAU,EAAQ,EAAR,C;UACzB,IAAI,+CAAgC,WAAW,CAA/
C,C;YACI,YAAY,SiBjmCgE,WjBimC5C,CiBjmC4C,EjBimCzC,QiBjmCyC,C;YjBkmC5E,4BAA2C,aAAjC,0BA
A0B,KAA1B,CAAiC,EAAW,IAAX,CAA3C,C;YACA,4BAAMd,aAAX,SAA9B,SiBtmCmD,WjBsmC/B,QiBtmC
+B,CjBsmCrB,CAAW,EAAW,IAAX,CAAnD,C;;YAEA,4BAA+C,aAArC,0BAA0B,SAA1B,CAAqC,EAAW,IAA
X,CAA/C,C;;aAIZ,c;QACI,MAAM,+B;;QACV,IAAM,cAAN,KAAM,EAAc,KAAd,EAAqB,cAArB,EAAqC,CA
ArC,EQ/xCH,MAAO,KR+xCmD,SAAS,KAAT,IQ/xCnD,ER+xCmE,cAAe,OQ/xCIF,CR+xCJ,EAA4G,IAA5G,CA
AN,C;UACI,SAAS,gCAAS,S;;UAIIB,iBAA8B,I;UAC9B,iBAAiB,K;UACjB,kBAakB,CAAC,O;UACnB,IAAI,W
AAW,iBAAM,KAAN,MAAgB,EAA3B,IAAwC,QAAN,KAAM,CAAN,KAAGB,EAAtD,C;YACI,cAAc,I;YACd,I
AAI,oCAAW,uBAAX,EAAW,MAAX,CAAJ,C;cAAyB,MAAM,gCAAYB,eAAzB,C;;UAEnC,OAAO,QAAQ,MA
Af,C;YACI,IAAI,cAAc,WAAIB,C;cA8CZ,UA7CwC,K;cA8CxY,gBAAO,mBA9CiB,KA8Cb,O;gBAAJ,W;kBA
Ac,SA9C4B,UA8CIB,YA9CP,KA8CO,YAAK,GAAL,EA9CkB,MAAM,E;;;gBA8Cd,iB;;cA9CzB,QA+CT,G;;YA
7CK,aAAa,I;YACS,mBAae,K;YA0CjD,UAHgC,Y;YAIhC,Y;cAAO,mBA3CqB,KA2CjB,O;cAAJ,W;gBAAc,WA
AU,YA3CH,KA2CG,YAAK,GAAL,E;gBAAV,SA3CqC,CAAM,kBAAK,EAAL,CAAN,uCAAKB,oBAAM,E;;;c
A2CzC,iB;;YA3C7B,kBAAgB,KiB5nCgE,WjBmqClF,YiBnqCkF,EjBwqCrF,GiBxqCqF,C;YjB6nChF,IAAI,WuB
IjCgC,YAAU,CvB0jC9C,C;cAAyB,MAAM,+B;YAC/B,gBAAS,WAAU,OAAAnB,I;YACqB,mBAae,K;YAuChD,
UAHgC,Y;YAIhC,Y;cAAO,mBAxCoB,KAwChB,O;cAAJ,W;gBAAc,WAAU,YAxCJ,KAwCI,YAAK,GAAL,E;g
BAAV,SAxCoC,CAAM,kBAAK,GAAL,CAAN,mC;;;cAwChB,iB;;YAx7B,eAAe,KiB/nCiE,WjBmqClF,YiBnq
CkF,EjBwqCrF,GiBxqCqF,C;YjBgoChF,gBAAS,QAAS,OAAIB,I;YACA,aAAW,wBAAwB,QAAXB,C;YACX,IA
AI,cAAY,IAAZ,IAAoB,2BAAY,MAAZ,MAAxB,C;cAA0C,MAAM,gCAAYB,yCAAzB,C;YACHD,aAAW,M;YA
CX,iBAAYB,WAAV,WAAU,EAAQ,EAAR,C;YACzB,IAAI,aAAW,CAAF,C;cACI,cAAY,WiBtoCgE,WjBsoC5C,
CiBtoC4C,EjBsoCzC,UiBtoCyC,C;cjBuoC5E,4BAAYB,aAAT,OAAN,OAAM,CAAS,EAAW,MAAX,CAAzB,C;c
ACA,4BAAMd,aAAX,SAA9B,WiB3oCmD,WjB2oC/B,UiB3oC+B,CjB2oCrB,CAAW,EAAW,MAAX,CAAnD,C;
cACA,IAAI,QAAQ,MAAZ,C;gBAAoB,MAAM,gCAAYB,mCAAzB,C;;cAE1B,4BAA6B,aAAT,OAAV,WAAU,C
AAS,EAAW,MAAX,CAA7B,C;;;MAKhB,OAAW,UAAJ,GAaiB,MAAD,aAAhB,GAA6B,M;K;IAIxC,0C;MAC
I,aAAa,KAAM,O;MACnB,iBAAiB,C;MACjB,IAAI,SAAS,CAAT,IAAc,YAAY,IAAZ,mBAAM,CAAN,EAALB,C

;QAAoC,+B;;MACHC,YAAC,SAAS,UAAT,IAAD,IAAwB,E;MAAxS,S;QAA4D,gBAA7B,yBAaKB,iBAAN,KAAM,CAAIB,C;QAA6B,c;;UU4ThD,U;UADhB,IAAI,wCAAsB,mBAA1B,C;YAAqC,aAAO,I;YAAP,e;;UACrB,6B;UAAhB,OAAgB,gBAAhB,C;YAAgB,2B;YAAM,IAAI,CV5T4C,CAAa,kBAAK,EAAL,CAAb,oCU4TjC,OV5TiC,EU4ThD,C;cAAyB,aAAO,K;cAAP,e;;;UAC/C,aAAO,I;;;QV7TyD,iB;;MAAhE,S;QAEI,OAAW,iBAAM,CAAN,MAAY,EAAhB,sD;;MAGX,OAAiB,WAAN,KAAM,EAAW,GAAX,CAAV,GAAyC,OAAR,QAAN,KAAM,EAAK,CAAL,CAAQ,CAAzC,GAA6D,OOAN,KAAM,C;K;IAKxE,0D;MAII,QAHgC,U;MAIhC,OAAO,IAAI,gBAAJ,IAJqC,SAIvB,CAAU,iCAAK,CAAL,EAAV,CAArB,C;QAAyC,a;;MAJzC,OiBnqC4F,oBjBmqClF,UiBnqCkF,EjBwqCrF,CiBxqCqF,C;K;IjBqqChG,qD;MACI,QAAQ,U;MACR,OAAO,IAAI,gBAAJ,IAAc,UAAU,iCAAK,CAAL,EAAV,CAArB,C;QAAyC,a;;MACzC,OAAO,C;K;;;IAmBX,8B;MAA+C,qCAAQ,OAAR,E;K;IAC/C,+B;MAAgD,2CAAS,OAAT,E;K;IAEHd,sC;MAAiD,oBAAS,sBAAGB,CAAhB,CAAT,C;K;IACjD,wC;MAAmD,oBAAU,uBAAiB,CAAjB,CAAD,yBAAuB,CAAvB,EAAT,C;K;IACnD,oD;MAAoE,oBAAU,sBAAGB,CAAhB,CAAD,yBAA sB,iBAAAtB,EAAT,C;K;IACpE,0C;MACI,IAAI,sEAAqB,SAArB,CAAJ,C;QAAA,OACI,gBAAgB,KAAhB,C;;QADJ,OAGI,iBAAiB,cAAc,KAAAd,CAAjB,C;;K;IAGR,4C;MACI,IAAI,kEAAgC,mBAAhC,CAAJ,C;QAAA,OACI,gBAAgB,cAAc,MAAd,CAAhB,C;;QADJ,OAGI,iBAAwB,WAAP,MAAO,yBAAsB,UAAAtB,CAAxB,C;;K;IwMI3CR,8B;MAEGD,QAAM,SAAN,M;aAC5C,a;UAD4C,OACHB,I;aAC5B,c;UAF4C,OAef,I;aAC7B,c;UAH4C,OAGf,I;aAC7B,S;UAJ4C,OAIpB,G;aACxB,S;UAL4C,OAKpB,G;aACxB,O;UAN4C,OAMtB,G;aACtB,M;UAP4C,OAovB,G;;UpMuEwB,MAAM,6BAA8B,CoMtEnE,mBAAgB,SpMsEmD,YAA9B,C;;K;IoMnEvD,4C;MACwE,QAAM,SAAN,C;aACpE,I;UADoE,6C;aAEpE,I;UAFoE,8C;aAGpE,I;UAHoE,8C;aAIpE,G;UAJoE,yC;aAKpE,G;UALoE,yC;aAMPe,G;UANoE,uC;aAOpE,G;UAPoE,sC;;UAQ5D,MAAM,gCAAyB,uCAAoC,SAA7D,C;;K;IAGIB,yD;MAGQ,KAAC,eAAD,C;QAEQ,IADE,OACF,Q;UAHZ,sC;;UAIoB,MAAM,gCAAyB,4EAAqD,OAARd,CAAzB,C;;QAIIB,QAAM,OOAN,C;eACI,E;YATZ,uC;eAUy,E;YAVZ,yC;eAWY,E;YAXZ,yC;;YAYoB,MAAM,gCAAyB,yDAAkC,OOAIC,CAAzB,C;;K;IC5F9B,4B;K;;MC4BI,kC;;IAXA,gC;MAAA,oC;MAM0B,2BAAC,iC;K;8CACpC,Y;MAAkC,OOAA,iCAAoB,W;K;6CADhC,Y;MAAA,yC;K;;IAN1B,4C;MAAA,2C;QAAA,0B;;MAAA,oC;K;IAWA,gC;MAAA,oC;K;;IAAA,4C;MAAA,2C;QAAA,0B;;MAAA,oC;K;;IAKJ,oB;K;qCaCl,oB;MAK8D,4BAAiB,IAAjB,EAAB,QAAB,C;K;sCAE9D,oB;MAK+D,wBAAM,QAAD,aAAL,C;K;sCAG/D,Y;MAMqC,QAAC,iBAAa,a;K;yCAEnD,Y;MAMwC,OOAA,iBAAa,a;K;;4EAIzD,yB;MAAA,4C;MAAA,mC;QAQuE,MAAM,WAAM,0BAAAN,C;O;KAR7E,C;mFAUA,yB;MAAA,4C;MAAA,mC;QAQsE,MAAM,WAAM,0BAAAN,C;O;KAR5E,C;IAY8B,4C;MAAiD,mB;MAAhD,gB;MAAoB,4B;K;4CAC/C,Y;MAAsC,OOAA,SAAK,aAAL,cAAoB,eAApB,C;K;6CAEtC,oB;MAAkD,4BAAiB,SAAjB,EAAB,4BAAa,QAAB,CAAvB,C;K;;IChGV,sC;MAAC,gB;K;IAOf,4E;MAA8G,mB;MAA7G,4B;MAA6B,8B;MAAgD,sB;K;+DACpG,Y;MAAsC,OOAGC,aAA/B,iBAAW,OAAX,UAAoB,gBAApB,CAA+B,EAAW,iBAAW,KAAtB,CAAhC,cAA8D,aAA9D,C;K;gEACtC,oB;MAAkD,+CAAa,gBAAb,EAAB,iBAAxB,EAAC,0BAAS,QAAT,CAAP,C;K;;+CAGtD,Y;MAAmC,+CAAa,WAAB,EAAB,IAArB,EAAB,gCAAS,KAAP,C;K;;IAUO,wC;MAAC,gB;K;IAOf,gF;MAAkH,mB;MAAjH,4B;MAA+B,8B;MAAkD,sB;K;mEAC1G,Y;MAAsC,OOAGC,aAA/B,iBAAW,OAAX,GAAoB,gBAAW,EAAW,iBAAW,KAAtB,CAAhC,cAA8D,aAA9D,C;K;oEACtC,oB;MAAkD,mDAAe,gBAAf,EAA0B,iBAA1B,EAAsC,0BAAS,QAAT,CAAT,C;K;;iDAGtD,Y;MAAmC,mDAAe,WAaf,EAAB,IAAvB,EAAB,gCAAS,KAAT,C;K;;IAGvC,0B;MAGB8B,yE;MAC1B,mB;K;oCAEA,Y;MAA4B,qB;K;iDAE5B,oB;MAWc,Q;MADV,gBAAgB,QAAS,gBAAO,SAAP,C;MACf,IAAI,gDAA+B,4CAAnC,C;QAEN,iBAAiB,mBAAU,SAAV,C;QACjB,IAAI,mBAAy,SAAZ,gBAAyB,CAAzB,IAA8B,mBAAy,UAAZ,eAAyB,CAA3D,C;UAA8D,gBAAAS,QAAT,C;QAC9D,iB;;QAEA,YAAy,QAAS,kBAAS,SAAT,C;QAErB,mBAAiB,4BAAU,K;QAC3B,IAAI,sDAA+B,kDAAAnC,C;UAAgE,gBAAAS,QAAT,C;QACrD,8BAAX,YAAW,C;;MAVf,qB;K;0CACJ,oB;MACI,MAAM,6BAAAsB,iDAA+C,cAA/C,qCAA0E,QAA1E,MAAtB,C;K;;qFC7Fd,yB;MAAA,yC;MAAA,wB;QA2BI,WAAW,8B;QAhB6B,KAIbxC,E;QAJBA,OAKBO,IAAK,a;O;KA7BhB,C;uFAeA,4B;MAYI,WAAW,mB;MACX,O;MACA,OOAO,IAAK,a;K;IAYe,qC;MAAC,kB;MAAc,wB;K;;sCAR9C,Y;MAQgC,iB;K;sCARhC,Y;MAQ8C,oB;K;wCAR9C,2B;MAAA,sBAQgC,qCARhC,EAQ8C,8CAR9C,C;K;oCAAA,Y;MAAA,OAQgC,iDARhC,IAQ8C,8CAR9C,O;K;oCAAA,Y;MAAA,c;MAQgC,sD;MAAc,yD;MAR9C,a;K;kCAAA,iB;MAAA,4IAQgC,sCARhC,IAQ8C,4CAR9C,I;K;iGAUA,yB;MAAA,yC;MAGBA,8C;MAhBA,wB;QA6BI,WAAW,8B;QACX,aAjB8C,KAIbJc,E;QAJBb,OAKBO,oBAAW,MAAX,EAAMB,IAAK,aAAxB,C;O;KA/BX,C;mGAgBA,yB;MAAA,8C;MAAA,mC;QAaI,WAAW,mB;QACX,aAAa,O;QACb,OOAO,oBAAW,MAAX,EAAMB,IAAK,aAAx

B,C;O;KafX,C;IzJZA,2E;MASI,sC;MAAA,4C;K;IATJ,mGAWY,Y;MAAQ,2B;KAXpB,E;IAAA,4DAaQ,kB;MA
CI,wBAAW,MAAX,C;K;IAdZ,wF;IOJewC,sC;MACpC,0B;K;;IAGJ,kC;MAUI,OAA2C,CAA3C,2BAA6B,uBAA7
B,EAAoC,KAApC,CAA2C,e;K;IAE/C,8B;K;kDAuBI,4B;MASI,MAAM,qCAA8B,8CAA9B,C;K;;;IAa4B,8C;MA
GtC,6B;MAEmD,UAMX,M;MAPxC,kBACmD,mE;MAEnD,eAC0B,K;MAE1B,cACwC,kE;MAExC,gBACmC,g
B;K;iGAG/B,Y;MAAQ,0C;K;0DAEZ,kB;MACI,cAAY,I;MACZ,gBAAC,M;K;IAGsE,iG;MAAA,uB;QAExE,Q;Q
AAZ,qCAAY,8D;QACZ,sCAAA,a;QAFb,OAGA,yB;O;K;2DAJJ,+B;MAAKD,OAAcC,wDAAtC,c;K;IAOyE,uH;M
AAA,uB;QAEExG,Q;QAAf,iBAAe,8F;QACf,eAAK,2B;QAA6B,mC;QtMjGtB,gBAAT,Q;QsMsG0D,kB;QAJzD,sB
AAsB,SAAK,W;QAC3B,IAAI,eAAa,eAAjB,C;UAEL,iC;UACA,mBAAY,oCAAwB,eAAxB,EAAyC,kEAAzC,C;;
UAGZ,mBAAY,kE;;QAEhB,oBAAa,e;QAZjB,OAcA,yB;O;K;6DAfJ,0C;MAAqF,OAAcC,qEAAiC,c;K;IAqBzB,
mI;MAAA,qB;QACxD,yCAAgB,uB;QAGhB,qCAAY,Y;QACZ,uCAAC,E;QACIB,W;O;K;iEATA,iC;MAGwB,w
CAAA,mCAAb,EAAoC,kFAApC,C;K;mDAQxB,Y;MAMuB,UADC,MACD,EAIH,MAJG,EAaK,M;MAjBxB,OA
AO,IAAP,C;QAEI,aAAa,IAAK,S;QACF,SAAL,IAAK,O;QAAL,mB;UACyB,gBAArB,0D;U3JxBhB,U;UADP,yB
;U2JyBe,O3JxBR,sF;;Q2JuBC,WAAW,M;QAGX,IAAI,mDAAoB,MAApB,QAAJ,C;;YAIiB,SAAT,ezJxJV,CyJwJ
uD,IzJxJvD,EyJwJ6D,YzJxJ7D,EyJwJoE,IzJxJpE,EAA8C,KAA9C,C;;YyJyJQ,gC;cACE,I1JzJhB,oBDgDQ,WAA
O,c2JyG0B,C3JzG1B,CAAP,CChDR,C;c0J0JgB,Q;;cALI,O;;UAAR,c;UAQA,IAAI,MAAM,yBAAV,C;YACI,I1Jv
KhB,oBDgDQ,W2JuHoB,0E3JvHpB,CChDR,C;;U0J0KY,gBAAC,gB;UACd,IAAK,oBAAW,MAAX,C;;K;;0EC1
MrB,4B;MAoKI,QAhKK,SAGKG,GAhKoB,KAgKpB,I;MACR,IAAI,CAjKC,SAiKD,GAjKwB,KAiKxB,IAAiB,C
AAjB,IAAsB,eAjKE,KAiKF,MAjKrB,SAiKL,C;QAA6C,a;;MAjK7C,OakKO,C;K;kEhKX,yB;MAAA,0B;MAA
A,mC;QA2KI,QAnKK,SAmKG,GAnKe,K;QAAvB,OAAgC,OAoKzB,KApKgB,KAOXX,GAAW,CAAC,CAAC,I
ApKF,KAOKC,KAAmB,KAAK,CAAC,CAAD,IAAL,CAAnB,CAAD,KAAkC,EAAID,KApKyB,C;O;KARpC,C;4
EAUA,4B;MAoJI,QAhJK,SAGJG,GAhJoB,KAgJpB,I;MACR,IAAI,CAjJC,SAiJD,GAjJwB,KAiJxB,IAAiB,CAAj
B,IAAsB,eAjJE,KAiJF,MAjJrB,SAiJL,C;QAA6C,a;;MAj7C,OakJO,C;K;kEhJX,yB;MAAA,4B;MAAA,mC;QA
2JI,QAnJK,SAmJG,GAnJe,K;QAAvB,OAAgC,QAoJzB,KApJgB,KAOJX,GAAW,CAAC,CAAC,IApJF,KAOJC,K
AAmB,KAAK,CAAC,CAAD,IAAL,CAAnB,CAAD,KAAkC,EAAID,KApJyB,C;O;KARpC,C;4EAUA,4B;MAoII,
QAhIK,SAGIG,GAhIc,KAgId,I;MACR,IAAI,CAjIC,SAiID,GAjIkB,KAiIB,IAAiB,CAAjB,IAAsB,eAjIJ,KAiII,M
AjIrB,SAiIL,C;QAA6C,a;;MAjI7C,OakIO,C;K;kEhIX,4B;MA2II,QAnIK,SAmIG,GAnIS,K;MAAjB,OAoIO,KA
pIU,KAOIL,GAAW,CAAC,CAAC,IApIR,KAOIO,KAAmB,KAAK,CAAC,CAAD,IAAL,CAAnB,CAAD,KAAkC,
EAAID,K;K;4EAIIX,yB;MAqMA,0B;MArMA,mC;QAIkB,kBAAT,oBAAL,SAAK,C;QAqML,QAAQ,gBArMe,K
AqMf,C;QACR,IAAI,gBAtMmB,KAsMnB,eAAiB,CAAjB,IAAsB,mBAtMH,KAsMG,GAAa,WAAb,CAA1B,C;U
AA6C,W;;QAtM7C,OAuMO,C;O;KA3MX,C;kEAMA,4B;MAGNI,QAxMK,oBAAL,SAAK,CAwMG,QAxMU,K
AwMV,C;MAxMR,OAyMO,MAzMW,KAyMN,KAAa,MAzMP,KAyMO,CAAD,KAAmB,KAAM,CAAD,aAAL,
CAAnB,CAAD,YAAkC,EAAIC,CAAX,CAAL,C;K;4EAvMX,4B;MAoGI,QAhGK,SAGGG,GAhGoB,KAgGpB,I;
MACR,IAAI,CAjGC,SAiGD,GAjGwB,KAiGxB,IAAiB,CAAjB,IAAsB,eAjGE,KAiGF,MAjGrB,SAiGL,C;QAA6
C,a;;MAjG7C,OakGO,C;K;kEhGX,yB;MAAA,0B;MAAA,mC;QA2GI,QAnGK,SAmGG,GAnGe,K;QAAvB,O
AAgC,OAoGzB,KApGgB,KAOGX,GAAW,CAAC,CAAC,IApGF,KAOGC,KAAmB,KAAK,CAAC,CAAD,IAAL,
CAAnB,CAAD,KAAkC,EAAID,KApGyB,C;O;KARpC,C;4EAUA,4B;MAoFI,QAhFK,SAGFG,GAhFoB,KAgFpB
,I;MACR,IAAI,CAjFC,SAiFD,GAjFwB,KAiFxB,IAAiB,CAAjB,IAAsB,eAjFE,KAiFF,MAjFrB,SAiFL,C;QAA6C,
a;;MAjF7C,OakFO,C;K;kEhFX,yB;MAAA,4B;MAAA,mC;QA2FI,QAnFK,SAmFG,GAnFe,K;QAAvB,OAAgC
,QAoFzB,KApFgB,KAOFX,GAAW,CAAC,CAAC,IApFF,KAOFC,KAAmB,KAAK,CAAC,CAAD,IAAL,CAAnB,
CAAD,KAAkC,EAAID,KApFyB,C;O;KARpC,C;4EAUA,4B;MAoEI,QAhEK,SAGEG,GAhEc,KAgEd,I;MACR,I
AAI,CAjEC,SAiED,GAjEkB,KAiEIB,IAAiB,CAAjB,IAAsB,eAjEJ,KAiEI,MAjErB,SAiEL,C;QAA6C,a;;MAjE7C
,OAKEO,C;K;kEhEX,4B;MA2EI,QAnEK,SAmEG,GAnES,K;MAAjB,OAoEO,KApEU,KAOEL,GAAW,CAAC,
CAAC,IApER,KAOEO,KAAmB,KAAK,CAAC,CAAD,IAAL,CAAnB,CAAD,KAAkC,EAAID,K;K;4EAIEX,yB;
MAqIA,0B;MArIA,mC;QAIkB,kBAAT,oBAAL,SAAK,C;QAqIL,QAAQ,gBArIe,KaqIf,C;QACR,IAAI,gBAtImB
,KAsInB,eAAiB,CAAjB,IAAsB,mBAtIH,KAsIG,GAAa,WAAb,CAA1B,C;UAA6C,W;;QAtI7C,OAuIO,C;O;KA3I
X,C;kEAMA,4B;MAGJI,QAxIK,oBAAL,SAAK,CAwIG,QAxIU,KAwIV,C;MAxIR,OAyIO,MAzIW,KAyIN,KAA
a,MAZIP,KAyIO,CAAD,KAAmB,KAAM,CAAD,aAAL,CAAnB,CAAD,YAAkC,EAAIC,CAAX,CAAL,C;K;2EA
vIX,4B;MAoCI,QAhCA,SAGCQ,GAhCY,KAGCZ,I;MACR,IAAI,CAjCJ,SAiCI,GAjCgB,KAiChB,IAAiB,CAAjB,

IAAsB,eAjCN,KAiCM,MAjC1B,SAiCA,C;QAA6C,a;;MAjC7C,OAKCO,C;K;IEAhCX,yB;MAAA,0B;MAAA,mC
;QA2CI,QAnCA,SAmCQ,GAnCO,K;QAaf,OAAwB,OAoCjB,KApCQ,KAOCH,GAAW,CAAC,CAAC,IAPCV,K
AoCS,KAAmB,KAAK,CAAC,CAAD,IAAL,CAAnB,CAAD,KAAkC,EAAID,KAPCiB,C;O;KAR5B,C;4EAUA,4
B;MAoBI,QAhBA,SAGBQ,GAhBY,KAgBZ,I;MACR,IAAI,CAjBJ,SAiBI,GAjBgB,KAiBhB,IAAiB,CAAjB,IAAs
B,eAjBN,KAiBM,MAjB1B,SAiBA,C;QAA6C,a;;MAjB7C,OAKBO,C;K;mEAhBX,yB;MAAA,4B;MAAA,mC;QA
2BI,QAnBA,SAmBQ,GAnBO,K;QAaf,OAAwB,QAoBjB,KApBQ,KAOBH,GAAW,CAAC,CAAC,IAPBV,KAOB
S,KAAmB,KAAK,CAAC,CAAD,IAAL,CAAnB,CAAD,KAAkC,EAAID,KAPBiB,C;O;KAR5B,C;4EAUA,4B;M
AII,QAAQ,YAAO,KAAP,I;MACR,IAAI,aAAS,KAAT,IAAiB,CAAjB,IAAsB,eAAI,KAAJ,MAAa,SAAvC,C;QA
A6C,a;;MAC7C,OAAO,C;K;mEAGX,4B;MAQI,QAAQ,YAAO,K;MACf,OAAO,KAAK,QAAW,CAAC,CAAC,I
AAM,KAAP,KAAmB,KAAK,CAAC,CAAD,IAAL,CAAnB,CAAD,KAAkC,EAAID,K;K;4EAGX,yB;MAGeA,0B
;MAhEA,mC;QAIkB,kBAAT,oBAAL,SAAK,C;QAgEL,QAAQ,gBAhEe,KAgEf,C;QACR,IAAI,gBAjEmB,KAiEn
B,eAAiB,CAAjB,IAAsB,mBAjEH,KAiEG,GAAa,WAAb,CAA1B,C;UAA6C,W;;QAJE7C,OAKEO,C;O;KATEx,C;
kEAMA,4B;MA2EI,QAnEK,oBAAL,SAAK,CAMeG,QAnEU,KAmEV,C;MAnER,OAoEO,MApEW,KAOEN,KA
Aa,MApEP,KAOEO,CAAD,KAAmB,KAAm,CAAD,aAAL,CAAnB,CAAD,YAAkC,EAAIC,CAAX,CAAL,C;K;6
EAIEX,yB;MAGDA,0B;MAhDA,mC;QAIS,cAAe,oBAAN,KAAM,C;QAgDpB,QAhDA,SAGDQ,KAAO,OAAP,C;
QACR,IAjDA,SAiDI,KAAS,OAAT,eAAiB,CAAjB,IAAsB,mBAAI,OAaj,GAjD1B,SAiD0B,CAA1B,C;UAA6C,
W;;QAJD7C,OAKDO,C;O;KATDX,C;mEAMA,yB;MAAA,0B;MAAA,mC;QAQS,cAAU,oBAAN,KAAM,C;QAm
Df,QAnDA,SAmDQ,QAAO,OAAP,C;QAnDR,OAAYB,OAoDIB,MAAK,YAAa,MAAM,OAAN,CAAD,KAAmB,
KAAM,CAAD,aAAL,CAAnB,CAAD,YAAkC,EAAIC,CAAX,CAAL,CAPdKB,S;O;KAR7B,C;6EAUA,yB;MAGC
A,0B;MAhCA,mC;QAIS,cAAe,oBAAN,KAAM,C;QAgCpB,QAhCA,SAGCQ,KAAO,OAAP,C;QACR,IAjCA,SAi
CI,KAAS,OAAT,eAAiB,CAAjB,IAAsB,mBAAI,OAaj,GAjC1B,SAiC0B,CAA1B,C;UAA6C,W;;QAJC7C,OAKC
O,C;O;KATCX,C;mEAMA,yB;MAAA,4B;MAAA,mC;QAQS,cAAU,oBAAN,KAAM,C;QAmCf,QAnCA,SAmCQ
,QAAO,OAAP,C;QAnCR,OAAYB,QAoCIB,MAAK,YAAa,MAAM,OAAN,CAAD,KAAmB,KAAm,CAAD,aAA
L,CAAnB,CAAD,YAAkC,EAAIC,CAAX,CAAL,CAPcKB,S;O;KAR7B,C;6EAUA,yB;MAGBA,0B;MAhBA,mC;
QAIS,cAAe,oBAAN,KAAM,C;QAgBpB,QAhBA,SAGBQ,KAAO,OAAP,C;QACR,IAjBA,SAiBI,KAAS,OAAT,e
AAiB,CAAjB,IAAsB,mBAAI,OAaj,GAjB1B,SAiB0B,CAA1B,C;UAA6C,W;;QAJB7C,OAKBO,C;O;KATBX,C;m
EAMA,4B;MAQS,cAAU,oBAAN,KAAM,C;MAMbF,QAnBA,SAmBQ,QAAO,OAAP,C;MAnBR,OAoBO,MAAK
,YAAa,MAAM,OAAN,CAAD,KAAmB,KAAm,CAAD,aAAL,CAAnB,CAAD,YAAkC,EAAIC,CAAX,CAAL,CA
pBkB,Q;K;6EAE7B,yB;MAAA,0B;MAAA,mC;QAI,QAAQ,cAAO,KAAP,C;QACR,IAAI,cAAS,KAAT,eAAiB,
CAAjB,IAAsB,mBAAI,KAAJ,GAAa,SAAb,CAA1B,C;UAA6C,W;;QAC7C,OAAO,C;O;KANX,C;mEASA,4B;M
AQI,QAAQ,iBAAO,KAAP,C;MACR,OAAO,MAAK,UAAa,MAAM,KAAm,CAAD,KAAmB,KAAm,CAAD,aAA
L,CAAnB,CAAD,YAAkC,EAAIC,CAAX,CAAL,C;K;kEAGX,yB;MpGiQ2C,iB;MoGjqB3C,mC;QAUI,QAAQ,Y
AAO,K;QACJ,iBAAS,G;QAAT,S;UAAAsB,OpGspBc,MAAiC,MoGtpB/C,CpGspB+C,CoGtpB/C,KpGspBc,MAAi
C,MoGtpBrC,KpGspBqC,C;;QoGtpBhF,OAAO,OAAGD,IAAI,KAApD,GAA+D,C;O;KAX1E,C;mEAca,yB;MpG
0I6C,iB;MoG1I7C,mC;QAKCI,QAxBK,SAwBG,GAXBY,K;QAYBT,iBAAK,G;QAAL,S;UAAy,OpGuG0B,MAA
W,MoGvGrC,CpGuGqC,CoGvGrC,KpGuG0B,MAAW,MoGhIxC,KpGgIwC,C;;QoGhI5D,OAYBO,OAAsC,IAzBz
B,KAYBb,GAAqD,C;O;KANChE,C;mEAYA,yB;MpG8H6C,iB;MoG9H7C,mC;QASBI,QAZA,SAYQ,GAZO,K;Q
AaJ,iBAAK,G;QAAL,S;UAAy,OpGuG0B,MAAW,MoGvGrC,CpGuGqC,CoGvGrC,KpGuG0B,MAAW,MoGpH7
C,KpGoH6C,C;;QoGpH5D,OAaO,OAAsC,IAb9B,KAAr,GAAqD,C;O;KAvBhE,C;mEAYA,yB;MpGkH6C,iB;Mo
GIH7C,mC;QAUI,QAAQ,YAAO,K;QACJ,iBAAK,G;QAAL,S;UAAy,OpGuG0B,MAAW,MoGvGrC,CpGuGqC,
CoGvGrC,KpGuG0B,MAAW,MoGvG3B,KpGuG2B,C;;QoGvG5D,OAAO,OAAsC,IAAI,KAA1C,GAAqD,C;O;K
AXhE,C;4ECnTA,yB;MAAA,8B;MAAA,4B;QAOyC,Q;QAAA,gFAAoB,C;O;KAP7D,C;ICM0B,4C;MA+CtB,qC;
MA/CuB,kB;MAAgB,kB;MAAgB,kB;MAMvD,iBAAsB,iBAAU,UAAV,EAAiB,UAAjB,EAAwB,UAAxB,C;K;0
CAETB,+B;M5MWA,IAAI,E4MViB,CAAT,sBAAY,GAAZ,KAA4C,CAAT,sBAAY,GAA/C,MAA+E,CAAT,sBA
AY,GAAIF,C5MUR,CAAJ,C;QACI,c4MVI,2E;Q5MWJ,MAAM,gCAAyB,OAAQ,WAAjC,C;;M4MTN,OAAO,C
AAA,KAAm,IAAI,EAAV,KAAgB,KAAm,IAAI,CAA1B,IAA+B,KAA/B,I;K;uCAGX,Y;MAGkC,OAAE,UAAF,
oBAAS,UAAT,SAAGB,U;K;qCAEID,iB;MAEWB,gB;MADpB,IAAI,SAAS,KAAb,C;QAAoB,OAAO,I;MACP,iE;
MAAD,mB;QAA6B,OAAO,K;;MAAvD,mBAAmB,M;MACnB,OAAO,IAAK,UAAAL,KAAgB,YAAa,U;K;uCAGX

C, Y;MAA+B,qB;K;8CAE/B,iB;MAAoD,wBAAU,KAAM,UAAhB,I;K;gDAEpD,wB;MAKI,OAAA,IAAK,MAAL ,GAAa,KAAb,KAAuB,IAAK,MAAL,KAAC,KAAd,IACf,IAAK,MAAL,IAAc,KADtB,C;K;gDAGJ,+B;MAKI,OA AA,IAAK,MAAL,GAAa,KAAb,KAAuB,IAAK,MAAL,KAAC,KAAd,KACd,IAAK,MAAL,GAAa,KAAb,KAAsB, IAAK,MAAL,KAAC,KAAd,IACf,IAAK,MAAL,IAAc,KADrB,CADc,CAAvB,C;K;IAIJ,mC;MAAA,uC;MACI,2B AIuC,G;MAEvC,eAIoC,uCAA0B,M;K;;;IAXIE,+C;MAAA,8C;QAAA,6B;;MAAA,uC;K;;IA9CA,iD;MAAA,uD; MAG6C,0BAAK,KAAL,EAAY,KAAZ,EAAMb,CAAnB,C;MAH7C,Y;K;IA6DJ,qC;MAAA,yC;K;8CAEI,Y;MA C2B,yBAAC,CAAd,EAAiB,CAAJB,EAAoB,EAAPB,C;K;;;IAH/B,iD;MAAA,gD;QAAA,+B;;MAAA,yC;K;4FCx DI,yB;MAAA,2D;MAAA,4B;QAAQ,MAAM,6BAAoB,6BAAPB,C;O;KAAd,C;;;ICSJ,uB;MAG2C,+BAAoB,KA APB,C;K;4EAE3C,wC;MAO4F,sB;K;IAE5F,6C;MAAA,e;MAAA,iB;MAAA,uB;K;IAAA,2C;MAAA,8C;O;MAK I,wF;MAKA,sF;MAMA,wE;K;;;IAXA,yD;MAAA,iC;MAAA,iD;K;;IAKA,wD;MAAA,iC;MAAA,gD;K;;IAMA,iD ;MAAA,iC;MAAA,yC;K;;IAhBJ,uC;MAAA,iJ;K;;IAAA,4C;MAAA,a;aAAA,c;UAAA,sD;aAAA,a;UAAA,qD;aA AA,M;UAAA,8C;;UAAA,gE;;K;;IAyBA,+B;MAAA,mC;K;;;IAAA,2C;MAAA,0C;QAAA,yB;;MAAA,mC;K;IAG oC,qC;MACHc,qBAAsC,W;MACtC,gBAA2B,iC;K;uFAGvB,Y;MAMW,Q;MALP,IAAL,kBAAW,iCAAf,C;QACI ,gBAAS,mC;QACT,qBAAC,I;;MAGIB,OAAO,gF;K;6CAGf,Y;MAAwC,yBAAW,iC;K;wCAEnD,Y;MAAkC,OA AI,oBAAJ,GAA2B,SAAN,UAAM,CAA3B,GAA2C,iC;K;8CAE7E,Y;MAAkC,+BAAoB,UAApB,C;K;;IAGG,oC; MAAC,4B;K;wEAAA,Y;MAAA,2B;K;kDAEtC,Y;MAAwC,W;K;6CAExC,Y;MAAkC,OAAM,SAAN,UAAM,C; K;;oFC2C5C,yB;MAAA,gD;MAAA,4B;QAM6C,OAAmB,aAAIB,YAAY,GAAM,C;O;KANhE,C;oGAQA,yB;Mx G7FA,iB;MwG6FA,4B;QAMqD,OxG7FM,MAAO,OwG6FZ,YAAY,GxG7FA,CwG6Fb,GAA6C,EAA7C,I;O;KA NrD,C;sGAQA,yB;MAAA,kE;MAAA,4B;QAMsD,OAAmB,sBAAIB,YAAW,GAAO,C;O;KANzE,C;8FAQA,yB; MAAA,0D;MAAA,0B;MAAA,4B;QAOMD,OAAuC,OAApB,kBAAIB,YAAY,GAAM,CAAoB,C;O;KAP1F,C;4F ASA,yB;MAAA,wD;MAAA,0B;MAAA,4B;QAOKD,OAA2B,OAAAnB,iBAAR,SAAQ,CAAmB,C;O;KAP7E,C;IA UA,2C;MAaI,OAA+E,OAA9E,SAAQ,KAAI,WAAa,CAAJB,CAAR,GAakD,CAAIB,YAAY,GAAM,MAAK,CA AL,IAAU,WAAa,CAAvB,CAA4B,C;K;IAEnF,4C;MAaI,OAA+E,OAA9E,SAAQ,IAAI,CAAJ,IAAS,WAAa,CAAT B,CAAR,GAAwD,CAAIB,YAAY,GAAM,OAAK,WAAa,CAAIB,CAAsB,C;K;oFAEnF,yB;MAAA,gD;MAAA,4 B;QAM8C,OAAqB,aAApB,YAAY,KAAQ,C;O;KANnE,C;oGAQA,yB;MxGtKA,iB;MwGsKA,4B;QAOI,OxGvK uD,MAAO,OwGuK7D,YAAY,KxGvKiD,CwGuK9D,GAA+C,EAA/C,I;O;KAPJ,C;sGASA,yB;MAAA,kE;MAAA ,4B;QAMuD,OAAqB,sBAAPB,YAAW,KAAS,C;O;KAN5E,C;8FAQA,yB;MAAA,0D;MAAA,4B;MAAA,4B;QA OqD,OAAyC,QAApB,kBAAPB,YAAY,KAAQ,CAAoB,C;O;KAP9F,C;4FASA,yB;MAAA,wD;MAAA,4B;MAA A,4B;QAOoD,OAA2B,QAAAnB,iBAAR,SAAQ,CAAmB,C;O;KAP/E,C;IAUA,2C;MAaI,OAAoF,QAAAnF,SAAQ, KAAI,WAAa,EAAjB,CAAR,GAAqD,CAAPB,YAAY,KAAQ,MAAK,EAAL,IAAW,WAAa,EAAxB,CAA8B,C;K; IAExF,4C;MAaI,OAAoF,QAAAnF,SAAQ,IAAI,EAAJ,IAAU,WAAa,EAAvB,CAAR,GAA4D,CAAPB,YAAY,KA AQ,OAAK,WAAa,EAAIB,CAAuB,C;K;0E/MIRxY,yB;MAaA,kF;MAbA,wB;QAUbI,IAAI,CAbI,KAAr,C;UACI,c Ada,qB;UAeb,MAAM,8BAAyB,OAAQ,WAAjC,C;;O;KAZbD,C;0EAaA,yB;MAAA,kF;MAAA,qC;QAUI,IAAI,C AAC,KAAL,C;UACI,cAAc,a;UACd,MAAM,8BAAyB,OAAQ,WAAjC,C;;O;KAZd,C;sFAgBA,yB;MAWA,kF;M AXA,wB;QAQW,yB;QAEp,IAfsB,KAelB,QAAJ,C;UACI,cAhB2B,0B;UAIb3B,MAAM,8BAAyB,OAAQ,WAAjC ,C;;UAEN,wBAnBkB,K;;QAATB,4B;O;KARJ,C;wFAWA,yB;MAAA,kF;MAAA,qC;QAYI,IAAI,aAAJ,C;UACI,c AAc,a;UACd,MAAM,8BAAyB,OAAQ,WAAjC,C;;UAEN,OAAO,K;;O;KAhBf,C;oEAoBA,yB;MAaA,4E;MAbA, wB;QAUbI,IAAI,CAbE,KAAr,C;UACI,cAdW,e;UAEX,MAAM,2BAAsB,OAAQ,WAA9B,C;;O;KAZbD,C;sEAaA ,yB;MAAA,4E;MAAA,qC;QAUI,IAAI,CAAC,KAAL,C;UACI,cAAc,a;UACd,MAAM,2BAAsB,OAAQ,WAA9B, C;;O;KAZd,C;kFAgBA,yB;MAcA,4E;MAdA,wB;QAWW,uB;QAEp,IAfoB,KAehB,QAAJ,C;UACI,cAhByB,0B;U AIbZB,MAAM,2BAAsB,OAAQ,WAA9B,C;;UAEN,sBAnBgB,K;;QAAPB,0B;O;KAXJ,C;oFAcA,yB;MAAA,4E; MAAA,qC;QAYI,IAAI,aAAJ,C;UACI,cAAc,a;UACd,MAAM,2BAAsB,OAAQ,WAA9B,C;;UAEN,OAAO,K;;O; KAhBf,C;oEAqBA,yB;MAAA,4E;MAAA,0B;QAMiD,MAAM,2BAAsB,OAAQ,WAA9B,C;O;KANvD,C;I8CnHi C,uB;MA2D7B,8B;MA1DA,kB;K;mFAS8B,Y;MAAQ,iD;K;mFAMR,Y;MAAQ,gD;K;wFAItC,yB;MAAA,gB;M AAA,8B;MAAA,mB;QAWgB,Q;QADR,mB;UADJ,OACiB,I;;UADjB,OAeY,2E;O;KAXhB,C;uCACa,Y;MAQQ, kBADE,UACf,kB;QADJ,OACkB,UAAM,U;;QADxB,OAeY,I;K;gCAGhB,Y;MAOQ,kBADE,UACf,kB;QADJ,O ACkB,UAAM,W;;QADxB,OAeY,sBAAU,UAAV,O;K;IAKhB,4B;MAAA,gC;K;wHAKI,yB;MAAA,iC;MAAA,w B;QAOI,uBAAO,KAAP,C;O;KAPJ,C;wHASA,yB;MAAA,kD;MAAA,iC;MAAA,4B;QAOI,uBAAO,cAAc,SAAD,

CAAP,C;O;KAPJ,C;;;IAdJ,wC;MAAA,uC;QAAA,sB;;MAAA,gC;K;IAwBsB,mC;MACIB,0B;K;sCAGA,iB;MAA
4C,+CAAoB,uBAAa,KAAM,UAAAnB,C;K;wCACHE,Y;MAA+B,OAAU,SAAV,cAAU,C;K;wCACzC,Y;MAAkC,
oBAAU,cAAV,M;K;gCA/F1C,Y;MAAA,c;MAOI,sD;MAPJ,a;K;8BAAA,iB;MAAA,2IAOI,sCAPJ,G;K;IAmG
A,kC;MAOI,OAAO,mBAAQ,SAAR,C;K;IAEX,mC;MAQI,IAAI,8CAAJ,C;QAA6B,MAAM,eAAM,U;K;gFAG7C
,yB;MAAA,4B;MAAA,qB;MAx CQ,kD;MAwCR,wB;QAOW,Q;;UACI,OAIDH,WakDW,OAIDX,C;;UAmDN,gC
;YACS,OA3CH,WAAO,cA2CI,CA3CJ,CAAP,C;;YAwCD,O;;QAAP,W;O;KAPJ,C;kFAcA,yB;MAAA,4B;MAAA
,qB;MAtdQ,kD;MA sDR,mC;QAOW,Q;;UACI,OAHEH,WAgEW,gBAhEX,C;;UAiEN,gC;YACS,OAzDH,WAAO,
cAyDI,CAzDJ,CAAP,C;;YAsDD,O;;QAAP,W;O;KAPJ,C;8EAgBA,yB;MAAA,oD;MAAA,gB;MAAA,8B;MAAA
,4B;QAUW,Q;QADP,yB;QACA,OAAO,gF;O;KAVX,C;+EAaA,yB;MAAA,gB;MAAA,8B;MAAA,uC;QAegB,U
ADL,M;QAAM,gBAAGB,2B;QACzB,sB;UAAQ,yF;;UACA,mBAAU,SAAV,C;QAFZ,a;O;KAdJ,C;kFAoBA,yB;
MAAA,gB;MAAA,8B;MAAA,0C;QAUW,Q;QADP,IAAI,mBAAJ,C;UAAe,OAAO,Y;QAcTb,OAAO,gF;O;KAV
X,C;qEAaA,yB;MAAA,gB;MAAA,8B;MAAA,kD;QAIb0B,UADf,M;QAAM,gBAAGB,2B;QACzB,sB;UAAQ,m
BAAU,gFAAV,C;;UACA,mBAAU,SAAV,C;QAFZ,a;O;KAhBJ,C;mEAwBA,yB;MAAA,4B;MAAA,gB;MAAA,8
B;MAAA,uC;YAe8C,I;YADnC,M;QACH,wB;UAAa,gB;UAAO,SA7JhB,WA6JwB,UAAU,gFAAV,CA7JxB,C;;U
A8JI,oBAAO,eAAP,C;QAFZ,a;O;KAdJ,C;gFAoBA,yB;MAAA,gB;MAAA,8B;MAAA,iC;MA1GA,qB;MAtdQ,k
D;MAgKR,uC;QAWW,Q;QACH,wB;UA/GG,U;;YA+GkC,U;YA9G9B,SAhEH,gBA8KuB,UAAU,sFAAV,CA9K
vB,C;;YAiEN,gC;cACS,SAzDH,gBAAO,cAyDI,CAzDJ,CAAP,C;;cAsDD,O;;UA+GU,a;;UACL,uBAAO,eAAP,C;
QAFZ,W;O;KAXJ,C;wEAiBA,yB;MAAA,4B;MAAA,uC;QA cW,Q;QAAM,gBAAGB,2B;QACzB,sB;UAAQ,gB;;
UACO,OAnMX,WAmMmB,UAAU,SAAV,CAnMnB,C;;QAI MR,W;O;KAdJ,C;wFAoBA,yB;MA/IA,4B;MAAA,q
B;MAtdQ,kD;MAqMR,uC;QAWW,Q;QAAM,gBAAGB,2B;QACzB,sB;UAAQ,gB;;UApJL,U;;YACI,SAhEH,WA
oNkB,oBApNIB,C;;YAiEN,gC;cACS,SAzDH,WAAO,cAyDI,CAzDJ,CAAP,C;;cAsDD,O;;UAqJK,a;;QAFZ,W;O;
KAXJ,C;4EAmBA,6B;MAUI,Q;MAAA,iD;QAAYB,Y;;MACzB,OAAO,S;K;4EAGX,yB;MAAA,gB;MAAA,8B;M
AAA,oC;QAU0B,Q;QAAtB,IAAI,mBAAJ,C;UAAe,OAAO,gFAAP,C;;QACf,OAAO,S;O;KAXX,C;I3CtTgC,sC;
MAAC,uB;QAAA,UAAkB,kC;mBAA4C,O;;K;;0DAE/F,yB;MAAA,2D;MAAA,mB;QAKoC,MAAM,8B;O;KAL1
C,C;oEAOA,yB;MAAA,2D;MAAA,yB;QAMkD,MAAM,6BAAoB,sCAAmC,MAAvD,C;O;KANxD,C;gEAUA,iB
;MAUI,OAAO,O;K;kEAGX,4B;MAUI,OAAO,gB;K;oEAGX,2B;MAUI,OAAgB,MAAT,QAAS,C;K;oEAGpB,4B
;MAUI,gB;MACA,OAAO,S;K;kEAGX,4B;MAWI,MAAM,SAAN,C;MACA,OAAO,S;K;kEAGX,4B;MAUI,OAA
O,MAAM,SAAN,C;K;sEAGX,gC;MAWI,OAAW,UAAU,SAAV,CAAJ,GAAqB,SAArB,GAA+B,I;K;8EAG1C,g
C;MAWI,OAAW,CAAC,UAAU,SAAV,CAAL,GAAsB,SAAtB,GAAGC,I;K;wEAG3C,yB;MAWI,iBAAc,CAAd,
UAA sB,KAAtB,U;QACI,OAAO,KAAP,C;;K;wE6MjJR,iB;MAIKF,Y;K;ICY9C,6B;MACHc,kB;MACA,oB;K;8B
AGA,Y;MAGyC,aAAG,UAAH,UAAW,WAAX,M;K;;gCAvB7C,Y;MAGBI,iB;K;gCAhBJ,Y;MAiBI,kB;K;kCAjB
J,yB;MAAA,gBAgBI,qCAhBJ,EAI BI,wCAjBJ,C;K;8BAAA,Y;MAAA,c;MAGBI,sD;MACA,uD;MAjBJ,a;K;4BA
AA,iB;MAAA,4IAGBI,sCAhBJ,IAiBI,wCAjBJ,I;K;IA0BA,6B;MAMoD,gBAAK,SAAL,EAAW,IAAX,C;K;IAEp
D,8B;MAI8C,iBAAO,eAAP,EAAC,gBAAd,E;K;IAiBD,sC;MACzC,kB;MACA,oB;MACA,kB;K;gCAGA,Y;MAG
yC,aAAG,UAAH,UAAW,WAAX,UAAoB,UAApB,M;K;;kCAx B7C,Y;MAGBI,iB;K;kCAhBJ,Y;MAiBI,kB;K;kC
AjBJ,Y;MAKBI,iB;K;oCAIBJ,gC;MAAA,kBAgBI,qCAhBJ,EAI BI,wCAjBJ,EAKBI,qCAIBJ,C;K;gCAAA,Y;MAA
A,c;MAGBI,sD;MACA,uD;MACA,sD;MAIBJ,a;K;8BAAA,iB;MAAA,4IAGBI,sCAhBJ,IAiBI,wCAjBJ,IAKBI,sCA
IBJ,I;K;IA2BA,8B;MAImD,iBAAO,eAAP,EAAC,gBAAd,EAA sB,eAAtB,E;K;I7NIE1B,qB;MAErB,6B;MAFwD,g
B;K;IAExD,2B;MAAA,+B;MACI,iBAGoC,UAAM,CAAN,C;MAEpC,iBAGoC,UAAM,MAAN,C;MAEpC,kBAG
mC,C;MAEnC,iBAGkC,C;K;;IANbtC,uC;MAAA,sC;QAAA,qB;;MAAA,+B;K;kGAsBA,iB;MAOmE,OAAa,0BA
2O1C,SAAL,GAAiB,GA3O8B,EAAU,KA2OpD,KAAL,GAAiB,GA3O8B,C;K;sGAehf,iB;MAM2D,OAAa,0BA
mOIC,SAAL,GAAiB,GAnOsB,EAAU,KEoO5C,KAAL,GAAiB,KFpOsB,C;K;sGAExE,yB;MA0PA,6B;MC3PA,8
C;MDCA,wB;QAMyD,OCAS,YAAiB,CD6PhD,cAAU,SAAL,GAAiB,GAAtB,CC7PgD,MAAjB,EDAe,KCAc,KA
A7B,C;O;KDNIE,C;sGAQA,yB;MA4PA,WAS6D,wB;MAT7D,+B;MiB7PA,gD;MjBCA,wB;QAM0D,OiBAS,aA
AkB,CjB+PhD,eAAW,oBAAL,SAAK,CAAL,UAA N,CiB/PgD,MAAIB,EjBAGB,KiBAc,KAA9B,C;O;KjBNnE,C;
4FAQA,yB;MA0OA,6B;MA1OA,wB;QAEsD,OCMD,cAAU,CD2O5B,cAAU,SAAL,GAAiB,GAAtB,CC3O4B,M
AAK,GAAW,CD2O5C,cAjPsC,KAiP5B,KAAL,GAAiB,GAAtB,CC3O4C,MAAX,IAAf,C;O;KDRrD,C;4FAGA,y
B;MAuOA,6B;MAvOA,wB;QAEuD,OCGF,cAAU,CD2O5B,cAAU,SAAL,GAAiB,GAAtB,CC3O4B,MAAK,GA

AW,CC4O5C,cF/OuC,KE+O7B,KAAL,GAAiB,KAAtB,CD5O4C,MAAX,IAAf,C;O;KDLrD,C;4FAGA,yB;MAoO
A,6B;MAPoA,wB;QAEqD,OCAA,cAAU,CD2O5B,cAAU,SAAL,GAAiB,GAAtB,CC3O4B,MAAK,GDAI,KCAO
,KAAX,IAAf,C;O;KDFrD,C;4FAGA,yB;MA2OA,WAS6D,wB;MAT7D,+B;MA3OA,wB;QAEuD,OiBAA,eAAW,
CjBkP7B,eAAW,oBAAL,SAAK,CAAL,UAAN,CiBIP6B,MAAK,KjBAI,KiBAO,KAAX,CAAhB,C;O;KjBFvD,C;
8FAIA,yB;MA6NA,6B;MA7NA,wB;QAEuD,OCMD,cAAU,CD8N7B,cAAU,SAAL,GAAiB,GAAtB,CC9N6B,M
AAK,GAAY,CD8N9C,cApOwC,KAoO9B,KAAL,GAAiB,GAAtB,CC9N8C,MAAZ,IAAf,C;O;KDRtD,C;8FAGA,
yB;MA0NA,6B;MA1NA,wB;QAEwD,OCGF,cAAU,CD8N7B,cAAU,SAAL,GAAiB,GAAtB,CC9N6B,MAAK,G
AAY,CC+N9C,cFIOyC,KEkO/B,KAAL,GAAiB,KAAtB,CD/N8C,MAAZ,IAAf,C;O;KDLtD,C;8FAGA,yB;MAuN
A,6B;MAvNA,wB;QAEsD,OCAA,cAAU,CD8N7B,cAAU,SAAL,GAAiB,GAAtB,CC9N6B,MAAK,GDAK,KCA
O,KA AZ,IAAf,C;O;KDFtD,C;8FAGA,yB;MA8NA,WAS6D,wB;MAT7D,+B;MA9NA,wB;QAEwD,OiBAA,eAA
W,CjBqO9B,eAAW,oBAAL,SAAK,CAAL,UAAN,CiBrO8B,MAAK,UjBAK,KiBAO,KA AZ,CAAhB,C;O;KjBFx
D,C;8FAIA,yB;MAgNA,6B;MAhNA,wB;QAEuD,OCMD,cAAe,YAAL,CDiN7B,cAAU,SAAL,GAAiB,GAAtB,C
CjN6B,MAAK,EAAY,CDiN9C,cAvNwC,KAuN9B,KAAL,GAAiB,GAAtB,CCjN8C,MAAZ,CAAf,C;O;KDRtD,C
;8FAGA,yB;MA6MA,6B;MA7MA,wB;QAEwD,OCGF,cAAe,YAAL,CDiN7B,cAAU,SAAL,GAAiB,GAAtB,CCj
N6B,MAAK,EAAY,CCKN9C,cFrNyC,KEqN/B,KAAL,GAAiB,KAAtB,CDIN8C,MAAZ,CAAf,C;O;KDLtD,C;8F
AGA,yB;MA0MA,6B;MA1MA,wB;QAEsD,OCAA,cAAe,YAAL,CDiN7B,cAAU,SAAL,GAAiB,GAAtB,CCjN6B
,MAAK,EDAK,KCAO,KA AZ,CAAf,C;O;KDFtD,C;8FAGA,yB;MAiNA,WAS6D,wB;MAT7D,+B;MAjNA,wB;Q
AEwD,OiBAA,eAAW,CjBwN9B,eAAW,oBAAL,SAAK,CAAL,UAAN,CiBxN8B,MAAK,UjBAK,KiBAO,KA AZ
,CAAhB,C;O;KjBFx D,C;0FAIA,yB;MAmMA,6B;MC7LA,4C;MDNA,wB;QAEqD,OCMD,WDoMjB,cAAU,SA
L,GAAiB,GAAtB,CCpMiB,EDoMjB,cA1MoC,KA0M1B,KAAL,GAAiB,GAAtB,CCpMiB,C;O;KDRpD,C;0FAG
A,yB;MAgMA,6B;MC7LA,4C;MDHA,wB;QAEsD,OCGF,WDoMjB,cAAU,SAAL,GAAiB,GAAtB,CCpMiB,ECq
MjB,cFxMqC,KEwM3B,KAAL,GAAiB,KAAtB,CDrMiB,C;O;KDLpD,C;0FAGA,yB;MA6LA,6B;MC7LA,4C;M
DAA,wB;QAEoD,OCAA,WDoMjB,cAAU,SAAL,GAAiB,GAAtB,CCpMiB,EDAkB,KCAIB,C;O;KDFpD,C;0FA
GA,yB;MAoMA,WAS6D,wB;MAT7D,+B;MiBpMA,8C;MjBAA,wB;QAEsD,OiBAA,YjB2MjB,eAAW,oBAAL,S
AAK,CAAL,UAAN,CiB3MiB,EjBAmb,KiBAnB,C;O;KjBFtD,C;0FAIA,yB;MA sLA,6B;MCxKA,kD;MDdA,wB;
QAMqD,OCcD,cD2KjB,cAAU,SAAL,GAAiB,GAAtB,CC3KiB,ED2KjB,cAzLoC,KAyL1B,KAAL,GAAiB,GAA
tB,CC3KiB,C;O;KDPpD,C;0FAOA,yB;MA+KA,6B;MCxKA,kD;MDPA,wB;QAMsD,OCOF,cD2KjB,cAAU,SA
AL,GAAiB,GAAtB,CC3KiB,EC4KjB,cFnLqC,KEmL3B,KAAL,GAAiB,KAAtB,CD5KiB,C;O;KDbpD,C;0FAOA,
yB;MAwKA,6B;MCxKA,kD;MDAA,wB;QAMoD,OCAA,cD2KjB,cAAU,SAAL,GAAiB,GAAtB,CC3KiB,EDAK
B,KCAIB,C;O;KDNpD,C;0FAOA,yB;MA2KA,WAS6D,wB;MAT7D,+B;MiB3KA,oD;MjBAA,wB;QAMsD,OiB
AA,ejB8KjB,eAAW,oBAAL,SAAK,CAAL,UAAN,CiB9KiB,EjBAmb,KiBAnB,C;O;KjBNtD,C;oGAQA,yB;MAy
JA,6B;MC7LA,4C;MDoCA,wB;QAMiD,OCxCG,WDoMjB,cAAU,SAAL,GAAiB,GAAtB,CCpMiB,EDoMjB,cA5
JqC,KA4J3B,KAAL,GAAiB,GAAtB,CCpMiB,C;O;KDKcP D,C;oGAOA,yB;MAkJA,6B;MC7LA,4C;MD2CA,wB;
QAMkD,OC/CE,WDoMjB,cAAU,SAAL,GAAiB,GAAtB,CCpMiB,ECqMjB,cFtJsC,KEsJ5B,KAAL,GAAiB,KAA
tB,CDrMiB,C;O;KDyCpD,C;oGAOA,yB;MA2IA,6B;MC7LA,4C;MDkDA,wB;QAMgD,OCtDI,WDoMjB,cAAU,
SAAL,GAAiB,GAAtB,CCpMiB,EDsDmB,KCtDnB,C;O;KDgDpD,C;oGAOA,yB;MA8IA,WAS6D,wB;MAT7D,+
B;MiBpMA,8C;MjBsDA,wB;QAMkD,OiB1DI,YjB2MjB,eAAW,oBAAL,SAAK,CAAL,UAAN,CiB3MiB,EjB0D
oB,KiB1DpB,C;O;KjBoDtD,C;0FAQA,yB;MA4HA,6B;MCxKA,kD;MDuOJ,0B;MAAA,+B;MA3LI,wB;QAQ6C,
OA8LR,eAAW,OC5OI,cD2KjB,cAAU,SAAL,GAAiB,GAAtB,CC3KiB,ED2KjB,cA7H4B,KA6HIB,KAAL,GAAi
B,GAAtB,CC3KiB,CAkLf,KD0DW,CAAX,C;O;KATMrC,C;0FASA,yB;MAmHA,6B;MCxKA,kD;MCwOJ,4B;M
AAA,iC;MFnLI,wB;QAQ+C,OE sLR,gBAA Y,QD7OC,cD2KjB,cAAU,SAAL,GAAiB,GAAtB,CC3KiB,EC4KjB,c
FrH8B,KEqHpB,KAAL,GAAiB,KAAtB,CD5KiB,CA4Lb,KCiDY,CAAZ,C;O;KF9LvC,C;0FASA,yB;MA0GA,6B
;MCxKA,kD;MD8DA,wB;QAQ2C,OChES,cD2KjB,cAAU,SAAL,GAAiB,GAAtB,CC3KiB,EDgES,KChET,C;O;
KDwDpD,C;0FASA,yB;MA2GA,WAS6D,wB;MAT7D,+B;MiB3KA,oD;MjBgEA,wB;QAQ6C,OiBIES,ejB8KjB,e
AAW,oBAAL,SAAK,CAAL,UAAN,CiB9KiB,EjBkEU,KiBIEV,C;O;KjB0DtD,C;0EAUA,yB;MAAA,0B;MAAA,
+B;MAAA,mB;QAM0C,sBAAW,OAAL,SAAK,KAAX,C;O;KAN1C,C;0EAQA,yB;MAAA,0B;MAAA,+B;MAA
A,mB;QAM0C,sBAAW,OAAL,SAAK,KAAX,C;O;KAN1C,C;kGAQA,yB;MAAA,8C;MAuEA,6B;MAvEA,wB;
QAE8D,0BA8E3B,cAAU,SAAL,GAAiB,GAAtB,CA9E2B,EA8E3B,cA9EoD,KA8E1C,KAAL,GAAiB,GAAtB,C

A9E2B,C;O;KAF9D,C;0FAIA,yB;MAAA,+B;M6LxOJ,0B;M7LwOI,wB;QAEmD,sB6LvOgC,O7LuO1B,IAAK,K6LvOX,G7LuOoB,KAAM,K6LvOM,C7LuOhC,C;O;KAFnD,C;wFAGA,yB;MAAA,+B;M6LtOJ,0B;M7LsOI,wB;QAEkD,sB6LrO+B,O7LqOzB,IAAK,K6LrOX,G7LqOmB,KAAM,K6LrOM,C7LqO/B,C;O;KAFID,C;0FAGA,yB;MAAA,+B;M6LpOJ,0B;M7LoOI,wB;QAEmD,sB6LnOgC,O7LmO1B,IAAK,K6LnOX,G7LmOoB,KAAM,K6LnOM,C7LmOhC,C;O;KAFnD,C;0EAGA,yB;MAAA,+B;M6LlOJ,0B;M7LkOI,mB;QAEiC,sB6LjOqB,OAAP,C7LiOR,S6LjOe,C7LiOrB,C;O;KAFjC,C;gFAIA,Y;MASmC,gB;K;kFACnC,yB;M6LlOJ,4B;M7L0OI,mB;QASqC,O6LhPiD,Q7LgP5C,S6LhPY,G7LgPE,G6LhP8B,C;O;K7LuOtF,C;8EAUA,Y;MASiC,OAAC,SAAL,GAAiB,G;K;gFACID,yB;MAAA,WASqD,wB;MATrD,mB;QASmC,OAAC,oBAAL,SAAK,CAAL,U;O;KATnC,C;kFAWA,Y;MAEqC,W;K;oFACrC,yB;MAAA,iC;M6L5QJ,4B;M7L4QI,mB;QASuC,uB6LIR+C,Q7LkRnC,S6LIRG,G7LkRW,G6LIRqB,C7LkR/C,C;O;KATvC,C;gFAUA,yB;MAAA,6B;MAAA,mB;QASmC,qBAAU,SAAL,GAAiB,GAAtB,C;O;KATnC,C;kFAUA,yB;MAAA,WAS6D,wB;MAT7D,+B;MAAA,mB;QASqC,sBAAW,oBAAL,SAAK,CAAL,UAAAN,C;O;KATrC,C;kFAWA,Y;MAMqC,OApDC,SAAL,GAAiB,G;K;oFAqDID,Y;MAMuC,OA3DD,SAAL,GAAiB,G;K;+BA6DID,Y;MAAyC,OAAQ,CA7DX,SAAL,GAAiB,GA6DD,Y;K;+BA1UrD,Y;MAAA,c;MAG4D,qD;MAH5D,a;K;6BAAA,iB;MAAA,2IAG4D,oCAH5D,G;K;wEA8UA,yB;MAAA,+B;MAAA,4B;QAU0C,sBAAM,SAAN,C;O;KAV1C,C;0EAWA,yB;MAAA,0B;MAAA,+B;MAAA,4B;QAW2C,sBAAW,OAAL,SAAK,CAAX,C;O;KAX3C,C;0EAYA,yB;MAAA,0B;MAAA,+B;MAAA,4B;QAWyC,sBAAW,OAAL,SAAK,CAAX,C;O;KAXzC,C;0EAYA,yB;MAAA,0B;MAAA,+B;MAAA,4B;QAW0C,sBAAW,OAAL,SAAK,SAAX,C;O;KAX1C,C;Igc9WA,6B;MACqB,sB;K;uCAKjB,iB;MAM6C,OhCyUP,UgCzUO,aAAQ,KAAR,ChCyUP,C;K;uCgCvUtC,wB;MAOI,aAAQ,KAAR,IAAiB,KhCiOc,K;K;kFgC7NL,Y;MAAQ,OAAA,YAAQ,O;K;oCAE9C,Y;MAC8E,+BAAS,YAAT,C;K;IAGxD,oC;MAAiC,wB;MAAhC,oB;MACnB,eAAoB,C;K;4CACpB,Y;MAAyB,sBAAQ,YAAM,O;K;8CACvC,Y;MAAyD,Q;MAA9B,IAAI,eAAQ,YAAM,OAAIB,C;QAAA,OhCmTO,UgCnTiB,aAAM,mBAAN,EAAM,2BAAN,OhCmTjB,C;;QgCnT+C,MAAM,2BAAuB,YAAM,WAA7B,C;K;;0CAG3F,mB;MAIS,Q;MAAL,IAAI,eAAC,0EAAD,QAAJ,C;QAAiC,OAAO,K;MAExC,OAAe,WAAR,YAAQ,EAAS,OhC2MO,KgC3MhB,C;K;+CAGnB,oB;MACY,Q;MAA2B,gBAA3B,gE;MAA2B,c;;Qd0nDvB,U;QADhB,IAAI,wCAAsB,mBAA1B,C;UAAqC,aAAO,I;UAAP,e;;QACrB,6B;QAAhB,OAAgB,gBAAhB,C;UAAgB,2B;Uc1nD6B,2Bd0nDR,Oc1nDQ,Q;UAAA,W;YAAuB,oBAAR,YAAQ,Ed0nD/B,OIBn7CF,KgCvMiC,C;;Ud0nD9C,IAAI,OAAJ,C;YAAyB,aAAO,K;YAAP,e;;QAC/C,aAAO,I;;Mc3nDH,iB;K;mCAGJ,Y;MAAkC,OAAA,IAAK,QAAQ,OAAb,KAAqB,C;K;;IA/CvD,sC;MAAA,oD;MACgC,uBAAK,cAAU,IAAV,CAAL,C;MADhC,Y;K;+oCAPJ,Y;MAAA,OAKqB,qDALrB,M;K;oCAAA,Y;MAAAA,c;MAKqB,wD;MALrB,a;K;kCAAA,iB;MAAA,2IAKqB,0CALrB,G;K;gFAyDA,yB;MAAA,yC;MAWSc,yC;QAAA,wB;UAAW,OAAA,aAAK,KAAL,ChCsLV,K;S;O;MgCjMvC,6B;QAWI,OAAO,oBAAW,+BAAU,IAAV,GAAgB,uBAAhB,CAAX,C;O;KAXX,C;kFACa,oB;MAGqE,e;K;I/BtE7C,oB;MAEpB,4B;MAFuD,gB;K;IAEvD,0B;MAAA,8B;MACI,iBAGmC,SAAK,CAAL,C;MAEnC,iBAGmC,SAAK,EAAL,C;MAEnC,kBAGmC,C;MAEnC,iBAGkC,E;K;;IANbtC,sC;MAAA,qC;QAAA,oB;;MAAA,8B;K;oGAsBA,yB;MD2QA,6B;MC3PA,8C;MAhBA,wB;QAM0D,OAIbQ,YAAy,IAAK,KAAjB,EAA6B,CD6P5D,cC9QsC,KD8Q5B,KAAL,GAAiB,GAAtB,CC7P4D,MAA7B,C;O;KAvBIE,C;oGAQA,yB;MCoQA,6B;MD5PA,8C;MARA,wB;QAM2D,OASO,YAAy,IAAK,KAAjB,EAA6B,CC8P5D,cDvQuC,KCuQ7B,KAAL,GAAiB,KAAtB,CD9P4D,MAA7B,C;O;KAFIE,C;gGAQA,yB;MAAA,8C;MAAA,wB;QAOKE,mBAAY,IAAK,KAAjB,EAAuB,KAAM,KAA7B,C;O;KAPIE,C;oGASA,yB;MAGRA,kBAS6D,sB;MAT7D,+B;MgBjRA,gD;MhBCA,wB;QAM0D,OgBAS,aAAkB,ChBmRhd,eAAW,oBAAL,SAAK,CAAL,iBAAN,CgBnRgD,MAAiB,EhBAgB,KgBac,KAA9B,C;O;KhBnNE,C;0FAQA,yB;MD0OA,6B;MC1OA,wB;QAEsD,OAMD,cAAK,IAAK,KAAK,GAAW,CD2O5C,cCjP6B,KDiPnB,KAAL,GAAiB,GAAtB,CC3O4C,MAAX,IAAf,C;O;KARrD,C;0FAGA,yB;MCwOA,6B;MDxOA,wB;QAEuD,OAGF,cAAK,IAAK,KAAK,GAAW,CC4O5C,cD/O8B,KC+OpB,KAAL,GAAiB,KAAtB,CD5O4C,MAAX,IAAf,C;O;KALrD,C;0FAGA,yB;MAAA,6B;MAAA,wB;QAEqD,qBAAK,IAAK,KAAK,GAAC,KAAM,KAAX,IAAf,C;O;KAFrD,C;0FAGA,yB;MA+PA,kBAS6D,sB;MAT7D,+B;MA/PA,wB;QAEuD,OgBAA,eAAW,ChBsQ7B,eAAW,oBAAL,SAAK,CAAL,iBAAN,CgBtQ6B,MAAK,KhBAI,KgBAO,KAAX,CAAhB,C;O;KhBFvD,C;4FAIA,yB;MD6NA,6B;MC7NA,wB;QAEuD,OAMD,cAAK,IAAK,KAAK,GAAY,CD8N9C,cCpO+B,KDoOrB,KAAL,GAAiB,GAAtB,CC9N8C,MAAZ,IAAf,C;O;KARtD,C;4FAGA,yB;MC2NA,6B;MD3NA,wB;QAEwD,OAGF,cAAK,IAAK,KAAK,GAAY,CC+N9C,cDIogC,KCkOtB,KAAL,GAAiB,KAAtB,CD/N8C,MAAZ,IAAf,C;O;KALtD,C;4FAGA,yB;MAAA,6B;MAAA,wB;QAEsD,qBAAK

,IAAK,KAAK,GAAM,KAAM,KAAZ,IAAf,C;O;KAFtD,C;4FAGA,yB;MAkPA,kBAS6D,sB;MAT7D,+B;MAIPA,wB;QAEwD,OgBAA,eAAW,ChByP9B,eAAW,oBAAL,SAAK,CAAL,iBAAN,CgBzP8B,MAAK,UhBAK,KgBAO ,KAAZ,CAAhB,C;O;KhBFxD,C;4FAIA,yB;MDgNA,6B;MChNA,wB;QAEuD,OAMD,cAAe,YAAV,IAAK,KAA K,EAAy,CDiN9C,cCvN+B,KDuNrB,KAAL,GAAiB,GAAtB,CCjN8C,MAAZ,CAAf,C;O;KARtD,C;4FAGA,yB; MC8MA,6B;MD9MA,wB;QAEwD,OAGF,cAAe,YAAV,IAAK,KAAK,EAAy,CCKn9C,cDrNgC,KCqNtB,KAAL ,GAAiB,KAAtB,CDiN8C,MAAZ,CAAf,C;O;KALtD,C;4FAGA,yB;MAAA,6B;MAAA,wB;QAEsD,qBA Ae,YAA V,IAAK,KAAK,EAAM,KAAM,KAAZ,CAAf,C;O;KAFtD,C;4FAGA,yB;MAqOA,kBAS6D,sB;MAT7D,+B;MAr OA,wB;QAEwD,OgBAA,eAAW,ChB409B,eAAW,oBAAL,SAAK,CAAL,iBAAN,CgB508B,MAAK,UhBAK,Kg BAO,KAAZ,CAAhB,C;O;KhBFxD,C;wFAIA,yB;MDmMA,6B;MC7LA,4C;MANA,wB;QAEqD,OAMD,WAAW, IAAX,EDoMjB,cC1M2B,KD0MjB,KAAL,GAAiB,GAAtB,CCpMiB,C;O;KARpD,C;wFAGA,yB;MCiMA,6B;MD 9LA,4C;MAHA,wB;QAEsD,OAGF,WAAW,IAAX,ECqMjB,cDxM4B,KCwMIB,KAAL,GAAiB,KAAtB,CDrMiB ,C;O;KALpD,C;wFAGA,yB;MAAA,4C;MAAA,wB;QAEoD,kBAAW,IAAX,EAAiB,KAAjB,C;O;KAFpD,C;wFA GA,yB;MAwNA,kBAS6D,sB;MAT7D,+B;MgBxNA,8C;MhBAA,wB;QAEsD,OgBAA,YhB+NjB,eAAW,oBAAL, SAAK,CAAL,iBAAN,CgB/NiB,EhBAmb,KgBAnB,C;O;KhBFtD,C;wFAIA,yB;MDsLA,6B;MCxKA,kD;MAdA, wB;QAMqD,OAcD,cAAc,IAAd,ED2KjB,cCzL2B,KDyLjB,KAAL,GAAiB,GAAtB,CC3KiB,C;O;KApBpD,C;wF AOA,yB;MCgLA,6B;MDzKA,kD;MAPA,wB;QAMsD,OAOf,cAAc,IAAd,EC4KjB,cDnL4B,KCmLiB,KAAL,GA AaiB,KAAtB,CD5KiB,C;O;KAbpD,C;wFAOA,yB;MAAA,kD;MAAA,wB;QAMoD,qBAAc,IAAd,EAAoB,KAApB ,C;O;KANpD,C;wFAOA,yB;MA+LA,kBAS6D,sB;MAT7D,+B;MgB/LA,oD;MhBAA,wB;QAMsD,OgBAA,ehBk MjB,eAAW,oBAAL,SAAK,CAAL,iBAAN,CgBIMiB,EhBAmb,KgBAnB,C;O;KhBNtD,C;KGAQA,yB;MDyJA,6 B;MC7LA,4C;MAoCA,wB;QAMiD,OAxCG,WAAW,IAAX,EDoMjB,cC5J4B,KD4JIB,KAAL,GAAiB,GAAtB,C CpMiB,C;O;KAKpD,C;KGAQA,yB;MCmJA,6B;MD9LA,4C;MA2CA,wB;QAMkD,OA/CE,WAAW,IAAX,ECq MjB,cDtJ6B,KCsJnB,KAAL,GAAiB,KAAtB,CDrMiB,C;O;KAYCpD,C;KGAQA,yB;MAIDA,4C;MAkDA,wB;QA MgD,OAtDI,WAAW,IAAX,EAsDA,KAtDA,C;O;KAGpD,C;KGAQA,yB;MAkKA,kBAS6D,sB;MAT7D,+B;Mg BxNA,8C;MhBsDA,wB;QAMkD,OgB1DI,YhB+NjB,eAAW,oBAAL,SAAK,CAAL,iBAAN,CgB/NiB,EhB0DoB, KgB1DpB,C;O;KhBoDtD,C;wFAQA,yB;MD4HA,6B;MCxKA,kD;MDuOJ,0B;MAAA,+B;MC3LI,wB;QAQ6C,O D8LR,eAAW,OC5OI,cAAc,IAAd,ED2KjB,cC7HmB,KD6HT,KAAL,GAAiB,GAAtB,CC3KiB,CAkLf,KD0DW,C AAX,C;O;KCTMrC,C;wFASA,yB;MCoHA,6B;MDzKA,kD;MCwOJ,4B;MAAA,iC;MDnLI,wB;QAQ+C,OCsLR,g BAAY,QD7OC,cAAc,IAAd,EC4KjB,cDrHqB,KCqHX,KAAL,GAAiB,KAAtB,CD5KiB,CA4Lb,KCiDY,CAAZ,C; O;KD9LvC,C;wFASA,yB;MA9DA,kD;MA8DA,wB;QAQ2C,OAhES,cAAc,IAAd,EAgEL,KAhEK,C;O;KAwDpD ,C;wFASA,yB;MA+HA,kBAS6D,sB;MAT7D,+B;MgB/LA,oD;MhBgEA,wB;QAQ6C,OgBIES,ehBkmjB,eAAW,o BAAL,SAAK,CAAL,iBAAN,CgBIMiB,EhBkEU,KgBIEV,C;O;KhB0DtD,C;wEAUA,yB;MAAA,6B;MAAA,mB; QAMyC,qBAAK,SAAK,QAAV,C;O;KANzC,C;wEAQA,yB;MAAA,6B;MAAA,mB;QAMyC,qBAAK,SAAK,QA AV,C;O;KANzC,C;gGAQA,yB;MAAA,8C;MAAA,wB;QAE6D,0BAAU,IAAV,EAAGB,KAAhB,C;O;KAF7D,C; wFAIA,yB;MAAA,6B;MAAA,2B;QAOMD,qBAAK,aAAS,QAAAd,C;O;KAPnD,C;wFASA,yB;MAAA,6B;MAAA, 2B;QAOMD,qBAAK,cAAU,QAAf,C;O;KAPnD,C;wFASA,yB;MAAA,6B;MAAA,wB;QAEiD,qBAAK,IAAK,KA AL,GAAc,KAAM,KAAzB,C;O;KAFjD,C;SFAGA,yB;MAAA,6B;MAAA,wB;QAEgD,qBAAK,IAAK,KAAL,GA Aa,KAAM,KAAxB,C;O;KAFhD,C;wFAGA,yB;MAAA,6B;MAAA,wB;QAEiD,qBAAK,IAAK,KAAL,GAAc,KA AM,KAAzB,C;O;KAFjD,C;wEAGA,yB;MAAA,6B;MAAA,mB;QAEgC,qBAAU,CAAL,SAAL,C;O;KAFhC,C;8 EAIA,yB;MAAA,0B;MAAA,mB;QAUmC,OAAK,OAAL,SAAK,C;O;KAVxC,C;gFAWA,yB;MAAA,4B;MAAA, mB;QAUqC,OAAK,QAAL,SAAK,C;O;KAV1C,C;4EAWA,Y;MASiC,gB;K;8EACjC,yB;MAAA,kBASqD,sB;MA TrD,mB;QASmC,OAAK,oBAAL,SAAK,CAAL,iB;O;KATnC,C;gFAWA,yB;MDwDJ,0B;MAAA,+B;MCxDI,mB; QASqC,OD0DA,eAAW,OC1DX,SD0DW,CAAX,C;O;KCnErC,C;kFAUA,yB;MC+CJ,4B;MAAA,iC;MD/CI,mB; QASuC,OCiDA,gBAAY,QDjDZ,SCiDY,CAAZ,C;O;KD1DvC,C;8EAUA,Y;MAEmC,W;K;gFACnC,yB;MAAA,k BAS6D,sB;MAT7D,+B;MAAA,mB;QASqC,sBAAW,oBAAL,SAAK,CAAL,iBAAN,C;O;KATrC,C;gFAWA,yB; MASA,gD;MATA,mB;QAQqC,OAoe,aAAa,SAAb,C;O;KAFvC,C;kFASA,yB;MAAA,gD;MAAA,mB;QAMuC,o BAAa,SAAb,C;O;KANvC,C;8BAQA,Y;MAAYC,OArDD,oBAAL,SAAK,CAAL,iBAqDe,W;K;::;8BAhWtD,Y;M AAA,c;MAG2D,qD;MAH3D,a;K;4BAAA,iB;MAAA,2IAG2D,oCAH3D,G;K;sEAoWA,yB;MAAA,6B;MAAA,4B ;QAWwC,qBAAU,SAAV,C;O;KAXxC,C;wEAYA,yB;MAAA,6B;MAAA,4B;QAWyC,qBAAU,SAAV,C;O;KAX

zC,C;wEAYA,yB;MAAA,6B;MAAA,4B;QAUuC,qBAAK,SAAL,C;O;KAVvC,C;wEAWA,yB;MAAA,6B;MAAA,4B;QAWwC,qBAAK,SAAK,QAAY,C;O;KAXxC,C;uEAaA,yB;MAAA,gD;MAAA,4B;QASyC,oBAaKb,SAAlB,C;O;KATzC,C;wEAUA,yB;MAAA,gD;MAAA,4B;QAS0C,oBAaA,SAAb,C;O;KAT1C,C;Igc3ZA,4B;MACqB,sB;K;sCAKjB,iB;MAM4C,OhCuXT,SgCvXS,aAAQ,KAAR,ChCuXT,C;K;sCgCrXnC,wB;MAOI,aAAQ,KAAR,IAAiB,KhCyQY,K;K;iFgCrQH,Y;MAAQ,OAAA,YAAQ,O;K;mCAE9C,Y;MAC6E,8BAAS,YAAT,C;K;IAGvD,mC;MAAgC,uB;MAA/B,oB;MACnB,eAAoB,C;K;2CACpB,Y;MAAyB,sBAAQ,YAAM,O;K;4CACvC,Y;MAAwD,Q;MAA9B,IAAI,eAAQ,YAAM,OAAIB,C;QAAA,OhCiWK,SgCjWmB,aAAM,mBAAN,EAAM,2BAAN,OhCiWnB,C;;QgCjWgD,MAAM,2BAAuB,YAAM,WAA7B,C;K;;yCAGzF,mB;MAIS,Q;MAAL,IAAI,eAAC,0EAAD,OAAJ,C;QAAgC,OAAO,K;MAEvC,OAAe,WAAR,YAAQ,EAAS,OhCmPK,KgCnPd,C;K;8CAGnB,oB;MACY,Q;MAA2B,gBAA3B,gE;MAA2B,c;;Qf0nDvB,U;QADhB,IAAI,wCAAsB,mBAA1B,C;UAAqC,aAAO,I;UAAP,e;;QACrB,6B;QAAhB,OAAgB,gBAAhB,C;UAAgB,2B;Ue1nD6B,2Bf0nDR,Oe1nDQ,O;UAAA,W;YAAAsB,oBAAR,YAAQ,Ef0nD9B,OjB34CJ,KgC/OkC,C;;Uf0nD7C,IAAI,OAAJ,C;YAAyB,aAAO,K;YAAP,e;;QAC/C,aAAO,I;;Me3nDH,iB;K;kCAGJ,Y;MAAkC,OAAA,IAAK,QAAQ,OAAb,KAAqB,C;K;;IA/CvD,qC;MAAA,mD;MACgC,sBAAK,eAAS,IAAT,CAAL,C;MADhC,Y;K;;mCAPJ,Y;MAAA,OAKqB,oDALrB,M;K;mCAAA,Y;MAAA,c;MAKqB,wD;MALrB,a;K;iCAAA,iB;MAAA,2IAKqB,0CALrB,G;K;8EAYDA,yB;MAAA,uC;MAWOC,wC;QAAA,wB;UAAW,OAAA,aAAK,KAAL,ChC8NV,K;S;O;MgCzOrC,6B;QAWI,OAAO,mBAAU,gCAAS,IAAT,GAAe,sBAAF,CAAV,C;O;KAXX,C;gFACa,oB;MAGkE,e;K;I6LnE5C,wC;MASBiB,iC;MATBsD,2BAAgB,KAAhB,EAAuB,YAAvB,EAAqC,CAArC,C;K;kFAC7B,Y;MAAQ,iB;K;yFACD,Y;MAAQ,gB;K;2CAExC,iB;MAA8C,W7NwCoB,Y6NxCPB,U7NwCqC,KAAjB,E6NxCX,K7NwCwC,KAA7B,C6NxCpB,K;MAAA,S;QAAkB,O7NwCE,Y6NxCF,K7NwCmB,KAAjB,E6NxC0,S7NwCsB,KAA7B,C6NxCF,K;;MAAIB,W;K;kCAE9C,Y;MAKkC,O7NiCgC,Y6NjChC,U7NiCiD,KAAjB,E6NjCxB,S7NiCqD,KAA7B,C6NjChC,I;K;iCAEIC,iB;MAEY,UAAwB,M;MADhC,2CAAuB,kBAaAa,KAAM,UAAAnB,KACf,2CAAS,KAAM,MAAf,cAAwB,6CAAQ,KAAM,KAAd,QAAxB,CADe,CAAvB,C;K;mCAGJ,Y;MACI,OAAI,cAAJ,GAAe,EAAf,GAAwB,MAAK,U7NyQA,K6NzQL,QAAqB,S7NyQhB,K6NzQL,I;K;mCAE5B,Y;MAAkC,OAAE,UAAF,qBAAU,S;K;IAE5C,+B;MAAA,mC;MACI,aAC8B,cAAU,4BAAK,UAAF,EA0B,4BAAK,UAA/B,C;K;;IAFIC,2C;MAAA,0C;QAAA,yB;;MAAA,mC;K;;IAYJ,oD;MA4CI,uC;MATCI,IAAI,SAAQ,CAAZ,C;QAAuB,MAAa,gCAAYB,wBAAZB,C;MACpC,IAAI,SAAQ,WAAZ,C;QAA2B,MAAa,gCAAYB,wEAAzB,C;MAG5C,aAGyB,K;MAEzB,YAGwB,4BAA0B,KAA1B,EAAiC,YAAjC,EAA+C,IAA/C,C;MAExB,YAGuB,I;K;yCAEvB,Y;MAAgD,mCAAwB,UAAxB,EAA+B,SAA/B,EAAqC,SAArC,C;K;wCAEHd,Y;MAMqC,OAAI,YAAO,CAAX,G7NvB6B,Y6NuBf,U7NvBgC,KAAjB,E6NuBP,S7NvBoC,KAA7B,C6NuBf,IAAd,G7NvB6B,Y6NuBG,U7NvBc,KAAjB,E6NuBW,S7NvBkB,KAA7B,C6NuBG,I;K;uCAErE,iB;MAEY,UAAwB,M;MADhC,iDAA6B,kBAaAa,KAAM,UAAAnB,KACrB,2CAAS,KAAM,MAAf,cAAwB,6CAAQ,KAAM,KAAd,QAAxB,KAA8C,cAAQ,KAAM,KADvC,CAA7B,C;K;yCAGJ,Y;MACI,OAAI,cAAJ,GAAe,EAAf,GAAwB,OAAM,MAAK,U7NiNN,K6NjNC,QAAqB,S7NiNtB,K6NjNC,IAAN,SAAGD,SAAhD,I;K;yCAE5B,Y;MAAkC,OAAI,YAAO,CAAX,GAAgB,UAAF,qBAAU,SAAV,cAAqB,SAAnC,GAAGD,UAAF,2BAAgB,SAAhB,eAA4B,CAAC,SAAD,IAA5B,C;K;IAEHf,qC;MAAA,yC;K;kEACI,sC;MAQ2F,2BAAgB,UAAhB,EAA4B,QAA5B,EAAsC,IAAtC,C;K;;IAT/F,iD;MAAA,gD;QAAA,+B;;MAAA,yC;K;;IAoBiC,oD;MAAuC,uB;MACxE,sBAA2B,I;MAC3B,iBAAmC,OAAO,CAA1C,G7NxDkE,Y6NwDrB,K7NxDsC,KAAjB,E6NwDZ,I7NxDyC,KAA7B,C6NwDrB,KAA7C,G7NxDkE,Y6NwDF,K7NxDmB,KAAjB,E6NwDO,I7NxDsB,KAA7B,C6NwDF,K;MACHe,c7N2RmC,S6N3RhB,I7N2RgB,C;M6N1RnC,cAAuB,cAAJ,GAAa,KAAb,GAAwB,mB;K;gDAE3C,Y;MAAkC,qB;K;iDAEIC,Y;MACI,YAAY,W;MACZ,IAAI,6BAAS,mBAAT,QAAJ,C;QACI,IAAI,CAAC,cAAL,C;UAAc,MAAa,6B;QAC3B,iBAAU,K;;QAEV,c7NID6C,S6NkD7C,W7NIDuD,KAAK,G6NkDpD,W7NID+D,KAAX,IAAf,C;;M6NoDjD,OAAO,K;K;;IC3Hf,yB;K;mCAII,Y;MAA4B,uB;K;;IAMhC,0B;K;oCAII,Y;MAA4B,wB;K;;IAMhC,wB;K;kCAII,Y;MAA4B,sB;K;;IAMhC,yB;K;mCAII,Y;MAA4B,uB;K;;I9M5BP,qB;MAErB,6B;MAFwD,gB;K;IAExD,2B;MAAA,+B;MACI,iBAGoC,a;MAEpC,iBAGoC,c;MAEpC,kBAGmC,C;MAEnC,iBAGkC,E;K;;IANtC,uC;MAAA,sC;QAAA,qB;;MAAA,+B;K;sGAsBA,yB;MjBqRA,WAS6D,wB;MAT7D,+B;MiB7PA,gD;MAxBA,wB;QAM0D,OAYBS,aAAa,IAAK,KAAIB,EAA8B,CjB+P5D,eAAW,oBiBxRyB,KjBwR9B,KAAK,CAAL,UAAAN,CiB/P4D,MAA9B,C;O;KA/BnE,C;sGAQA,yB;Mf8QA,aAS6D,0B;MAT7D,+B;Me9PA,gD;MAhBA,wB;QAM2D,OAIbQ,aAAa,IAAK,KAAIB,EAA8B,CfgQ5D,eAAW,oBeJR0B,KfiR/B,KAAK,CAAL,YAAN,CehQ4D,MAA9B,C;O;KAvBnE,C;sGAQA,yB;MhByRA,kBAS6D

,sB;MAT7D,+B;MgBjRA,gD;MARA,wB;QAMyD,OASU,aAAa,IAAK,KAAIB,EAA8B,ChBmR5D,eAAW,oBgB5
RwB,KhB4R7B,KAAK,CAAL,iBAAN,CgBnR4D,MAA9B,C;O;KAFnE,C;kGAQA,yB;MAAA,gD;MAAA,wB;Q
AOmE,oBAAa,IAAK,KAAIB,EAAwB,KAAM,KAA9B,C;O;KAPnE,C;4FASA,yB;MjBoPA,WAS6D,wB;MAT7D
,+B;MiBpPA,wB;QAEuD,OASA,eAAM,IAAK,KAAK,KAAW,CjBkP7C,eAAW,oBiB3PiB,KjB2PtB,KAAK,CAA
L,UAAN,CiBIP6C,MAAX,CAAhB,C;O;KAXvD,C;4FAGA,yB;MfkPA,aAS6D,0B;MAT7D,+B;MeIPa,wB;QAE
wD,OAMD,eAAM,IAAK,KAAK,KAAW,CfmP7C,eAAW,oBezPkB,KfyPvB,KAAK,CAAL,YAAN,CenP6C,MA
AX,CAAhB,C;O;KARvD,C;4FAGA,yB;MhBkQA,kBAS6D,sB;MAT7D,+B;MgBIQA,wB;QAEsD,OAGC,eAAM,
IAAK,KAAK,KAAW,ChBsQ7C,eAAW,oBgBzQgB,KhByQrB,KAAK,CAAL,iBAAN,CgBtQ6C,MAAX,CAAhB,
C;O;KALvD,C;4FAGA,yB;MAAA,+B;MAAA,wB;QAEuD,sBAAM,IAAK,KAAK,KAAK,KAAM,KAAX,CAAh
B,C;O;KAFvD,C;8FAIA,yB;MjBuOA,WAS6D,wB;MAT7D,+B;MiBvOA,wB;QAEwD,OASA,eAAM,IAAK,KAA
K,UAAy,CjBqO/C,eAAW,oBiB9OmB,KjB8OxB,KAAK,CAAL,UAAN,CiBrO+C,MAAZ,CAAhB,C;O;KAXxD,
C;8FAGA,yB;MfqOA,aAS6D,0B;MAT7D,+B;MerOA,wB;QAEyD,OAMD,eAAM,IAAK,KAAK,UAAy,CfsO/C,e
AAW,oBe5OoB,Kf4OzB,KAAK,CAAL,YAAN,CetO+C,MAAZ,CAAhB,C;O;KARxD,C;8FAGA,yB;MhBqPA,kB
AS6D,sB;MAT7D,+B;MgBrPA,wB;QAEuD,OAGC,eAAM,IAAK,KAAK,UAAy,ChByP/C,eAAW,oBgB5PkB,Kh
B4PvB,KAAK,CAAL,iBAAN,CgBzP+C,MAAZ,CAAhB,C;O;KALxD,C;8FAGA,yB;MAAA,+B;MAAA,wB;QA
EwD,sBAAM,IAAK,KAAK,UAAM,KAAM,KAAZ,CAAhB,C;O;KAFxD,C;8FAIA,yB;MjB0NA,WAS6D,wB;M
AT7D,+B;MiB1NA,wB;QAEwD,OASA,eAAM,IAAK,KAAK,UAAy,CjBwN/C,eAAW,oBiBjOmB,KjBiOxB,KA
AK,CAAL,UAAN,CiBxN+C,MAAZ,CAAhB,C;O;KAXxD,C;8FAGA,yB;MfwNA,aAS6D,0B;MAT7D,+B;MexN
A,wB;QAEyD,OAMD,eAAM,IAAK,KAAK,UAAy,CfyN/C,eAAW,oBe/NoB,Kf+NzB,KAAK,CAAL,YAAN,Cez
N+C,MAAZ,CAAhB,C;O;KARxD,C;8FAGA,yB;MhBwOA,kBAS6D,sB;MAT7D,+B;MgBxOA,wB;QAEuD,OA
GC,eAAM,IAAK,KAAK,UAAy,ChB4O/C,eAAW,oBgB/OkB,KhB+OvB,KAAK,CAAL,iBAAN,CgB5O+C,MAA
Z,CAAhB,C;O;KALxD,C;8FAGA,yB;MAAA,+B;MAAA,wB;QAEwD,sBAAM,IAAK,KAAK,UAAM,KAAM,K
AAZ,CAAhB,C;O;KAFxD,C;0FAIA,yB;MjB6MA,WAS6D,wB;MAT7D,+B;MiBpMA,8C;MATA,wB;QAEsD,O
ASA,YAAy,IAAZ,EjB2MjB,eAAW,oBiBpNe,KjBoNpB,KAAK,CAAL,UAAN,CiB3MiB,C;O;KAXtD,C;0FAGA
,yB;Mf2MA,aAS6D,0B;MAT7D,+B;MerMA,8C;MANA,wB;QAEuD,OAMD,YAAy,IAAZ,Ef4MjB,eAAW,oBel
NgB,KfkNtB,KAAK,CAAL,YAAN,Ce5MiB,C;O;KARtD,C;0FAGA,yB;MhB2NA,kBAS6D,sB;MAT7D,+B;MgB
xNA,8C;MAHA,wB;QAEqD,OAGC,YAAy,IAAZ,EhB+NjB,eAAW,oBgBIoc,KhBkOnB,KAAK,CAAL,iBAAN,
CgB/NiB,C;O;KALtD,C;0FAGA,yB;MAAA,8C;MAAA,wB;QAEsD,mBAAY,IAAZ,EAakB,KAAIB,C;O;KAFtD
,C;0FAIA,yB;MjBgMA,WAS6D,wB;MAT7D,+B;MiB3KA,oD;MArBA,wB;QAMsD,OAqBA,eAAe,IAAf,EjB8Kj
B,eAAW,oBiBnMe,KjBmMpB,KAAK,CAAL,UAAN,CiB9KiB,C;O;KA3BtD,C;0FAOA,yB;Mf0LA,aAS6D,0B;M
AT7D,+B;Me5KA,oD;MAdA,wB;QAMuD,OAcD,eAAe,IAAf,Ef+KjB,eAAW,oBe7LgB,Kf6LrB,KAAK,CAAL,Y
AAN,Ce/KiB,C;O;KApBtD,C;0FAOA,yB;MhBsMA,kBAS6D,sB;MAT7D,+B;MgB/LA,oD;MAPA,wB;QAMqD,
OAOc,eAAe,IAAf,EhBkMjB,eAAW,oBgBzMc,KhByMnB,KAAK,CAAL,iBAAN,CgBIMiB,C;O;KAbtD,C;0FAO
A,yB;MAAA,oD;MAAA,wB;QAMsD,sBAae,IAAf,EAAqB,KAArB,C;O;KANtD,C;oGAQA,yB;MjBmKA,WAS6
D,wB;MAT7D,+B;MiBpMA,8C;MAiCA,wB;QAMkD,OArCI,YAAy,IAAZ,EjB2MjB,eAAW,oBiBtKgB,KjBsKr
B,KAAK,CAAL,UAAN,CiB3MiB,C;O;KA+BtD,C;oGAOA,yB;Mf6JA,aAS6D,0B;MAT7D,+B;MerMA,8C;MAw
CA,wB;QAMmD,OA5CG,YAAy,IAAZ,Ef4MjB,eAAW,oBehKiB,KfgKtB,KAAK,CAAL,YAAN,Ce5MiB,C;O;K
AsCtD,C;oGAOA,yB;MhByKA,kBAS6D,sB;MAT7D,+B;MgBxNA,8C;MA+CA,wB;QAMiD,OAnDK,YAAy,IA
AZ,EhB+NjB,eAAW,oBgB5Ke,KhB4KpB,KAAK,CAAL,iBAAN,CgB/NiB,C;O;KA6CtD,C;oGAOA,yB;MatDA,
8C;MAsDA,wB;QAMkD,OA1DI,YAAy,IAAZ,EA0DA,KA1DA,C;O;KAoDtD,C;0FAQA,yB;MjBsIA,WAS6D,w
B;MAT7D,+B;MiB3KA,oD;MjB4OJ,0B;MAAA,+B;MiBvMI,wB;QAQ6C,OjB0MP,eAAW,OiBjPK,eAAe,IAAf,E
jB8KjB,eAAW,oBiBvIM,KjBuIX,KAAK,CAAL,UAAN,CiB9KiB,CA4KjB,KjBqEY,SAAX,C;O;KiBINtC,C;0FA
SA,yB;Mf8HA,aAS6D,0B;MAT7D,+B;Me5KA,oD;Mf6OJ,4B;MAAA,iC;Me/LI,wB;QAQ+C,OfkMP,gBAAY,Qel
PE,eAAe,IAAf,Ef+KjB,eAAW,oBe/HQ,Kf+Hb,KAAK,CAAL,YAAN,Ce/KiB,CAsLf,Kf4Da,SAAZ,C;O;Ke1MxC
,C;0FASA,yB;MhBwIA,kBAS6D,sB;MAT7D,+B;MgB/LA,oD;MhBkQJ,6B;MgB3MI,wB;QAQ2C,OhB8MP,cgBv
QkB,eAAe,IAAf,EhBkMjB,eAAW,oBgBzII,KhByIT,KAAK,CAAL,iBAAN,CgBIMiB,CAGMnB,KhBuEW,QAAV
,C;O;KgBtNpC,C;0FASA,yB;MAhEA,oD;MAGEA,wB;QAQ6C,OAIes,eAAe,IAAf,EAkEL,KAIEK,C;O;KA0DtD
,C;0EAUA,yB;MAAA,+B;MAAA,mB;QAM0C,sBAAM,SAAK,MAAX,C;O;KAN1C,C;0EAQA,yB;MAAA,+B;

MAEY,UAAwB,M;MADhC,kDAA8B,kBAaA,KAAM,UAAAnB,KACtB,2CAAS,KAAM,MAAf,cAAwB,6CAAQ, KAAM,KAAd,QAAXB,KAA8C,kBAAQ,KAAM,KAAd,CADxB,CAA9B,C;K;0CAGJ,Y;MACI,OAAI,cAAJ,GAA e,EAAf,GAAwB,OAAM,M/MkND,CArCkB,U+M7KX,U/M6KsB,KAAL,KAAoB,CAVzB,U+MnKD,U/MmKO,y B+MnKG,E/MmKH,CAAN,CAUyB,MAApB,CAAN,CAqClB,MAAK,Q+MINJ,Q/MkND,CArCkB,U+M7K0B,S/ M6Kf,KAAL,KAAoB,CAVzB,U+MnKmC,S/MmK7B,yB+MnKsC,E/MmKtC,CAAN,CAUyB,MAApB,CAAN,C AqClB,MAAK,Q+MINJ,IAAN,SAAqF,cAAU,6BAAU,EAAV,CAA V,CAAyB,QAA9G,I;K;0CAE5B,Y;MAAKC, OAAI,uBAAO,CAAX,GAAgB,UAAf,qBAAU,SAA V,cAAqB,SAArB,WAAd,GAAgD,UAAf,2BAAgB,SAAhB,c AA6B,SAAD,aAA5B,W;K;IAEhF,sC;MAAA,0C;K;mEACI,sC;MAQ+F,4BAAiB,UAAjB,EAA6B,QAA7B,EA Au C,IAAvC,C;K;;;IATnG,kD;MAAA,iD;QAAA,gC;;MAAA,0C;K;;IAoBkC,qD;MAA0C,wB;MAC5E,sBAA2B,I;M AC3B,iBAAmC,kBAAO,CAA1C,G/MhDmE,a+MgDtB,K/MhDwC,KAAIB,E+MgDb,I/MhD2C,KAA9B,C+MgDt B,KAA7C,G/MhDmE,a+MgDH,K/MhDqB,KAAIB,E+MgDM,I/MhDwB,KAA9B,C+MgDH,K;MACHe,c/M0SsC, U+M1SnB,I/M0SmB,C;M+MzStC,cAAuB,cAAJ,GAAa,KAAb,GAAwB,mb;K;iDAE3C,Y;MAAKC,qB;K;mDAE I C,Y;MACI,YAAY,W;MACZ,IAAI,6BAAS,mBAAT,QAAJ,C;QACI,IAAI,CAAC,cAAL,C;UAAc,MAAa,6B;QAC 3B,iBAAU,K;;QAEV,c/M/C+C,U+M+C/C,W/M/C0D,KA AK,K+M+CvD,W/M/CkE,KAAX,CAAhB,C;;M+MiDn D,OAAO,K;K;;wEC7Hf,yB;MAAA,8C;MAAA,uB;QAOI,OAAO,MAAM,CAAN,EAAS,CAAT,C;O;KAPX,C;wE AUA,yB;MAAA,8C;MAAA,uB;QAOI,OAAO,MAAM,CAAN,EAAS,CAAT,C;O;KAPX,C;wEAUA,yB;MAAA,8 C;MAAA,uB;QAOI,OAAO,MAAM,CAAN,EAAS,CAAT,C;O;KAPX,C;wEAUA,yB;MAAA,8C;MAAA,uB;QAO I,OAAO,MAAM,CAAN,EAAS,CAAT,C;O;KAPX,C;oFC7BA,yB;MAAA,gD;MAAA,4B;QAM6C,OAAQ,ajO+R hB,ciO/RgB,C;O;KANrD,C;oGAQA,yB;M/GwCA,iB;M+GxCA,4B;QAMqD,O/GwCM,MAAO,OIH+O7B,ckH/O 6B,C;O;K+G9CIE,C;sGAQA,yB;MAAA,KE;MAAA,4B;QAMsD,OAAQ,sBjO+QzB,ciO/QyB,C;O;KAN9D,C;8FA QA,yB;MAAA,0D;MjOwWA,6B;MiOxWA,4B;QAOmD,OjO2WZ,ciO3WoB,kBjOsQtB,ciOtQsB,CjO2WpB,C;O; KiOlXvC,C;4FASA,yB;MAAA,wD;MjO+VA,6B;MiO/VA,4B;QAOkD,OjOkWX,ciOlWmB,iBjO6PrB,ciO7PqB,C jOkWnB,C;O;KiOzWvC,C;gFASA,yB;MAAA,4C;MjOsVA,6B;MiOtVA,sC;QAayD,OjOmVIB,ciOnV0B,WjO8O 5B,ciO9O4B,EAAW,QAAX,CjOmV1B,C;O;KiOhWvC,C;kFAgBA,yB;MAAA,8C;MjOsUA,6B;MiOtUA,sC;QAa 0D,OjOmUnB,ciOnU2B,YjO8N7B,ciO9N6B,EAA Y,QA AZ,CjOmU3B,C;O;KiOhVvC,C;oFAgBA,yB;MAAA,gD; MAAA,4B;QAM8C,OAAS,ajNgOhB,ciNhOgB,C;O;KANvD,C;oGAQA,yB;MAAA,gE;MAAA,4B;QAMsD,OAA S,qBjNwNxB,ciNxNwB,C;O;KAN/D,C;sGAQA,yB;MAAA,KE;MAAA,4B;QAMuD,OAAS,sBjNgNzB,ciNhNyB, C;O;KANhE,C;8FAQA,yB;MAAA,0D;MjN6SA,+B;MiN7SA,4B;QAOqD,OjNgTX,eiNhToB,kBjNuMvB,ciNvMu B,CjNgTpB,C;O;KiNvT1C,C;4FASA,yB;MAAA,wD;MjNoSA,+B;MiNpSA,4B;QAOoD,OjNuSV,eiNvSmB,iBjN 8LtB,ciN9LsB,CjNuSnB,C;O;KiN9S1C,C;+EASA,yB;MAAA,4C;MjN2RA,+B;MiN3RA,sC;QAa2D,OjNwRjB,ei NxR0B,WjN+K7B,ciN/K6B,EAAW,QAAX,CjNwR1B,C;O;KiNrS1C,C;iFAeA,yB;M/GgEA,4C;MIG4MA,+B;Mi N5QA,sC;QAa4D,OjNyQIB,ekGzMuB,WIGgG1B,ckGhG0B,EAAW,C+GhEK,Q/GgEL,IAAX,CIGyMvB,C;O;Ki NtR1C,C;oFAeA,yB;MIOWJI,6B;MkO1SJ,gD;MAKJA,4B;QAM8C,OAIJO,ajO+RhB,CDcE,cAAU,cAAL,GAAiB, GAAtB,CCdF,MiO/RgB,C;O;KA4IrD,C;oGAQA,yB;M/G1GA,iB;M+G0GA,4B;QAMsD,O/G1GK,MAAO,OnHu M3B,c2N1Ge,GAAY,GxG7FA,CwG6Fb,GAA6C,EAA7C,I;O;KOO rD,C;sGAQA,yB;MPbA,KE;MOaA,4B;QAMu D,OPbkB,sB3NkGIC,c2NIGgB,GAAW,GAAO,C;O;KOOzE,C;8FAQA,yB;MAAA,0D;MIO+LA,0B;MAAA,+B;M kO/LA,4B;QAOqD,OIOmMZ,eAAW,OkOnMS,kBIogGnB,cAAL,GAAiB,GkOhGO,CIOmMT,CAAX,C;O;KkO1 MzC,C;4FASA,yB;MAAA,wD;MIOsLA,0B;MAAA,+B;MkOtLA,4B;QAOoD,OIO0LX,eAAW,OkO1LQ,iBIOuFl B,cAAL,GAAiB,GkOvFM,CIO0LR,CAAX,C;O;KkOjMzC,C;gFAUA,yB;MAAA,4C;MIOqJA,+B;MkOrJA,sC;Q Aa2D,OIOkJjB,ekOIJ0B,WIOmD7B,ckOnD6B,EAAW,QAAX,CIOkJ1B,C;O;KkO/J1C,C;kFAeA,yB;MAAA,8C; MIOsIA,+B;MkOtIA,sC;QAa4D,OIOmIIB,ekOnI2B,YIOc9B,ckOpC8B,EAA Y,QA AZ,CIOmI3B,C;O;KkOhJ1C, C;oFAeA,yB;MhOgFI,6B;MgO3SJ,gD;MA2NA,4B;QAM+C,OA3NM,ajO+RhB,CCeE,cAAU,cAAL,GAAiB,KA AtB,CDfF,MiO/RgB,C;O;KAqNrD,C;oGAQA,yB;M/GnLA,iB;M+GmLA,4B;QAMuD,O/GnLI,MAAO,OjHkNzB, cyN3CpC,GAAY,KxGvKiD,CwGuK9D,GAA+C,EAA/C,I;O;KOMJ,C;sGAQA,yB;MPZA,KE;MOYA,4B;QAMw D,OPZoB,sBzNmCnC,cyNnCe,GAAW,KAAS,C;O;KOM5E,C;8FAQA,yB;MAAA,0D;MhOuHA,4B;MAAA,iC;M gOvHA,4B;QAouD,OhO2HZ,gBAA Y,QgO3HQ,kBhOwBrB,cAAL,GAAiB,KgOxBS,ChO2HR,CAAZ,C;O;KgOl I3C,C;4FASA,yB;MAAA,wD;MhO8GA,4B;MAAA,iC;MgO9GA,4B;QAOSD,OhOkHX,gBAA Y,QgOIHO,iBhOep B,cAAL,GAAiB,KgOfQ,ChOkHP,CAAZ,C;O;KgOzH3C,C;gFAUA,yB;MAAA,4C;MhOyFA,iC;MgOzFA,sC;QA

a6D,OhOsFhB,gBgOtF0B,WhOX9B,cgOW8B,EAAX,QAAX,ChOsF1B,C;O;KgOnG7C,C;kFAeA,yB;MAAA,8C;MhO0EA,iC;MgO1EA,sC;QAa8D,OhOuEjB,gBgOvE2B,YhO1B/B,cgO0B+B,EAAY,QAAX,ChOuE3B,C;O;KgOpF7C,C;ICtRA,qC;MAEI,SIOuIoD,ckOvI3C,CIOuI2C,EkOvIvC,CIOuIuC,C;MkOtIpD,SIOsIoD,ckOtI3C,CIOsI2C,EkOtIvC,CIOsIuC,C;MkOrIpD,OIOmDkE,YkOnDvD,EIOmDwE,KAAjB,EkOnDjD,EIOmD8E,KAA7B,CkOnDvD,KAAAX,GIOkFsD,SkOIFjC,EIOkF2C,KAAK,GkOIF3C,EIOkFuD,KAAZ,IAAf,CkOIFtD,GIOqEqD,SAAU,CAAT,SkOIFpB,EIOkF8B,KAAK,GkOIF9B,EIOkF0C,KAAZ,IAAf,CABs,MAAK,GkOrExB,CIOqEmC,KAAAX,IAAf,C;K;IkOIEzD,qC;MACI,SINwIsD,ekNxI7C,CINwI6C,EkNxIzC,CINwIyC,C;MkNvItD,SINuIsD,ekNvI7C,CINuI6C,EkNvIzC,CINuIyC,C;MkNtItD,OINqDmE,akNrDxD,EINqD0E,KAAIB,EkNrDID,EINqDgF,KAA9B,CkNrDxD,KAAAX,GIN+EwD,UkN/EnC,EIN+E8C,KAAK,UkN/E9C,EIN+E0D,KAAZ,CAAhB,CkN/ExD,GINkEuD,UAAW,CAAV,UkN/EtB,EIN+EiC,KAAK,UkN/EjC,EIN+E6C,KAAZ,CAAhB,CABU,MAAK,KkNIE3B,CINkEsC,KAAAX,CAAhB,C;K;IkN/D3D,uD;MAMBI,WAAO,CAAP,C;QAD8E,OIOwBZ,YkOvBID,KIOuBmE,KAAjB,EkOvBzC,GIOuBsE,KAA7B,CkOvBID,KAD8D,GACHD,GADgD,GIOuDxB,SkOtDf,GIOsDyB,KAAK,GkOtDxB,mBAAiB,GAAjB,EAAAsB,KAAAtB,EIO2WV,SkO3WuC,IIO2WvC,CkO3WU,CIOsDoC,KAAZ,IAAf,C;akOrDtD,WAAO,CAAP,C;QAF8E,OIOwBZ,YkOtBID,KIOsBmE,KAAjB,EkOtBzC,GIOsBsE,KAA7B,CkOtBID,KAF8D,GAEHd,GAFGd,GIO0CzB,SkOxCd,GIOwCwB,KAAK,GkOxCvB,mBAAiB,KAAjB,EAAwB,GAAxB,EIO0WV,SkO1WwC,CAAC,IAAD,IIO0WxC,CkO1WU,CIOwCkC,KAAAX,IAAf,C;;QkOvC7C,MAAA,gCAAYB,eAAzB,C;K;IAGzB,uD;MAMBI,sBAAO,CAAP,C;QADkF,OINQf,akNPnD,KINOqE,KAAIB,EkNP1C,GINowE,KAA9B,CkNPnD,KADkE,GACpD,GADoD,GINkC1B,UkNjCjB,GINiC4B,KAAK,UkNjC3B,mBAAiB,GAAjB,EAAAsB,KAAAtB,EINkWP,UkNIWoC,IINkWpC,CkNIWO,CINiCuC,KAAZ,CAAhB,C;akNhCxD,sBAAO,CAAP,C;QAFkF,OINQf,akNNnD,KINMqE,KAAIB,EkNN1C,GINMwE,KAA9B,CkNNnD,KAFkE,GAEPd,GAFOd,GINqB3B,UkNnBhB,GINmB2B,KAAK,KkNnB1B,mBAAiB,KAAjB,EAAwB,GAAxB,EINiWP,UkNjWsC,IAAD,alNiWrC,CkNjWO,CINmBqC,KAAAX,CAAhB,C;;QkNIB/C,MAAA,gCAAYB,eAAzB,C;K;IjOIDC,sB;MAEtB,8B;MAFYd,gB;K;IAEzD,4B;MAAA,gC;MACI,iBAGqC,WAAO,CAAP,C;MAErC,iBAGqC,WAAO,MAAP,C;MAErC,kBAGmC,C;MAEnC,iBAGkC,E;K;;IAnBtC,wC;MAAA,uC;QAAA,sB;;MAAA,gC;K;wGAsBA,iB;MAM0D,OAAa,0BA6OjC,SAAL,GAAiB,KA7OqB,EAAU,KF4O3C,KAAL,GAAiB,GE5OqB,C;K;oGAEvE,iB;MAOoE,OAAa,0BAoO3C,SAAL,GAAiB,KApO+B,EAAU,KAoOrD,KAAL,GAAiB,KApO+B,C;K;wGAEjF,yB;MA2PA,6B;MD5PA,8C;MCCA,wB;QAMyD,ODAS,YAAiB,CC8PhD,cAAU,SAAL,GAAiB,KAAAtB,CD9PgD,MAAjb,ECAe,KDac,KAA7B,C;O;KCNIE,C;wGAQA,yB;MA6PA,aAS6D,0B;MAT7D,+B;Me9PA,gD;MfCA,wB;QAM0D,OeAS,aAAkB,CfgQhD,eAAW,oBAAL,SAAK,CAAL,YAAN,CehQgD,MAAIB,EfAgB,KeAc,KAA9B,C;O;KfNnE,C;8FAQA,yB;MA2OA,6B;MA3OA,wB;QAEsD,ODMD,cAAU,CC4O5B,cAAU,SAAL,GAAiB,KAAAtB,CD5O4B,MAAK,GAAW,CD2O5C,cEjPsC,KFiP5B,KAAAL,GAAiB,GAAAtB,CC3O4C,MAAX,IAAf,C;O;KCRrD,C;8FAGA,yB;MAwOA,6B;MAxOA,wB;QAEuD,ODGF,cAAU,CC4O5B,cAAU,SAAL,GAAiB,KAAAtB,CD5O4B,MAAK,GCAI,KDAO,KAAAX,IAAf,C;O;KCFrD,C;8FAGA,yB;MA4OA,aAS6D,0B;MAT7D,+B;MA5OA,wB;QAEuD,OeAA,eAAW,CfmP7B,eAAW,oBAAL,SAAK,CAAL,YAAN,CenP6B,MAAK,KfAI,KeAO,KAAAX,CAAhB,C;O;KfFvD,C;gGAIA,yB;MA8NA,6B;MA9NA,wB;QAEuD,ODMD,cAAU,CC+N7B,cAAU,SAAL,GAAiB,KAAAtB,CD/N6B,MAAK,GAAy,CD8N9C,cEpOwC,KFoO9B,KAAAL,GAAiB,GAAAtB,CC9N8C,MAAZ,IAAf,C;O;KCRtD,C;gGAGA,yB;MA2NA,6B;MA3NA,wB;QAEwD,ODGF,cAAU,CC+N7B,cAAU,SAAL,GAAiB,KAAAtB,CD/N6B,MAAK,GAAy,CC+N9C,cAIoyC,KAKO/B,KAAAL,GAAiB,KAAAtB,CD/N8C,MAAZ,IAAf,C;O;KCLtD,C;gGAGA,yB;MAwNA,6B;MAxNA,wB;QAEsD,ODAA,cAAU,CC+N7B,cAAU,SAAL,GAAiB,KAAAtB,CD/N6B,MAAK,GCAK,KDAO,KAAZ,IAAf,C;O;KCFtD,C;gGAGA,yB;MA+NA,aAS6D,0B;MAT7D,+B;MA/NA,wB;QAEwD,OeAA,eAAW,CfsO9B,eAAW,oBAAL,SAAK,CAAL,YAAN,CetO8B,MAAK,UfAK,KeAO,KAAZ,CAAhB,C;O;KfFxD,C;gGAIA,yB;MAiNA,6B;MAjNA,wB;QAEuD,ODMD,cAAe,YAAL,CCKn7B,cAAU,SAAL,GAAiB,KAAAtB,CDIN6B,MAAK,EAAY,CDiN9C,cEvNwC,KFuN9B,KAAAL,GAAiB,GAAAtB,CCjN8C,MAAZ,CAAf,C;O;KCRtD,C;gGAGA,yB;MA8MA,6B;MA9MA,wB;QAEwD,ODGF,cAAe,YAAL,CCKn7B,cAAU,SAAL,GAAiB,KAAAtB,CDIN6B,MAAK,EAAY,CCkN9C,cArNyC,KaqN/B,KAAAL,GAAiB,KAAAtB,CDIN8C,MAAZ,CAAf,C;O;KCLtD,C;gGAGA,yB;MA2MA,6B;MA3MA,wB;QAEsD,ODAA,cAAe,YAAL,CCKn7B,cAAU,SAAL,GAAiB,KAAAtB,CDIN6B,MAAK,ECAK,KDAO,KAAZ,CAAf,C;O;KCFtD,C;gG

AGA,yB;MAkNA,aAS6D,0B;MAT7D,+B;MAINA,wB;QAEwD,OeAA,eAAW,CfyN9B,eAAW,oBAAL,SAAK,C
AAL,YAAN,CezN8B,MAAK,UfAK,KeAO,KAAZ,CAAhB,C;O;KfFxD,C;4FAIA,yB;MAoMA,6B;MD9LA,4C;M
CNA,wB;QAEqD,ODMD,WCqMjB,cAAU,SAAL,GAAiB,KAAtB,CDrMiB,EDoMjB,cE1MoC,KF0M1B,KAAL,
GAAiB,GAAtB,CCpMiB,C;O;KCRpD,C;4FAGA,yB;MAiMA,6B;MD9LA,4C;MCHA,wB;QAEsD,ODGF,WCqM
jB,cAAU,SAAL,GAAiB,KAAtB,CDrMiB,ECqMjB,cAxMqC,KAwM3B,KAAL,GAAiB,KAAtB,CDrMiB,C;O;KC
LpD,C;4FAGA,yB;MA8LA,6B;MD9LA,4C;MCAA,wB;QAEoD,ODAA,WCqMjB,cAAU,SAAL,GAAiB,KAAtB,
CDrMiB,ECAkB,KDAIB,C;O;KCFpD,C;4FAGA,yB;MAqMA,aAS6D,0B;MAT7D,+B;MerMA,8C;MfAA,wB;QA
EsD,OeAA,Yf4MjB,eAAW,oBAAL,SAAK,CAAL,YAAN,Ce5MiB,EfAmB,KeAnB,C;O;KfFtD,C;4FAIA,yB;MA
uLA,6B;MDzKA,kD;MCdA,wB;QAMqD,ODcD,cC4KjB,cAAU,SAAL,GAAiB,KAAtB,CD5KiB,ED2KjB,cEzLo
C,KFyL1B,KAAL,GAAiB,GAAtB,CC3KiB,C;O;KCpBpD,C;4FAOA,yB;MAgLA,6B;MDzKA,kD;MCPA,wB;QA
MsD,ODOF,cC4KjB,cAAU,SAAL,GAAiB,KAAtB,CD5KiB,EC4KjB,cAnLqC,KAmL3B,KAAL,GAAiB,KAAtB,
CD5KiB,C;O;KCbpD,C;4FAOA,yB;MAyKA,6B;MDzKA,kD;MCAA,wB;QAMoD,ODAA,cC4KjB,cAAU,SAAL,
GAAiB,KAAtB,CD5KiB,ECAkB,KDAIB,C;O;KCNpD,C;4FAOA,yB;MA4KA,aAS6D,0B;MAT7D,+B;Me5KA,o
D;MfAA,wB;QAMsD,OeAA,ef+KjB,eAAW,oBAAL,SAAK,CAAL,YAAN,Ce/KiB,EfAmB,KeAnB,C;O;KfNtD,C
;sGAQA,yB;MA0JA,6B;MD9LA,4C;MCoCA,wB;QAMiD,ODxCG,WCqMjB,cAAU,SAAL,GAAiB,KAAtB,CDr
MiB,EDoMjB,cE5JqC,KF4J3B,KAAL,GAAiB,GAAtB,CCpMiB,C;O;KCKpD,C;SGAOA,yB;MAmJA,6B;MD9L
A,4C;MC2CA,wB;QAMkD,OD/CE,WCqMjB,cAAU,SAAL,GAAiB,KAAtB,CDrMiB,ECqMjB,cAtJsC,KAsJ5B,K
AAL,GAAiB,KAAtB,CDrMiB,C;O;KCpD,C;SGAOA,yB;MA4IA,6B;MD9LA,4C;MCKDA,wB;QAMgD,ODtDI
,WCqMjB,cAAU,SAAL,GAAiB,KAAtB,CDrMiB,ECsDmB,KDtDnB,C;O;KCgDpD,C;SGAOA,yB;MA+IA,aAS6
D,0B;MAT7D,+B;MerMA,8C;MfsDA,wB;QAMkD,Oe1DI,Yf4MjB,eAAW,oBAAL,SAAK,CAAL,YAAN,Ce5Mi
B,Ef0DoB,Ke1DpB,C;O;KfoDtD,C;4FAQA,yB;MA6HA,6B;MDzKA,kD;MDuOJ,0B;MAAA,+B;ME3LI,wB;QA
Q6C,OF8LR,eAAW,OC5OI,cC4KjB,cAAU,SAAL,GAAiB,KAAtB,CD5KiB,ED2KjB,cE7H4B,KF6HIB,KAAL,G
AAiB,GAAtB,CC3KiB,CakLf,KD0DW,CAAX,C;O;KEtMrC,C;4FASA,yB;MAoHA,6B;MDzKA,kD;MCwOJ,4B;
MAAA,iC;MAnLI,wB;QAQ+C,OAsLR,gBAAY,QD7OC,cC4KjB,cAAU,SAAL,GAAiB,KAAtB,CD5KiB,EC4Kj
B,cArH8B,KAqHpB,KAAL,GAAiB,KAAtB,CD5KiB,CA4Lb,KCiDY,CAAZ,C;O;KA9LvC,C;4FASA,yB;MA2G
A,6B;MDzKA,kD;MC8DA,wB;QAQ2C,ODhes,cC4KjB,cAAU,SAAL,GAAiB,KAAtB,CD5KiB,ECgES,KDhET,
C;O;KCwDpD,C;4FASA,yB;MA4GA,aAS6D,0B;MAT7D,+B;Me5KA,oD;MfgEA,wB;QAQ6C,OelES,ef+KjB,eA
AW,oBAAL,SAAK,CAAL,YAAN,Ce/KiB,EfkeU,KeLEV,C;O;Kf0DtD,C;4EAUA,yB;MAAA,4B;MAAA,iC;MA
AA,mB;QAM2C,uBAAY,QAAL,SAAK,KAAZ,C;O;KAN3C,C;4EAQA,yB;MAAA,4B;MAAA,iC;MAAA,mB;Q
AM2C,uBAAY,QAAL,SAAK,KAAZ,C;O;KAN3C,C;oGAQA,yB;MAAA,8C;MAwEA,6B;MAxEA,wB;QAE+D,0
BA+E5B,cAAU,SAAL,GAAiB,KAAtB,CA/E4B,EA+E5B,cA/EqD,KA+E3C,KAAL,GAAiB,KAAtB,CA/E4B,C;O
;KAF/D,C;4FAIA,yB;MAAA,iC;M2LnNJ,4B;M3LmNI,wB;QAEqD,uB2LiNiC,Q3LkN1B,IAAK,K2LlNX,G3LkN
oB,KAAM,K2LlNM,C3LkNjC,C;O;KAFrD,C;0FAGA,yB;MAAA,iC;M2LjNJ,4B;M3LiNI,wB;QAEoD,uB2LhNg
C,Q3LgNzB,IAAK,K2LhNX,G3LgNmB,KAAM,K2LhNM,C3LgNhC,C;O;KAFpD,C;4FAGA,yB;MAAA,iC;M2L
/MJ,4B;M3L+MI,wB;QAEqD,uB2L9MiC,Q3L8M1B,IAAK,K2L9MX,G3L8MoB,KAAM,K2L9MM,C3L8MjC,C;
O;KAFrD,C;4EAGA,yB;MAAA,iC;M2L7MJ,4B;M3L6MI,mB;QAEkC,uB2L5MsB,QAAP,C3L4MR,S2L5Me,C3
L4MtB,C;O;KAFIC,C;kFAIA,yB;MAAA,0B;MAAA,mB;QAUmC,OAAK,OAAL,SAAK,C;O;KAVxC,C;oFAWA,
Y;MASqC,gB;K;gFACrC,Y;MASiC,OAAK,SAAL,GAAiB,K;K;kFACID,yB;MAAA,aASqD,0B;MATrD,mB;QAS
mC,OAAK,oBAAL,SAAK,CAAL,Y;O;KATnC,C;oFAWA,yB;MF+DJ,0B;MAAA,+B;ME/DI,mB;QASqC,OFIEE,
eAAW,OEjEb,SFiEa,CAAX,C;O;KE1EvC,C;sFAUA,Y;MAEuC,W;K;kFACvC,yB;MAAA,6B;MAAA,mB;QASm
C,qBAAU,SAAL,GAAiB,KAAtB,C;O;KATnC,C;oFAUA,yB;MAAA,aAS6D,0B;MAT7D,+B;MAAA,mB;QASqC
,sBAAW,oBAAL,SAAK,CAAL,YAAN,C;O;KATrC,C;oFAWA,Y;MAMqC,OApDC,SAAL,GAAiB,K;K;sFAqDI
D,Y;MAMuC,OA3DD,SAAL,GAAiB,K;K;gCA6DID,Y;MAAyC,OAAQ,CA7DX,SAAL,GAAiB,KA6DD,Y;K;,,,,;
gCA3UrD,Y;MAAA,c;MAG6D,qD;MAH7D,a;K;8BAAA,iB;MAAA,2IAG6D,oCAH7D,G;K;0EA+UA,yB;MAA
A,iC;MAAA,4B;QAW4C,uBAAY,SAAZ,C;O;KAX5C,C;4EAYA,yB;MAAA,iC;MAAA,4B;QAU6C,uBAAO,SA
AP,C;O;KAV7C,C;4EAWA,yB;MAAA,4B;MAAA,iC;MAAA,4B;QAW2C,uBAAY,QAAL,SAAK,CAAZ,C;O;K
AX3C,C;4EAYA,yB;MAAA,4B;MAAA,iC;MAAA,4B;QAW4C,uBAAY,QAAL,SAAK,SAAZ,C;O;KAX5C,C;li
C/WA,8B;MACqB,sB;K;wCAKjB,iB;MAM8C,OjCsVL,WiCtVK,aAAQ,KAAR,CjCsVL,C;K;wCiCpVzC,wB;MA

OI,aAAQ,KAAR,IAAiB,KjC4OgB,K;K;mFiCxOP,Y;MAAQ,OAAA,YAAQ,O;K;qCAE9C,Y;MAC+E,gCAAS,Y
AAT,C;K;IAGzD,qC;MAAkC,yB;MAAJC,oB;MACnB,eAAoB,C;K;6CACpB,Y;MAAYB,sBAAQ,YAAM,O;K;gD
ACvC,Y;MAA0D,Q;MAA9B,IAAI,eAAQ,YAAM,OAAIB,C;QAAA,OjCgUS,WiChUe,aAAM,mBAAN,EAAM,2
BAAN,OjCgUf,C;;QiChU8C,MAAM,2BAAuB,YAAM,WAA7B,C;K;;2CAG7F,mB;MAIS,Q;MAAL,IAAI,eAAC,
0EAAD,SAAJ,C;QAAkC,OAAO,K;MAEzC,OAAe,WAAR,YAAQ,EAAS,OjCsNS,KiCtNIB,C;K;gDAGnB,oB;M
ACY,Q;MAA2B,gBAA3B,gE;MAA2B,c;;QjB0nDvB,U;QADhB,IAAI,wCAAsB,mBAA1B,C;UAAqC,aAAO,I;U
AAP,e;;QACrB,6B;QAAhB,OAAgB,gBAAhB,C;UAAgB,2B;UiB1nD6B,2BjB0nDR,OiB1nDQ,S;UAAA,W;YAA
wB,oBAAR,YAAQ,EjB0nDhC,OhBx6CA,KiClNgC,C;;UjB0nD/C,IAAI,OAAJ,C;YAAyB,aAAO,K;YAAP,e;;QA
C/C,aAAO,I;;;MiB3nDH,iB;K;oCAGJ,Y;MAAkC,OAAA,IAAK,QAAQ,OAAb,KAAqB,C;K;;IA/CvD,uC;MAAA,
qD;MACgC,wBAAK,eAAW,IAAX,CAAL,C;MADhC,Y;K;;;qCAPJ,Y;MAAA,OAKqB,sDALrB,M;K;qCAAA,Y;
MAAA,c;MAKqB,wD;MALrB,a;K;mCAAA,iB;MAAA,2IAKqB,0CALrB,G;K;kFAyDA,yB;MAAA,2C;MAWwC
,0C;QAAA,wB;UAAW,OAAA,aAAK,KAAL,CjCiMV,K;S;O;MiC5MzC,6B;QAWI,OAAO,qBAAY,gCAAW,IAA
X,GAAiB,wBAAjB,CAAZ,C;O;KAXX,C;oFAcA,oB;MAGwE,e;K;IiM5ExE,sC;MAQ2D,OAAa,WAAb,SpOwQj
B,KAAL,GAAiB,GoOxQkB,EAAS,KAAT,C;K;IAExE,sC;MAQ4D,OAAa,WAAb,SIO+PIB,KAAL,GAAiB,KkO/
PmB,EAAS,KAAT,C;K;IAGzE,sC;MAQ0D,OAAc,WnOiR5B,oBmOjRc,SnOiRnB,KAAL,CAAL,iBmOjRiC,EA
AS,KAAT,C;K;IAExE,sC;MAOgD,uBAAc,SnNyQvB,KmNzQS,EAA6B,WAAW,KAAX,CAA7B,C;K;IAGhD,8B
;MAMqC,Q;MAAA,0DAAmB,kBAAkB,SAAlB,C;K;IAExD,qC;MAO+C,Q;MAAA,0CAAc,KAAAd,oBAAwB,kB
AAkB,SAAlB,C;K;IAGvE,+B;MAMuC,Q;MAAA,2DAAoB,kBAAkB,SAAlB,C;K;IAE3D,sC;MAOiD,Q;MAAA,
2CAAE,KAAf,oBAAYB,kBAAkB,SAAlB,C;K;IAE1E,6B;MAMmC,Q;MAAA,yDAAkB,kBAAkB,SAAlB,C;K;IA
ErD,oC;MAO6C,Q;MAAA,yCAAA,KAAb,oBAAuB,kBAAkB,SAAlB,C;K;IAEpE,8B;MAMqC,Q;MAAA,0DAA
mB,kBAAkB,SAAlB,C;K;IAExD,qC;MAO+C,Q;MAAA,0CAAc,KAAAd,oBAAwB,kBAAkB,SAAlB,C;K;IAMvE,
kC;MAM4C,kCAAsB,EAAtB,C;K;IAE5C,2C;MASmB,Q;MAAA,sBAAL,SAAK,EAAa,KAAb,C;MAAL,iB;QAA
4B,OAAO,I;;MAA7C,UAAU,I;MACV,InO/EkE,YmO+E9D,GnO/E+E,KAAjB,EAA6B,CD6P5D,SoO9KzB,6BA
AM,UpO8K6B,KAAL,GAAiB,GAAtB,CC7P4D,MAA7B,CmO+E9D,IAAJ,C;QAA2B,OAAO,I;MACIC,OpO8Oq
C,UAAW,OoO9OzC,GnOoL8B,KD0DW,CAAX,C;K;IoO3OzC,mC;MAM8C,mCAAuB,EAAvB,C;K;IAE9C,4C;
MASmB,Q;MAAA,sBAAL,SAAK,EAAa,KAAb,C;MAAL,iB;QAA4B,OAAO,I;;MAA7C,UAAU,I;MACV,InOrG
kE,YmOqG9D,GnOrG+E,KAAjB,EAA6B,CC8P5D,SkOzJzB,8BAAO,UiOyJ4B,KAAL,GAAiB,KAAtB,CD9P4D,
MAA7B,CmOqG9D,IAAJ,C;QAA4B,OAAO,I;MACnB,C,IOyNuC,WAAy,QkOzN5C,GnOwKgC,KCiDY,CAAZ,
C;K;IkOtN3C,iC;MAM0C,iCAAqB,EAArB,C;K;IAE1C,0C;MASI,WAAW,KAAX,C;MAEA,aAAa,SAAK,O;MA
CIB,IAAI,WAAU,CAAd,C;QAAiB,OAAO,I;MAExB,YAAkB,4BAAK,U;MACvB,S;MAEA,gBAAgB,qBAAK,C
AAL,C;MACHb,IAAI,YAAy,EAAhB,C;QACI,IAAI,WAAU,CAAV,IAAe,cAAa,EAAhC,C;UAAqC,OAAO,I;QA
C5C,QAAQ,C;;QAER,QAAQ,C;;MAGZ,uBAAuB,mB;MAEvB,qBAAqB,gB;MACrB,anOuMmC,SmOvMtB,KnO
uMsB,C;MmOtMnC,aAAa,W;MACb,aAAU,KAAV,MAAsB,MAAtB,M;QACI,YAAy,QAAQ,qBAAK,CAAL,CA
AR,EAAiB,KAAjB,C;QAEZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,InOnJ8D,YmOmJ1D,MnOnJ2E,KAA
jB,EmOmJjD,cnOnJ8E,KAA7B,CmOmJ1D,IAAJ,C;UACI,IAAI,+CAAkB,gBAAIB,QAAJ,C;YACI,iBnO5FwC,W
mO4FvB,KnO5FuB,EmO4Ff,MnO5Fe,C;YmO8FxC,InOvJsD,YmOuJlD,MnOvJmE,KAAjB,EmOuJzC,cnOvJsE,K
AA7B,CmOuJlD,IAAJ,C;cACI,OAAO,I;;;YAGX,OAAO,I;;;QAIIf,SnOnHkD,SAAE,YmOmHjE,MnOnH4D,KAAK
,EmOmHvD,MnOnHmE,KAAZ,CAAf,C;QmOqHID,mBAAmB,M;QACnB,SnOhJiD,SmOgJjD,MnOhJ2D,KAAK,
GAAW,CAkU5C,SmOILrB,KnOkLqB,CAIU4C,MAAX,IAAf,C;QmOiJjD,InOnK8D,YmOmK1D,MnOnK2E,KA
AjB,EmOmKjD,YnOnK8E,KAA7B,CmOmK1D,IAAJ,C;UAA2B,OAAO,I;;MAGtC,OAAO,M;K;IAGX,kC;MAM
4C,kCAAsB,EAAtB,C;K;IAE5C,2C;MASI,WAAW,KAAX,C;MAEA,aAAa,SAAK,O;MACIB,IAAI,WAAU,CAA
d,C;QAAiB,OAAO,I;MAExB,YAAmB,6BAAM,U;MACzB,S;MAEA,gBAAgB,qBAAK,CAAL,C;MACHb,IAAI,
YAAy,EAAhB,C;QACI,IAAI,WAAU,CAAV,IAAe,cAAa,EAAhC,C;UAAqC,OAAO,I;QAC5C,QAAQ,C;;QAER,
QAAQ,C;;MAIZ,uBAAuB,gD;MAEvB,qBAAqB,gB;MACrB,anN0IqC,UAAW,oBmN1InC,KnN0ImC,CAAX,C;
MmNzIrC,aAAa,2B;MACb,aAAU,KAAV,MAAsB,MAAtB,M;QACI,YAAy,QAAQ,qBAAK,CAAL,CAAR,EAAi
B,KAAjB,C;QAEZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,InN5M+D,amN4M3D,MnN5M6E,KAAIB,Em
N4MID,cnN5MgF,KAA9B,CmN4M3D,IAAJ,C;UACI,IAAI,+CAAkB,gBAAIB,QAAJ,C;YACI,iBnN1J0C,YmN0J
zB,KnN1JyB,EmN0JjB,MnN1JiB,C;YmN4J1C,InNhNuD,amNgNnD,MnNhNqE,KAAIB,EmNgN1C,cnNhNwE,K


```
jar/com/google/j2objc/annotations/ReflectionSupport.java
* /opt/cola/permits/1131003150_1612875443.99/0/j2objc-annotations-1-3-sources-3-
jar/com/google/j2objc/annotations/RetainedLocalRef.java
* /opt/cola/permits/1131003150_1612875443.99/0/j2objc-annotations-1-3-sources-3-
jar/com/google/j2objc/annotations/J2ObjCIncompatible.java
No license file was found, but licenses were detected in source scan.
```

```
/*
* Copyright 2012 Google Inc. All Rights Reserved.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):

```
* /opt/cola/permits/1131003150_1612875443.99/0/j2objc-annotations-1-3-sources-3-
jar/com/google/j2objc/annotations/Weak.java
* /opt/cola/permits/1131003150_1612875443.99/0/j2objc-annotations-1-3-sources-3-
jar/com/google/j2objc/annotations/AutoreleasePool.java
* /opt/cola/permits/1131003150_1612875443.99/0/j2objc-annotations-1-3-sources-3-
jar/com/google/j2objc/annotations/WeakOuter.java
```

1.205 rxjava 2.2.14

1.205.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/**
* Copyright (c) 2016-present, RxJava Contributors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
```

- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.
- */

Found in path(s):

- * /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/schedulers/AbstractDirectTask.java
- * /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/schedulers/SchedulerWhen.java
- * /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/schedulers/InstantPeriodicTask.java
- * /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/schedulers/ScheduledDirectTask.java
- * /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/schedulers/ScheduledDirectPeriodicTask.java

No license file was found, but licenses were detected in source scan.

/**

- * Copyright (c) 2016-present, RxJava Contributors.
- *
- * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in
- * compliance with the License. You may obtain a copy of the License at
- *
- * <http://www.apache.org/licenses/LICENSE-2.0>
- *
- * Unless required by applicable law or agreed to in writing, software distributed under the License is
- * distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
- or implied. See
- * the License for the specific language governing permissions and limitations under the License.
- */

Found in path(s):

- * /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/maybe/MaybeDoOnTerminate.java
- * /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/SingleOnSubscribe.java
- * /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/ObservableConverter.java
- * /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableBufferExactBoundary.java
- * /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/maybe/MaybeOnErrorComplete.java
- * /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/observers/LambdaConsumerIntrospection.java
- * /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableSerialized.java
- * /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-

jar/io/reactivex/functions/Cancellable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/observers/BiConsumerSingleObserver.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/BlockingFlowableLatest.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/schedulers/DisposeOnCancel.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/maybe/MaybeIgnoreElementCompletable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableDetach.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/fuseable/ScalarCallable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/observers/TestObserver.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableDoOnLifecycle.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/maybe/MaybeToFlowable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/fuseable/HasUpstreamPublisher.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/subscribers/SubscriberResourceWrapper.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableWindowBoundarySupplier.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableDoAfterNext.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableGenerate.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/single/SingleTimeout.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableRefCount.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/FlowableConverter.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/fuseable/ConditionalSubscriber.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/util/BackpressureHelper.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/BlockingObservableLatest.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableSkip.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/observers/SafeObserver.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableMap.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-

jar/io/reactivex/internal/operators/maybe/MaybeUsing.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableScalarXMap.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableToListSingle.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableHide.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/observers/ForEachWhileObserver.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/SingleTransformer.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/CompletableConverter.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableWindowTimed.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/observers/ResourceSingleObserver.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/single/SingleFlatMapIterableFlowable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableElementAt.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/subscribers/ResourceSubscriber.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/completable/CompletableLift.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/util/AtomicThrowable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableScan.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableConcatWithMaybe.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/mixed/SingleFlatMapObservable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/SingleEmitter.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableWindowBoundary.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/maybe/MaybeIgnoreElement.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableObserveOn.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/util/ArrayListSupplier.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableSampleTimed.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableFilter.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/maybe/MaybeIsEmpty.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableAnySingle.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/mixed/FlowableSwitchMapMaybe.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableReduceWithSingle.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableWindowTimed.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/SingleObserver.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/maybe/MaybeMaterialize.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/subscribers/LambdaSubscriber.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/util/MergerBiFunction.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/util/ObservableQueueDrain.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableConcatMapEagerPublisher.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/maybe/MaybeSwitchIfEmpty.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/maybe/MaybeToSingle.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/maybe/MaybeCallbackObserver.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableLastMaybe.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableMapPublisher.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/processors/FlowableProcessor.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableGroupBy.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableSingleSingle.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableScalarXMap.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/maybe/MaybeFromSingle.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableSampleWithObservable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableDistinctUntilChanged.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableFilter.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableSingleSingle.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-

jar/io/reactivex/internal/operators/flowable/AbstractFlowableWithUpstream.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/maybe/MaybeNever.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableThrottleFirstTimed.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/single/SingleSubscribeOn.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/processors/ReplayProcessor.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableFlatMapSingle.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableDoAfterNext.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/fuseable/SimplePlainQueue.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/plugins/RxJavaPlugins.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/FlowableTransformer.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableConcatWithCompletable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/single/SingleDetach.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableConcatMap.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableMergeWithSingle.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/queue/SpscArrayQueue.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableWithLatestFromMany.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/completable/CompletableToFlowable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/completable/CompletableMergeArray.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/observers/BlockingMultiObserver.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/single/SingleDefer.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableMergeWithCompletable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/completable/CompletableMergeDelayErrorArray.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/disposables/EmptyDisposable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/fuseable/HasUpstreamObservableSource.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-

jar/io/reactivex/internal/operators/flowable/FlowableFromIterable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/mixed/CompletableAndThenObservable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableElementAtSingle.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/maybe/MaybeHide.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableRetryWhen.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/mixed/MaybeFlatMapPublisher.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableCountSingle.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableReduceSeedSingle.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/parallel/ParallelConcatMap.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableSingleMaybe.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableSwitchMap.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/util/EmptyComponent.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservablePublishAlt.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/maybe/MaybeTimeoutMaybe.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/util/NotificationLite.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/fuseable/QueueFuseable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/single/SingleDelayWithPublisher.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/maybe/MaybeTakeUntilPublisher.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableMaterialize.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/util/ListAddBiConsumer.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableTakeLastOne.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableSwitchIfEmpty.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/completable/CompletableResumeNext.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableAny.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-

jar/io/reactivex/internal/util/ConnectConsumer.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/subscribers/InnerQueuedSubscriberSupport.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/completable/CompletableDoOnEvent.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/MaybeOperator.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/completable/CompletableMergeIterable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/AbstractObservableWithUpstream.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableUsing.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/completable/CompletableFromRunnable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/Observer.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/subscriptions/SubscriptionHelper.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableTakeLast.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/maybe/MaybeDoAfterSuccess.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableCreate.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableDoOnLifecycle.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/maybe/MaybeFilter.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/mixed/MaybeFlatMapObservable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableFlatMapSingle.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableRepeat.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableReduceSeedSingle.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableBuffer.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservablePublish.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableRetryBiPredicate.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableMergeWithCompletable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/single/SingleTakeUntil.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/CompletableOnSubscribe.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-

jar/io/reactivex/internal/operators/parallel/ParallelDoOnNextTry.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/util/QueueDrainHelper.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/maybe/MaybeUnsubscribeOn.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/single/SingleToFlowable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/FlowableOperator.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/completable/CompletableTimer.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableRefCount.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableFromIterable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/subscriptions/AsyncSubscription.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableFlatMapCompletable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableDelay.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableToList.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableSingle.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/observers/SubscriberCompletableObserver.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/util/BlockingHelper.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/util/BlockingIgnoringReceiver.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/functions/BooleanSupplier.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/observers/DeferredScalarObserver.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableConcatArray.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/processors/BehaviorProcessor.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableMapNotification.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/maybe/MaybeObserveOn.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableTakeLast.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/functions/Function9.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-

jar/io/reactivex/internal/operators/maybe/MaybeDefer.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/functions/Function8.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/fuseable/FuseToFlowable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/subscribers/SerializedSubscriber.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/functions/BiPredicate.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/disposables/ResettableConnectable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableAmb.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/subscriptions/SubscriptionArbiter.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableFlatMapMaybe.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableFlatMapCompletableCompletable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableDistinct.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/subjects/Subject.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableInternalHelper.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/functions/Function5.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/single/SingleNever.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableSequenceEqualSingle.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/exceptions/MissingBackpressureException.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/fuseable/HasUpstreamCompletableSource.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/schedulers/NewThreadWorker.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableDematerialize.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/ObservableOperator.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/observers/ConsumerSingleObserver.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/FlowableOnSubscribe.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/maybe/MaybeDoFinally.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableGenerate.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableFromCallable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/observers/DisposableMaybeObserver.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/mixed/ScalarXMapZHelper.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableThrottleLatest.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableRange.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableSubscribeOn.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableCountSingle.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/exceptions/OnErrorNotImplementedException.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableOnErrorNext.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableBufferTimed.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/Notification.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableFlatMap.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableSkipLast.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/observers/ResourceMaybeObserver.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableMergeWithSingle.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableReduceMaybe.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/fuseable/package-info.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservablePublishClassic.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/completable/CompletableSubscribeOn.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/maybe/MaybeMap.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableConcatWithSingle.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/util/AppendOnlyLinkedListArrayList.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/observers/BasicFuseableObserver.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservablePublishSelector.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-

jar/io/reactivex/functions/Function7.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableConcatWithCompletable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableFlattenIterable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/subscribers/TestSubscriber.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/subjects/BehaviorSubject.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/maybe/MaybeDelaySubscriptionOtherPublisher.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableOnErrorReturn.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableDematerialize.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/completable/CompletableToSingle.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableElementAt.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableDistinct.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/maybe/MaybeFlatMapIterableObservable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/subscriptions/DeferredScalarSubscription.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/subjects/UnicastSubject.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableSequenceEqualSingle.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/single/SingleDelayWithObservable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/subjects/ReplaySubject.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableTakeUntilPredicate.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableCollectSingle.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableBufferBoundary.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableSampleTimed.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/processors/AsyncProcessor.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableSamplePublisher.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/single/SingleDoOnTerminate.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-

jar/io/reactivex/internal/operators/single/SingleUsing.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableThrottleFirstTimed.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableConcatWithMaybe.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableIntervalRange.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableWithLatestFromMany.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/single/SingleDoOnSubscribe.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableFromPublisher.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/maybe/MaybeContains.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/observers/DisposableCompletableObserver.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/processors/PublishProcessor.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableLimit.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableMaterialize.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/completable/CompletableTakeUntilCompletable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/single/SingleCreate.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/completable/CompletablePeek.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/MaybeObserver.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableFromObservable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/flowables/ConnectableFlowable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableLastSingle.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableTimeout.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/completable/CompletableDetach.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/completable/CompletableCreate.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableAutoConnect.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/SingleOperator.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/single/SingleResumeNext.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-

jar/io/reactivex/internal/operators/completable/CompletableFromCallable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/BackpressureStrategy.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/schedulers/TestScheduler.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/single/SingleObserveOn.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/maybe/MaybeCreate.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableFlatMap.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/parallel/ParallelFlatMap.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/maybe/MaybeAmb.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableIntervalRange.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/functions/BiConsumer.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/observers/DisposableLambdaObserver.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/completable/CompletableOnErrorComplete.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/subscribers/StrictSubscriber.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableSkip.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/completable/CompletableDelay.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableMergeWithMaybe.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/subscriptions/ScalarSubscription.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/ObservableSource.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/annotations/SchedulerSupport.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/maybe/MaybeFlatMapSingle.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/subscribers/BlockingLastSubscriber.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/Scheduler.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableRetryBiPredicate.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableDoOnEach.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/single/SingleFlatMapCompletable.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/completable/CompletableConcat.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/CompletableEmitter.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableJust.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableLastMaybe.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/single/SingleContains.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/disposables/ReferenceDisposable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/observers/BlockingObserver.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/annotations/Beta.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableSkipLast.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/FlowableSubscriber.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableFromUnsafeSource.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/SingleConverter.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableOnBackpressureLatest.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/annotations/BackpressureKind.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/parallel/ParallelSortedJoin.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowablePublishAlt.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/CompletableOperator.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/mixed/MaterializeSingleObserver.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/observers/FutureObserver.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/observers/InnerQueuedObserverSupport.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableBufferTimed.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/disposables/CompositeDisposable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/parallel/ParallelRunOn.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableDebounceTimed.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableSwitchIfEmpty.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/subscribers/QueueDrainSubscriber.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/single/SingleLift.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/single/SingleFromPublisher.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/completable/CompletableDefer.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableBufferBoundarySupplier.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/single/SingleDoOnDispose.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableBlockingSubscribe.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableTake.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/single/SingleMaterialize.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableFlattenIterable.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/single/SingleJust.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/maybe/MaybeConcatArray.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/disposables/Disposables.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableError.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/fuseable/SimpleQueue.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableConcatWithSingle.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/subscriptions/ArrayCompositeSubscription.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/disposables/SequentialDisposable.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableInterval.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/fuseable/HasUpstreamSingleSource.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableTakePublisher.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableDefer.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/maybe/MaybeZipArray.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/util/QueueDrain.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableCombineLatest.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/processors/MulticastProcessor.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableReduceMaybe.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/CompletableSource.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableIgnoreElements.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableWithLatestFrom.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/observers/BlockingBaseObserver.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/observers/LambdaObserver.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/maybe/MaybeTakeUntilMaybe.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableBufferBoundary.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/functions/Action.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/completable/CompletableTimeout.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/functions/BiFunction.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/completable/CompletableFromPublisher.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableZipIterable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableInternalHelper.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/ObservableTransformer.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableConcatMapEager.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/maybe/MaybeCache.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/completable/CompletableFromObservable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/functions/Consumer.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableIgnoreElementsCompletable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/BlockingObservableNext.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableDetach.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-

jar/io/reactivex/internal/operators/completable/CompletableToObservable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/parallel/ParallelReduce.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/util/OpenHashSet.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/subscriptions/EmptySubscription.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableNever.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/maybe/MaybeFromRunnable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/fuseable/QueueSubscription.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/completable/CompletableDisposeOn.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/completable/CompletableFromSingle.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableHide.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableScan.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/single/SingleUnsubscribeOn.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/completable/CompletableEmpty.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableError.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/parallel/ParallelFlowableConverter.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/schedulers/SchedulerMultiWorkerSupport.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableGroupBy.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableUnsubscribeOn.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableEmpty.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableCache.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/subscribers/BlockingFirstSubscriber.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/observers/BaseTestConsumer.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableDebounce.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/annotations/Experimental.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-

jar/io/reactivex/internal/operators/observable/ObservableCollectSingle.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/ObservableEmitter.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableTakeLastTimed.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/Emitter.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/schedulers/RxThreadFactory.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/observers/QueueDrainObserver.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableZip.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/single/SingleDoAfterTerminate.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableBlockingSubscribe.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableCache.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableElementAtMaybe.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/maybe/MaybeUnsafeCreate.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/observers/BasicQueueDisposable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/maybe/MaybeEqualSingle.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableSequenceEqual.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/parallel/ParallelFilter.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/observers/CallbackCompletableObserver.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/functions/Function.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/functions/LongConsumer.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/util/SuppressAnimalSniffer.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableTakeUntil.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/maybe/AbstractMaybeWithUpstream.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/completable/CompletableErrorSupplier.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/maybe/MaybeConcatIterable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableEmpty.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableTake.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableToList.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/util/HalfSerializer.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableFlatMapPublisher.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableOnErrorNext.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableTimer.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/maybe/MaybeFromFuture.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableFromFuture.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/subjects/AsyncSubject.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/parallel/ParallelPeek.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/fuseable/FuseToObservable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/subscribers/DeferredScalarSubscriber.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/disposables/SerialDisposable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/observers/ResourceCompletableObserver.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/Flowable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableConcatMapEager.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/single/SingleHide.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableFromArray.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableLastSingle.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/observers/DisposableSingleObserver.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/disposables/Disposable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/subjects/CompletableSubject.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableZipIterable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/functions/IntFunction.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-

jar/io/reactivex/functions/Function3.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/processes/SerializedProcessor.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/Completable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/completable/CompletableUsing.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/single/SingleDoOnSuccess.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/annotations/CheckReturnValue.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/maybe/MaybeFlatMapSingleElement.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/maybe/MaybeFromCompletable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/mixed/FlowableConcatMapSingle.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/subscribers/DisposableSubscriber.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/completable/CompletableFromUnsafeSource.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableCollect.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/ObservableOnSubscribe.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableOnBackpressureDrop.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/functions/Function4.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableSkipWhile.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableDefer.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/single/SingleCache.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/single/SingleDoFinally.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableSkipLastTimed.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/maybe/MaybeSubscribeOn.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/parallel/ParallelFromArray.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/maybe/MaybeMergeArray.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/maybe/MaybeEmpty.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableRepeatUntil.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableSkipUntil.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/maybe/MaybeLift.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/completable/CompletableMerge.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/subscribers/ForEachWhileSubscriber.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/observers/ResourceObserver.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableSubscribeOn.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/fuseable/QueueDisposable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableJust.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/observers/FutureSingleObserver.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/maybe/MaybeDoOnEvent.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableThrottleLatest.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/disposables/RunnableDisposable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/maybe/MaybeCount.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableRangeLong.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableUnsubscribeOn.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/disposables/ActionDisposable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/subscribers/BoundedSubscriber.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/subscriptions/BasicIntQueueSubscription.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableDelaySubscriptionOther.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableScanSeed.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/maybe/MaybeErrorCallable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/single/SingleZipIterable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/maybe/MaybeTimer.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableRepeatWhen.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableBuffer.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableConcatMap.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/maybe/MaybeDelay.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/MaybeEmitter.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableFlatMapMaybe.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableObserveOn.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableMap.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableNever.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableElementAtMaybe.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/fuseable/HasUpstreamMaybeSource.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/maybe/MaybeFlatten.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/maybe/MaybeOnErrorReturn.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableRangeLong.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableMergeWithMaybe.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/parallel/ParallelMapTry.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/queue/SpscLinkedListArrayQueue.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/util/SorterFunction.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/completable/CompletableAndThenCompletable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowablePublishClassic.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/mixed/ObservableConcatMapMaybe.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableRetryPredicate.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableReplay.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableOnBackpressureBufferStrategy.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableRepeatWhen.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-

jar/io/reactivex/internal/operators/maybe/MaybeTimeoutPublisher.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/CompletableTransformer.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableWindowBoundarySelector.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/disposables/FutureDisposable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/BlockingFlowableNext.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/functions/ObjectHelper.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObserverResourceWrapper.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/observers/BlockingFirstObserver.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableFromFuture.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableTakeLastOne.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableWindow.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableAllSingle.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/disposables/DisposableContainer.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/disposables/SubscriptionDisposable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/flowables/GroupedFlowable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/maybe/MaybeJust.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableTakeUntil.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/util/ErrorMode.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/observers/DeferredScalarDisposable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableAutoConnect.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableFromArray.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/completable/CompletableAmb.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableLift.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableDebounce.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-

jar/io/reactivex/internal/operators/flowable/FlowableDoFinally.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/maybe/MaybeDetach.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/parallel/ParallelMap.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/util/LinkedList.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableRetryPredicate.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/BlockingObservableIterable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/completable/CompletableDoFinally.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/parallel/ParallelCollect.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/disposables/ArrayCompositeDisposable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/completable/CompletableConcatArray.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/CompletableObserver.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/maybe/MaybeFlatMapBiSelector.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/util/EndConsumerHelper.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableDebounceTimed.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/observers/EmptyCompletableObserver.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/mixed/FlowableConcatMapMaybe.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/MaybeConverter.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/observables/GroupedObservable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableTimeInterval.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/maybe/MaybeError.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/maybe/MaybeIsEmptySingle.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/schedulers/SingleScheduler.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/single/SingleDoOnError.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/single/SingleAmb.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/single/SingleDelayWithSingle.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/observers/InnerQueuedObserver.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableRepeatUntil.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableWindowBoundarySupplier.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/mixed/ObservableSwitchMapCompletable.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/completable/CompletableError.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableTimer.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/subscribers/BasicFuseableConditionalSubscriber.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableConcatMapPublisher.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/annotations/BackpressureSupport.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableSkipWhile.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/single/SingleInternalHelper.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/parallel/ParallelTransformer.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/Observable.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableOnErrorReturn.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/util/VolatileSizeArrayList.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/observers/ResumeSingleObserver.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/single/SingleMap.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/subscribers/DefaultSubscriber.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableAmb.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/single/SingleDelay.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/SingleSource.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/subscribers/FutureSubscriber.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/completable/CompletableCache.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableSingleMaybe.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/observers/DefaultObserver.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/mixed/ObservableSwitchMapSingle.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/disposables/CancellableDisposable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/completable/CompletableObserveOn.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/single/SingleError.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/BlockingFlowableIterable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableAll.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableTimeoutTimed.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/single/SingleFlatMapIterableObservable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/annotations/NonNull.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableRange.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/functions/Function6.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableWindow.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/maybe/MaybeDelayOtherPublisher.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableCount.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/functions/Functions.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableRetryWhen.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableAll.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableMapNotification.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/maybe/MaybeOnErrorNext.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/schedulers/ScheduledRunnable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/mixed/CompletableAndThenPublisher.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableTimeoutTimed.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/single/SingleZipArray.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableRepeat.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/util/ExceptionHandler.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableBufferBoundarySupplier.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/subscribers/BlockingSubscriber.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableIgnoreElementsCompletable.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/subjects/PublishSubject.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableCount.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableFlatMapCompletable.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/Maybe.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableJoin.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/maybe/MaybePeek.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/maybe/MaybeFlatMapNotification.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableLift.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableCollect.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/completable/CompletableNever.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableTakeLastTimed.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/subscribers/InnerQueuedSubscriber.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/single/SingleDoAfterSuccess.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/single/SingleFlatMap.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/single/SingleFlatMapMaybe.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/subscribers/BasicFuseableSubscriber.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/single/SingleFromUnsafeSource.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/functions/Predicate.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableDelay.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/single/SingleDelayWithCompletable.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-

jar/io/reactivex/internal/operators/flowable/FlowableInterval.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/mixed/ObservableConcatMapSingle.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/parallel/ParallelFilterTry.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/maybe/MaybeConcatArrayDelayError.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableAny.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableCombineLatest.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/single/SingleToObservable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableReplay.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/single/SingleOnErrorReturn.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/maybe/MaybeDelayWithCompletable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableFromCallable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableSkipUntil.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/maybe/MaybeFilterSingle.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/mixed/ObservableConcatMapCompletable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/exceptions/ProtocolViolationException.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/BlockingFlowableMostRecent.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/observers/DisposableObserver.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/single/SingleFromCallable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableSkipLastTimed.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/subscribers/SafeSubscriber.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/Single.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/MaybeTransformer.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableReduceWithSingle.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/observers/BlockingLastObserver.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/subscribers/SinglePostCompleteSubscriber.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/MaybeOnSubscribe.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/exceptions/Exceptions.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/observers/BasicIntQueueDisposable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/single/SingleDoOnEvent.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/mixed/FlowableSwitchMapSingle.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/processors/UnicastProcessor.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/parallel/ParallelReduceFull.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableReduce.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/subjects/SingleSubject.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/completable/CompletableHide.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/parallel/ParallelFailureHandling.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/subscribers/BlockingBaseSubscriber.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/observers/SerializedObserver.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableDoOnEach.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/completable/CompletableFromAction.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/parallel/ParallelJoin.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/schedulers/ExecutorScheduler.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableSwitchMap.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/BlockingObservableMostRecent.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/queue/MpscLinkedQueue.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableOnBackpressureBuffer.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/subjects/SerializedSubject.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/observables/ConnectableObservable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/disposables/ListCompositeDisposable.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/subjects/MaybeSubject.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableFlatMapCompletableCompletable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/FlowableEmitter.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableTimeInterval.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableWindowBoundary.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/mixed/FlowableConcatMapCompletable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableFromPublisher.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableTakeWhile.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableSerialized.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/fuseable/FuseToMaybe.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableToListSingle.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/single/SingleFlatMapPublisher.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/mixed/FlowableSwitchMapCompletable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/util/HashMapSupplier.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/subscriptions/BasicQueueSubscription.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableSequenceEqual.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/schedulers/ImmediateThinScheduler.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableZip.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/maybe/MaybeFlatMapCompletable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/maybe/MaybeToObservable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableAllSingle.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/exceptions/UndeliverableException.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/annotations/Nullable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/maybe/MaybeFlatMapIterableFlowable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-

jar/io/reactivex/internal/operators/single/SingleDematerialize.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableDelaySubscriptionOther.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/parallel/ParallelFlowable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/maybe/MaybeFromCallable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/schedulers/Timed.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableDistinctUntilChanged.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowablePublish.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/schedulers/Schedulers.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableWindowBoundarySelector.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/maybe/MaybeZipIterable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableAnySingle.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/maybe/MaybeFromAction.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableBufferExactBoundary.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/schedulers/NonBlockingThread.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/completable/CompletableConcatIterable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableTakeWhile.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableElementAtSingle.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/parallel/ParallelFromPublisher.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableTakeUntilPredicate.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableWithLatestFrom.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableScanSeed.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableUsing.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/mixed/ObservableSwitchMapMaybe.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/maybe/MaybeToPublisher.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-

jar/io/reactivex/internal/operators/completable/CompletableMaterialize.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/completable/CompletableMergeDelayErrorIterable.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableIgnoreElements.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/observable/ObservableDoFinally.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/single/SingleEquals.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/single/SingleTimer.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/disposables/DisposableHelper.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableTimeout.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/MaybeSource.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/subscriptions/BooleanSubscription.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/maybe/MaybeSwitchIfEmptySingle.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableCreate.java
No license file was found, but licenses were detected in source scan.

```
/**  
 * Copyright (c) 2016-present, RxJava Contributors.  
 * <p>  
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in  
 * compliance with the License. You may obtain a copy of the License at  
 * <p>  
 * http://www.apache.org/licenses/LICENSE-2.0  
 * <p>  
 * Unless required by applicable law or agreed to in writing, software distributed under the License is  
 * distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express  
 * or implied. See  
 * the License for the specific language governing permissions and limitations under the License.  
 */
```

Found in path(s):
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/operators/flowable/FlowableOnBackpressureError.java
No license file was found, but licenses were detected in source scan.

```
/**  
 * Copyright (c) 2016-present, RxJava Contributors.  
 *  
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in  
 * compliance with the License. You may obtain a copy of the License at
```

```
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License is
* distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
or implied. See
* the License for the specific language governing permissions and limitations under the License.
*/
/*
* Original License: https://github.com/JCTools/JCTools/blob/master/LICENSE
* Original location: https://github.com/JCTools/JCTools/blob/master/jctools-
core/src/main/java/org/jctools/util/Pow2.java
*/
```

Found in path(s):

```
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/util/Pow2.java
```

No license file was found, but licenses were detected in source scan.

```
/**
* Copyright (c) 2016-present, RxJava Contributors.
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in
* compliance with the License. You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
* Unless required by applicable law or agreed to in writing, software distributed under the License is
* distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
or implied. See
* the License for the specific language governing permissions and limitations under the License.
*/
```

Found in path(s):

```
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/schedulers/SchedulerRunnableIntrospection.java
```

No license file was found, but licenses were detected in source scan.

```
/**
* Copyright (c) 2016-present, RxJava Contributors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
```

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/annotations/package-info.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/schedulers/ComputationScheduler.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/exceptions/package-info.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/schedulers/SchedulerPoolFactory.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowablePublishMulticast.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/BackpressureOverflowStrategy.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableJoin.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/functions/package-info.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/exceptions/CompositeException.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/processors/package-info.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/subscribers/package-info.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/observables/package-info.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/observers/package-info.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/flowable/FlowableGroupJoin.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/operators/observable/ObservableGroupJoin.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/flowables/package-info.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/parallel/package-info.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/subjects/package-info.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/schedulers/package-info.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/schedulers/NewThreadScheduler.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/plugins/package-info.java

* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/internal/schedulers/IOScheduler.java

```
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/package-info.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-
jar/io/reactivex/internal/schedulers/TrampolineScheduler.java
* /opt/cola/permits/1473459956_1668478047.951896/0/rxjava-2-2-14-sources-jar/io/reactivex/disposables/package-
info.java
```

1.206 jacoco 0.8.5

1.206.1 Available under license :

License

=====

Copyright (c) 2009, 2019 Mountainminds GmbH & Co. KG and Contributors

The JaCoCo Java Code Coverage Library and all included documentation is made available by Mountainminds GmbH & Co. KG, Munich. Except indicated below, the Content is provided to you under the terms and conditions of the Eclipse Public License Version 2.0 ("EPL"). A copy of the EPL is available at [\[https://www.eclipse.org/legal/epl-2.0/\]](https://www.eclipse.org/legal/epl-2.0/)(<https://www.eclipse.org/legal/epl-2.0/>).

Please visit

[\[http://www.jacoco.org/jacoco/trunk/doc/license.html\]](http://www.jacoco.org/jacoco/trunk/doc/license.html)(<http://www.jacoco.org/jacoco/trunk/doc/license.html>)

for the complete license information including third party licenses and trademarks.

<h3>Google Code Prettify</h3>

<p>

Google Code Prettify 2010/07/21

is subject to the terms and conditions of the following license:

</p>

<pre>

Copyright 2011 Mike Samuel et al

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity

on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one

of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a

result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

```
</pre>
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<link rel="stylesheet" href="resources/doc.css" charset="UTF-8" type="text/css" />
<link rel="shortcut icon" href="resources/report.gif" type="image/gif" />
<title>JaCoCo - Eclipse Public License - Version 2.0</title>
<style type="text/css">
p.list {
margin-left: 0.5in;
margin-top: 0.05em;
margin-bottom: 0.05em;
}
</style>
</head>
<body>

<div class="breadcrumb">
<a href=" ../index.html" class="el_report">JaCoCo</a> &gt;
<span class="el_source">Eclipse Public License - Version 2.0</span>
</div>
<div id="content">

<h1>Eclipse Public License - v 2.0</h1>

<pre>
THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE
PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION
OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.
```

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial content Distributed under this Agreement, and

b) in the case of each subsequent Contributor:

i) changes to the Program, and

ii) additions to the Program;

where such changes and/or additions to the Program originate from and are Distributed by that particular Contributor. A Contribution "originates" from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include changes or additions to the Program that are not Modified Works.

"Contributor" means any person or entity that Distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions Distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

"Derivative Works" shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

"Modified Works" shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

"Distribute" means the acts of a) distributing or b) making available in any manner that enables the transfer of a copy.

"Source Code" means the form of a Program preferred for making

modifications, including but not limited to software source code, documentation source, and configuration files.

"Secondary License" means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

e) Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than

those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

3. REQUIREMENTS

3.1 If a Contributor Distributes the Program in any form, then:

a) the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and

b) the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:

i) effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and

iv) requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.

3.2 When the Program is Distributed as Source Code:

a) it must be made available under this Agreement, or if the Program (i) is combined with other material in a separate file or files made available under a Secondary License, and (ii) the initial Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available under the terms of such Secondary Licenses, and

b) a copy of this Agreement must be included with each copy of the Program.

3.3 Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations

of liability ("notices") contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all

risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the

responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.

</pre>

</div>

<div class="footer">

JaCoCo \${qualified.bundle.version}

Copyright © \${copyright.years} Mountainminds GmbH & Co. KG and Contributors

</div>

</body>

</html>

<h4>ASM</h4>

<p>

ASM 7.2 is subject to the terms and conditions of the following license:

</p>

<pre>

ASM: a very small and fast Java bytecode manipulation framework

Copyright (c) 2000-2011 INRIA, France Telecom

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from

this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```
</pre>
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <link rel="stylesheet" href="resources/doc.css" charset="UTF-8" type="text/css" />
  <link rel="shortcut icon" href="resources/report.gif" type="image/gif" />
  <title>JaCoCo - License</title>
</head>
<body>

<div class="breadcrumb">
  <a href=" ../index.html" class="el_report">JaCoCo</a> &gt;
  <span class="el_source">License</span>
</div>
<div id="content">

<h1>License</h1>

<p>
  Copyright &copy; ${copyright.years} Mountainminds GmbH &amp; Co. KG and Contributors
</p>

<p>
  The JaCoCo Java Code Coverage Library and all included documentation is made
  available by Mountainminds GmbH &amp; Co. KG, Munich. Except indicated below,
  the Content is provided to you under the terms and conditions of the Eclipse
  Public License Version 2.0 (&quot;EPL&quot;). A copy of the EPL is
  <a href="epl-2.0.html">provided</a> with this Content and is also available at
  <a href="https://www.eclipse.org/legal/epl-2.0/">https://www.eclipse.org/legal/epl-2.0/</a>.
</p>

<h2>Trademarks</h2>
```

<p>
Java and all Java-based trademarks are trademarks of Oracle Corporation in the United States, other countries, or both. Eclipse and all Eclipse related trademarks and logos are trademarks of the Eclipse Foundation, Inc. OSGi is a trademark, registered trademark, or service mark of The OSGi Alliance in the US and other countries. Apache Ant and Apache Maven are trademarks of the Apache Software Foundation. Android and Dalvik are trademarks of Google Inc. All other trademarks are the property of their respective owners.
</p>

<h2>Third Party Content</h2>

<p>
The Content includes items that have been sourced from third parties as set out below.
</p>

#{args4j.license}
#{asm.license}
#{googlecodeprettify.license}

</div>
<div class="footer">
JaCoCo #{qualified.bundle.version}
Copyright © #{copyright.years} Mountainminds GmbH & Co. KG and Contributors
</div>

</body>
</html>
<h3>args4j</h3>

<p>
args4j 2.0.28 is subject to the terms and conditions of the following license:
</p>

<pre>
Copyright (c) 2013 Kohsuke Kawaguchi and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

</pre>

1.207 asm-based-accessors-helper-used-by-json-smart 2.4.7

1.207.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work,

where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or

for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason

of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or

otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents

of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.208 okio 2.8.0

1.208.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2017 Square, Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/commonMain/okio/Utf8.kt

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2018 Square, Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/jvmMain/okio/-
DeprecatedUpgrade.kt

* /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/jvmMain/okio/-
DeprecatedUtf8.kt

* /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-
jar/commonMain/okio/PeekSource.kt

* /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/jvmMain/okio/-
DeprecatedOkio.kt

* /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/jvmMain/okio/Throttler.kt

No license file was found, but licenses were detected in source scan.

/*

* Licensed to the Apache Software Foundation (ASF) under one or more
* contributor license agreements. See the NOTICE file distributed with

* this work for additional information regarding copyright ownership.
* The ASF licenses this file to You under the Apache License, Version 2.0
* (the "License"); you may not use this file except in compliance with
* the License. You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/commonMain/okio/-Base64.kt
No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2014 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/jvmMain/okio/SegmentPool.kt
* /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/jvmMain/okio/Sink.kt
* /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/jvmMain/okio/Source.kt
* /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/jvmMain/okio/BufferedSource.kt
* /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/jvmMain/okio/BufferedSink.kt
* /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/jvmMain/okio/Buffer.kt
* /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/jvmMain/okio/Timeout.kt
* /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/commonMain/okio/SegmentPool.kt
* /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/jvmMain/okio/ForwardingSource.kt
* /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/jvmMain/okio/JvmOkio.kt
* /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/jvmMain/okio/InflaterSource.kt

* /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/commonMain/okio/Segment.kt
* /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/jvmMain/okio/GzipSink.kt
* /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/jvmMain/okio/DeflaterSink.kt
* /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/jvmMain/okio/AsyncTimeout.kt
* /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/jvmMain/okio/RealBufferedSource.kt
* /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/jvmMain/okio/ForwardingSink.kt
* /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/jvmMain/okio/GzipSource.kt
* /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/jvmMain/okio/RealBufferedSink.kt
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2015 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/jvmMain/okio/SegmentedByteString.kt
* /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/jvmMain/okio/ForwardingTimeout.kt
* /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/commonMain/okio/SegmentedByteString.kt
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2018 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,

- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.
- */

Found in path(s):

- * /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/commonMain/okio/internal/ByteString.kt
- * /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/commonMain/okio/-Util.kt
- * /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/jvmMain/okio/-Platform.kt
- * /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/commonMain/okio/-Platform.kt
- * /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/commonMain/okio/ByteString.kt
- * /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/commonMain/okio/internal/-Utf8.kt

No license file was found, but licenses were detected in source scan.

/*

- * Copyright (C) 2016 Square, Inc.
- *
- * Licensed under the Apache License, Version 2.0 (the "License");
- * you may not use this file except in compliance with the License.
- * You may obtain a copy of the License at
- *
- * <http://www.apache.org/licenses/LICENSE-2.0>
- *
- * Unless required by applicable law or agreed to in writing, software
- * distributed under the License is distributed on an "AS IS" BASIS,
- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.
- */

Found in path(s):

- * /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/jvmMain/okio/HashingSink.kt
- * /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/jvmMain/okio/Pipe.kt
- * /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/jvmMain/okio/HashingSource.kt
- * /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/commonMain/okio/Options.kt

No license file was found, but licenses were detected in source scan.

/*

- * Copyright 2014 Square Inc.
- *
- * Licensed under the Apache License, Version 2.0 (the "License");
- * you may not use this file except in compliance with the License.
- * You may obtain a copy of the License at
- *
- * <http://www.apache.org/licenses/LICENSE-2.0>
- *

* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/jvmMain/okio/ByteString.kt
No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2019 Square, Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/commonMain/okio/internal/SegmentedByteString.kt

* /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/commonMain/okio/internal/RealBufferedSink.kt

* /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/commonMain/okio/internal/RealBufferedSource.kt

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2019 Square, Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

- * See the License for the specific language governing permissions and
- * limitations under the License.
- */

Found in path(s):

- * /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/commonMain/okio/Okio.kt
- * /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/commonMain/okio/internal/Buffer.kt
- * /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/commonMain/okio/BufferedSource.kt
- * /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/commonMain/okio/BufferedSink.kt
- * /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/commonMain/okio/Timeout.kt
- * /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/commonMain/okio/Source.kt
- * /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/commonMain/okio/RealBufferedSink.kt
- * /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/commonMain/okio/RealBufferedSource.kt
- * /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/commonMain/okio/Buffer.kt
- * /opt/cola/permits/1340031362_1654618471.6843367/0/okio-2-8-0-sources-2-jar/commonMain/okio/Sink.kt

1.209 micronaut-serialization 1.3.2

1.209.1 Available under license :

Apache License

Version 2.0, January 2004

<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity

exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided

that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity,

or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.210 kafka-protobuf-provider 5.5.5

1.210.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
```

```
<modelVersion>4.0.0</modelVersion>
```

```
<parent>
```

```
<groupId>io.confluent</groupId>
```

```
<artifactId>kafka-schema-registry-parent</artifactId>
<version>5.5.5</version>
</parent>

<licenses>
  <license>
    <name>Confluent Community License</name>
    <url>http://www.confluent.io/confluent-community-license</url>
    <distribution>repo</distribution>
  </license>
  <license>
    <name>Apache License 2.0</name>
    <url>http://www.apache.org/licenses/LICENSE-2.0.html</url>
    <distribution>repo</distribution>
  </license>
</licenses>
```

```
<artifactId>kafka-protobuf-provider</artifactId>
<packaging>jar</packaging>
<name>kafka-protobuf-provider</name>
```

```
<dependencies>
  <dependency>
    <groupId>com.squareup.wire</groupId>
    <artifactId>wire-schema</artifactId>
  </dependency>
  <dependency>
    <groupId>org.jetbrains.kotlin</groupId>
    <artifactId>kotlin-stdlib</artifactId>
  </dependency>
  <dependency>
    <groupId>com.google.protobuf</groupId>
    <artifactId>protobuf-java</artifactId>
  </dependency>
  <dependency>
    <groupId>com.google.protobuf</groupId>
    <artifactId>protobuf-java-util</artifactId>
  </dependency>
  <dependency>
    <groupId>io.confluent</groupId>
    <artifactId>kafka-schema-registry-client</artifactId>
  </dependency>
  <dependency>
    <groupId>org.mockito</groupId>
    <artifactId>mockito-core</artifactId>
    <scope>test</scope>
  </dependency>
</dependencies>
```

```
<groupId>junit</groupId>
<artifactId>junit</artifactId>
<scope>test</scope>
</dependency>
</dependencies>

<build>
<plugins>
<plugin>
<groupId>com.github.os72</groupId>
<artifactId>protoc-jar-maven-plugin</artifactId>
</plugin>
</plugins>
</build>
</project>
```

Found in path(s):

* /opt/cola/permits/1340031662_1654757729.7702703/0/kafka-protobuf-provider-5-5-5-jar/META-INF/maven/io.confluent/kafka-protobuf-provider/pom.xml

1.211 io-grpc-grpc-api 1.47.0

1.211.1 Available under license :

Envoy

Copyright The Envoy Project Authors

Licensed under Apache License 2.0. See LICENSE for terms.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or

otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents

of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner].

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

protoc-gen-validate

Copyright 2019 Envoy Project Authors

Licensed under Apache License 2.0. See LICENSE for terms.

zero-allocation-hashing

Copyright 2015 Higher Frequency Trading <http://www.higherfrequencytrading.com>

Licensed under Apache License 2.0. See LICENSE for terms.

Apache License

Version 2.0, January 2004

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally

submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or

implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

/*

* Copyright 2015 The gRPC Authors

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Copyright 2014 The gRPC Authors

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

This product contains a modified portion of 'OkHttp', an open source HTTP & SPDY client for Android and Java applications, which can be obtained at:

* LICENSE:

* okhttp/third_party/okhttp/LICENSE (Apache License 2.0)

* HOMEPAGE:

* <https://github.com/square/okhttp>

* LOCATION_IN_GRP:

* okhttp/third_party/okhttp

This product contains a modified portion of 'Envoy', an open source cloud-native high-performance edge/middle/service proxy, which can be obtained at:

- * LICENSE:
 - * xds/third_party/envoy/LICENSE (Apache License 2.0)
- * NOTICE:
 - * xds/third_party/envoy/NOTICE
- * HOMEPAGE:
 - * <https://www.envoyproxy.io>
- * LOCATION_IN_GRPC:
 - * xds/third_party/envoy

This product contains a modified portion of 'protoc-gen-validate (PGV)', an open source protoc plugin to generate polyglot message validators, which can be obtained at:

- * LICENSE:
 - * xds/third_party/protoc-gen-validate/LICENSE (Apache License 2.0)
- * NOTICE:
 - * xds/third_party/protoc-gen-validate/NOTICE
- * HOMEPAGE:
 - * <https://github.com/envoyproxy/protoc-gen-validate>
- * LOCATION_IN_GRPC:
 - * xds/third_party/protoc-gen-validate

This product contains a modified portion of 'udpa', an open source universal data plane API, which can be obtained at:

- * LICENSE:
 - * xds/third_party/udpa/LICENSE (Apache License 2.0)
- * HOMEPAGE:
 - * <https://github.com/cncf/udpa>
- * LOCATION_IN_GRPC:
 - * xds/third_party/udpa

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise

designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. **Grant of Copyright License.** Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. **Grant of Patent License.** Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. **Redistribution.** You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must

include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly

negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.212 automation 1.11-8

1.212.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * dk.brics.automaton
 *
 * Copyright (c) 2001-2011 Anders Moeller
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * 3. The name of the author may not be used to endorse or promote products
 * derived from this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
 * IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
 * IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
 * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
 * DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
 * THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
 * THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */
```

Found in path(s):

```
* /opt/cola/permits/1162863214_1620924489.9/automaton-1-11-8-sources-6-
jar/dk/brics/automaton/BasicAutomata.java
* /opt/cola/permits/1162863214_1620924489.9/automaton-1-11-8-sources-6-
jar/dk/brics/automaton/TransitionComparator.java
* /opt/cola/permits/1162863214_1620924489.9/automaton-1-11-8-sources-6-
jar/dk/brics/automaton/DatatypesAutomatonProvider.java
* /opt/cola/permits/1162863214_1620924489.9/automaton-1-11-8-sources-6-
jar/dk/brics/automaton/Automaton.java
* /opt/cola/permits/1162863214_1620924489.9/automaton-1-11-8-sources-6-
jar/dk/brics/automaton/Datatypes.java
* /opt/cola/permits/1162863214_1620924489.9/automaton-1-11-8-sources-6-
jar/dk/brics/automaton/SpecialOperations.java
```

* /opt/cola/permits/1162863214_1620924489.9/0/automaton-1-11-8-sources-6-jar/dk/brics/automaton/BasicOperations.java
* /opt/cola/permits/1162863214_1620924489.9/0/automaton-1-11-8-sources-6-jar/dk/brics/automaton/StatePair.java
* /opt/cola/permits/1162863214_1620924489.9/0/automaton-1-11-8-sources-6-jar/dk/brics/automaton/MinimizationOperations.java
* /opt/cola/permits/1162863214_1620924489.9/0/automaton-1-11-8-sources-6-jar/dk/brics/automaton/State.java
* /opt/cola/permits/1162863214_1620924489.9/0/automaton-1-11-8-sources-6-jar/dk/brics/automaton/ShuffleOperations.java
* /opt/cola/permits/1162863214_1620924489.9/0/automaton-1-11-8-sources-6-jar/dk/brics/automaton/RunAutomaton.java
* /opt/cola/permits/1162863214_1620924489.9/0/automaton-1-11-8-sources-6-jar/dk/brics/automaton/RegExp.java
* /opt/cola/permits/1162863214_1620924489.9/0/automaton-1-11-8-sources-6-jar/dk/brics/automaton/AutomatonProvider.java
* /opt/cola/permits/1162863214_1620924489.9/0/automaton-1-11-8-sources-6-jar/dk/brics/automaton/Transition.java

No license file was found, but licenses were detected in source scan.

/*

* dk.brics.automaton - AutomatonMatcher

*

* Copyright (c) 2008-2011 John Gibson

* All rights reserved.

*

* Redistribution and use in source and binary forms, with or without

* modification, are permitted provided that the following conditions

* are met:

* 1. Redistributions of source code must retain the above copyright

* notice, this list of conditions and the following disclaimer.

* 2. Redistributions in binary form must reproduce the above copyright

* notice, this list of conditions and the following disclaimer in the

* documentation and/or other materials provided with the distribution.

* 3. The name of the author may not be used to endorse or promote products

* derived from this software without specific prior written permission.

*

* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR

* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES

* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.

* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,

* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT

* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,

* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY

* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT

* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF

* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

Found in path(s):

* /opt/cola/permits/1162863214_1620924489.9/0/automaton-1-11-8-sources-6-

1.213 objenesis 2.6

1.213.1 Available under license :

```
// -----  
// NOTICE file corresponding to the section 4d of The Apache License,  
// Version 2.0, in this case for Objenesis  
// -----
```

Objenesis

Copyright 2006-2017 Joe Walnes, Henri Tremblay, Leonardo Mesquita

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a

cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,

any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.214 javabeans-activation-framework-api

1.2.1

1.214.1 Available under license :

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.1

1. Definitions.

1.1. "Contributor" means each individual or entity that creates or contributes to the creation of Modifications.

1.2. "Contributor Version" means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. "Covered Software" means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. "Executable" means the Covered Software in any form other than Source Code.

1.5. "Initial Developer" means the individual or entity that first makes Original Software available under this License.

1.6. "Larger Work" means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. "License" means this document.

1.8. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. "Modifications" means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. "Original Software" means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. "Source Code" means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or

portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Oracle is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a

Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as "Participant") alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. If You assert a patent infringement claim against Participant alleging that the Participant Software directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

6.4. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and "commercial computer software documentation" as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION

LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor
Boston, MA 02110-1335
USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and

you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of

Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by

modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries

not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1335 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type
`show w'. This is free software, and you are welcome to redistribute
it under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the
```

program `Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

#

Certain source files distributed by Oracle America, Inc. and/or its affiliates are subject to the following clarification and special exception to the GPLv2, based on the GNU Project exception for its Classpath libraries, known as the GNU Classpath Exception, but only where Oracle has expressly included in the particular source file's header the words "Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the LICENSE file that accompanied this code."

You should also note that Oracle includes multiple, independent programs in this software package. Some of those programs are provided under licenses deemed incompatible with the GPLv2 by the Free Software Foundation and others. For example, the package includes programs licensed under the Apache License, Version 2.0. Such programs are licensed to you under their original licenses.

Oracle facilitates your further distribution of this package by adding the Classpath Exception to the necessary parts of its GPLv2 code, which permits you to use that code in combination with other independent modules not licensed under the GPLv2. However, note that this would not permit you to commingle code under an incompatible license with Oracle's GPLv2 licensed code by, for example, cutting and pasting such code into a file also containing Oracle's GPLv2 licensed code and then distributing the result. Additionally, if you were to remove the Classpath Exception from any of the files to which it applies and distribute the result, you would likely be required to license some or all of the other code in that distribution under the GPLv2 as well, and since the GPLv2 is incompatible with the license terms of some items included in the distribution by Oracle, removing the Classpath Exception could therefore effectively compromise your ability to further distribute the package.

Proceed with caution and we recommend that you obtain the advice of a lawyer skilled in open source matters before removing the Classpath

Exception or making modifications to this package which may subsequently be redistributed and/or involve the use of third party software.

CLASSPATH EXCEPTION

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

1.215 micronaut-jax-rs 3.4.0

1.215.1 Available under license :

Apache License

Version 2.0, January 2004

<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the

Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside

or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer,

and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.216 commons-io 2.11.0

1.216.1 Available under license :

Apache Commons IO
Copyright 2002-2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<https://www.apache.org/>).

Apache License
Version 2.0, January 2004

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally

submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or

implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.217 fabric8-::-kubernetes-model-::-events

4.13.3

1.217.1 Available under license :

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0
Bnd-LastModified: 1619068588388
Build-Jdk-Spec: 1.8
Bundle-Description: Java client for Kubernetes and OpenShift
Bundle-DocURL: <http://redhat.com>
Bundle-License: <http://www.apache.org/licenses/LICENSE-2.0.txt>
Bundle-ManifestVersion: 2
Bundle-Name: Fabric8 :: Kubernetes Model :: Events
Bundle-SymbolicName: io.fabric8.kubernetes-model-events
Bundle-Vendor: Red Hat
Bundle-Version: 4.13.3
Created-By: Apache Maven Bundle Plugin
Export-Package: io.fabric8.kubernetes.api.model.events;uses:="com.fasterxml.jackson.annotation,com.fasterxml.jackson.databind,com.fasterxml.jackson.databind.annotation,io.fabric8.kubernetes.api.builder,io.fabric8.kubernetes.api.model,io.fabric8.kubernetes.model.annotation";version="4.13.3"
Implementation-Title: Fabric8 :: Kubernetes Model :: Events
Implementation-Vendor: Red Hat
Implementation-Version: 4.13.3
Import-Package: com.fasterxml.jackson.annotation;version="[2.11,3)",com.fasterxml.jackson.databind;version="[2.11,3)",com.fasterxml.jackson.databind.annotation;version="[2.11,3)",io.fabric8.kubernetes.api.builder;version="[4.13,5)",io.fabric8.kubernetes.api.model,io.fabric8.kubernetes.model.annotation;version="[4.13,5)"
Require-Capability: osgi.ee;filter="(&(osgi.ee=JavaSE)(version=1.8))"
Specification-Title: Fabric8 :: Kubernetes Model :: Events
Specification-Vendor: Red Hat
Specification-Version: 4.13
Tool: Bnd-5.1.1.202006162103

Found in path(s):

* /opt/cola/permits/1288520163_1647861771.99/0/kubernetes-model-events-4-13-3-jar/META-INF/MANIFEST.MF

No license file was found, but licenses were detected in source scan.

Copyright (C) 2015 Red Hat, Inc.

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE>

2.0

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

Found in path(s):

* /opt/cola/permits/1288520163_1647861771.99/0/kubernetes-model-events-4-13-3-jar/META-INF/maven/io.fabric8/kubernetes-model-events/pom.xml

No license file was found, but licenses were detected in source scan.

*

* Copyright (C) 2015 Red Hat, Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*

Found in path(s):

* /opt/cola/permits/1288520163_1647861771.99/0/kubernetes-model-events-4-13-3-jar/manifest.vm

1.218 findbugs-jsr305 3.0.2

1.218.1 Available under license :

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

1. "Collective Work" means a work, such as a periodical issue, anthology or encyclopedia, in which the Work in its entirety in unmodified form, along with a number of other contributions, constituting separate and independent works in themselves, are assembled into a collective whole. A work that constitutes a Collective Work will not be considered a Derivative Work (as defined below) for the purposes of this License.

2. "Derivative Work" means a work based upon the Work or upon the Work and other pre-existing works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which the Work may be recast, transformed, or adapted, except that a work that constitutes a Collective Work will not be considered a Derivative Work for the purpose of this License. For the avoidance of doubt, where the Work is a musical composition or sound recording, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered a Derivative Work for the purpose of this License.

3. "Licensor" means the individual or entity that offers the Work under the terms of this License.

4. "Original Author" means the individual or entity who created the Work.

5. "Work" means the copyrightable work of authorship offered under the terms of this License.

6. "You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

2. Fair Use Rights. Nothing in this license is intended to reduce, limit, or restrict any rights arising from fair use, first sale or other limitations on the exclusive rights of the copyright owner under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

1. to reproduce the Work, to incorporate the Work into one or more Collective Works, and to reproduce the Work as incorporated in the Collective Works;

2. to create and reproduce Derivative Works;

3. to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission the Work including as incorporated in Collective Works;

4. to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission Derivative Works.

5.

For the avoidance of doubt, where the work is a musical composition:

1. Performance Royalties Under Blanket Licenses. Licensor waives the exclusive right to collect, whether individually or via a performance rights society (e.g. ASCAP, BMI, SESAC), royalties for the public performance or public digital performance (e.g. webcast) of the Work.

2. Mechanical Rights and Statutory Royalties. Licensor waives the exclusive right to collect, whether individually or via a music rights agency or designated agent (e.g. Harry Fox Agency), royalties for any phonorecord You create from the Work ("cover version") and distribute, subject to the compulsory license created by 17 USC Section 115 of the US Copyright Act (or the equivalent in other jurisdictions).

6. Webcasting Rights and Statutory Royalties. For the avoidance of doubt, where the Work is a sound recording, Licensor waives the exclusive right to collect, whether individually or via a performance-rights society (e.g. SoundExchange), royalties for the public digital performance (e.g. webcast) of the Work, subject to the compulsory license created by 17 USC Section 114 of the US Copyright Act (or the equivalent in other jurisdictions).

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. All rights not expressly granted by Licensor are hereby reserved.

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

1. You may distribute, publicly display, publicly perform, or publicly digitally perform the Work only under the terms of this License, and You must include a copy of, or the Uniform Resource Identifier for, this License with every copy or phonorecord of the Work You distribute, publicly display, publicly perform, or publicly digitally perform. You may not offer or impose any terms on the Work that alter or restrict the terms of this License or the recipients' exercise of the rights granted hereunder. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute, publicly display, publicly perform, or publicly digitally perform the Work with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this License Agreement. The above applies to the Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Work itself to be made subject to the terms of this License. If You create a Collective Work, upon notice from any Licensor You must, to the extent practicable, remove from the Collective Work any credit as required by clause 4(b), as requested. If You create a Derivative Work, upon notice from any Licensor You must, to the extent practicable, remove from the Derivative Work any credit as required by clause 4(b), as requested.

2. If you distribute, publicly display, publicly perform, or publicly digitally perform the Work or any Derivative Works or Collective Works, You must keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or (ii) if the Original Author and/or Licensor designate another party or parties (e.g. a sponsor institute, publishing entity, journal) for attribution in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; the title of the Work if supplied; to the extent reasonably practicable, the Uniform Resource Identifier, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and in the case of a Derivative Work, a credit identifying the use of the Work in the Derivative Work (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). Such credit may be implemented in any reasonable manner; provided, however, that in the case of a Derivative Work or Collective Work, at a minimum such credit will appear where any other comparable authorship credit appears and in a manner at least as prominent as such other comparable authorship credit.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

1. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Derivative Works or Collective Works from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

2. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

1. Each time You distribute or publicly digitally perform the Work or a Collective Work, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

2. Each time You distribute or publicly digitally perform a Derivative Work, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.

3. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

4. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

5. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

1.219 spring-framework 5.3.23

1.219.1 Available under license :

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the

editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the

same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<https://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Spring Framework \${version}
Copyright (c) 2002-\${copyright} Pivotal, Inc.

This product is licensed to you under the Apache License, Version 2.0 (the "License"). You may not use this product except in compliance with the License.

This product may include a number of subcomponents with separate copyright notices and license terms. Your use of the source code for these subcomponents is subject to the terms and conditions of the subcomponent's license, as noted in the license.txt file.

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of,

publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution

notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing

the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<https://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

=====

SPRING FRAMEWORK \${version} SUBCOMPONENTS:

Spring Framework \${version} includes a number of subcomponents with separate copyright notices and license terms. The product that includes this file does not necessarily use all the open source subcomponents referred to below. Your use of the source code for these subcomponents is subject to the terms and conditions of the following licenses.

>>> ASM 9.1 (org.ow2.asm:asm:9.1, org.ow2.asm:asm-commons:9.1):

Copyright (c) 2000-2011 INRIA, France Telecom
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 1999-2009, OW2 Consortium <<https://www.ow2.org/>>

>>> CGLIB 3.3 (cglib:cglib:3.3):

Per the LICENSE file in the CGLIB JAR distribution downloaded from https://github.com/cglib/cglib/releases/download/RELEASE_3_3_0/cglib-3.3.0.jar, CGLIB 3.3 is licensed under the Apache License, version 2.0, the text of which is included above.

>>> Objenesis 3.2 (org.objenesis:objenesis:3.2):

Per the LICENSE file in the Objenesis ZIP distribution downloaded from <http://objenesis.org/download.html>, Objenesis 3.2 is licensed under the

Apache License, version 2.0, the text of which is included above.

Per the NOTICE file in the Objenesis ZIP distribution downloaded from <http://objenesis.org/download.html> and corresponding to section 4d of the Apache License, Version 2.0, in this case for Objenesis:

Objenesis

Copyright 2006-2019 Joe Walnes, Henri Tremblay, Leonardo Mesquita

To the extent any open source components are licensed under the EPL and/or other similar licenses that require the source code and/or modifications to source code to be made available (as would be noted above), you may obtain a copy of the source code corresponding to the binaries for such open source components and modifications thereto, if any, (the "Source Files"), by downloading the Source Files from <https://spring.io/projects>, Pivotal's website at <https://network.pivotal.io/open-source>, or by sending a request, with your name and address to: Pivotal Software, Inc., 875 Howard Street, 5th floor, San Francisco, CA 94103, Attention: General Counsel. All such requests should clearly specify: OPEN SOURCE FILES REQUEST, Attention General Counsel. Pivotal can mail a copy of the Source Files to you on a CD or equivalent physical medium.

This offer to obtain a copy of the Source Files is valid for three years from the date you acquired this Software product. Alternatively, the Source Files may accompany the Software.

1.220 micronaut-groovy 3.3.1

1.220.1 Available under license :

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all

other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and

subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed

as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the

Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.221 json-path 5.2.0

1.221.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2020 the original author or authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1473459942_1668478053.02154/0/json-path-5-2-0-sources-
jar/io/restassured/path/json/mapper/factory/DefaultYassonObjectMapperFactory.java
* /opt/cola/permits/1473459942_1668478053.02154/0/json-path-5-2-0-sources-
jar/io/restassured/path/json/mapper/factory/JsonbObjectMapperFactory.java
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2019 the original author or authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1473459942_1668478053.02154/0/json-path-5-2-0-sources-
jar/io/restassured/path/json/mapper/factory/GsonObjectMapperFactory.java
* /opt/cola/permits/1473459942_1668478053.02154/0/json-path-5-2-0-sources-
jar/io/restassured/path/json/exception/JsonPathException.java
* /opt/cola/permits/1473459942_1668478053.02154/0/json-path-5-2-0-sources-
```

jar/io/restassured/internal/path/json/ConfigurableJsonSlurper.groovy
* /opt/cola/permits/1473459942_1668478053.02154/0/json-path-5-2-0-sources-
jar/io/restassured/path/json/config/JsonParserType.java
* /opt/cola/permits/1473459942_1668478053.02154/0/json-path-5-2-0-sources-
jar/io/restassured/path/json/mapper/factory/DefaultGsonObjectMapperFactory.java
* /opt/cola/permits/1473459942_1668478053.02154/0/json-path-5-2-0-sources-
jar/io/restassured/path/json/mapper/factory/DefaultJackson1ObjectMapperFactory.java
* /opt/cola/permits/1473459942_1668478053.02154/0/json-path-5-2-0-sources-
jar/io/restassured/path/json/mapper/factory/DefaultJackson2ObjectMapperFactory.java
* /opt/cola/permits/1473459942_1668478053.02154/0/json-path-5-2-0-sources-
jar/io/restassured/path/json/config/JsonPathConfig.java
* /opt/cola/permits/1473459942_1668478053.02154/0/json-path-5-2-0-sources-
jar/io/restassured/path/json/mapper/factory/Jackson1ObjectMapperFactory.java
* /opt/cola/permits/1473459942_1668478053.02154/0/json-path-5-2-0-sources-
jar/io/restassured/path/json/mapper/factory/JohnzonObjectMapperFactory.java
* /opt/cola/permits/1473459942_1668478053.02154/0/json-path-5-2-0-sources-
jar/io/restassured/path/json/mapping/JsonPathObjectDeserializer.java
* /opt/cola/permits/1473459942_1668478053.02154/0/json-path-5-2-0-sources-
jar/io/restassured/internal/path/json/JSONAssertion.groovy
* /opt/cola/permits/1473459942_1668478053.02154/0/json-path-5-2-0-sources-
jar/io/restassured/path/json/mapper/factory/DefaultJohnzonObjectMapperFactory.java
* /opt/cola/permits/1473459942_1668478053.02154/0/json-path-5-2-0-sources-
jar/io/restassured/path/json/mapper/factory/Jackson2ObjectMapperFactory.java
* /opt/cola/permits/1473459942_1668478053.02154/0/json-path-5-2-0-sources-
jar/io/restassured/path/json/JsonPath.java

1.222 azure-java-sdk-bom-(bill-of-materials)

1.2.4

1.222.1 Available under license :

Apache License

Version 2.0, January 2004

<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common

control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or

documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill,

work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.223 javax.inject:javax.inject 1

1.223.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2009 The JSR-330 Expert Group
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1299411403_1650627395.55/0/javax-inject-1-sources-jar/javax/inject/Provider.java
* /opt/cola/permits/1299411403_1650627395.55/0/javax-inject-1-sources-jar/javax/inject/Named.java
* /opt/cola/permits/1299411403_1650627395.55/0/javax-inject-1-sources-jar/javax/inject/Qualifier.java
* /opt/cola/permits/1299411403_1650627395.55/0/javax-inject-1-sources-jar/javax/inject/Inject.java
* /opt/cola/permits/1299411403_1650627395.55/0/javax-inject-1-sources-jar/javax/inject/package-info.java
* /opt/cola/permits/1299411403_1650627395.55/0/javax-inject-1-sources-jar/javax/inject/Scope.java
* /opt/cola/permits/1299411403_1650627395.55/0/javax-inject-1-sources-jar/javax/inject/Singleton.java
```

1.224 jackson-integration-for-metrics 4.0.5

1.224.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
Manifest-Version: 1.0
Bnd-LastModified: 1545938158164
Build-Jdk: 1.8.0_191
Built-By: artem
Bundle-Description: A set of Jackson modules which provide serializers
for most Metrics classes.
Bundle-License: http://www.apache.org/licenses/LICENSE-2.0.html
Bundle-ManifestVersion: 2
Bundle-Name: Jackson Integration for Metrics
Bundle-SymbolicName: io.dropwizard.metrics.json
Bundle-Version: 4.0.5
Created-By: Apache Maven Bundle Plugin
Export-Package: com.codahale.metrics.json;uses:="com.codahale.metrics,
com.fasterxml.jackson.core,com.fasterxml.jackson.databind";version="4
```

.0.5"

Implementation-Title: Jackson Integration for Metrics

Implementation-URL: <http://metrics.dropwizard.io/metrics-json>

Implementation-Vendor-Id: io.dropwizard.metrics

Implementation-Version: 4.0.5

Import-Package: com.codahale.metrics;version="[4.0,5)",com.codahale.metrics.health;version="[4.0,5)";resolution:=optional,com.fasterxml.jackson.core;version="[2.9,3)",com.fasterxml.jackson.databind;version="[2.9,3)",com.fasterxml.jackson.databind.module;version="[2.9,3)",com.fasterxml.jackson.databind.ser;version="[2.9,3)",com.fasterxml.jackson.databind.ser.std;version="[2.9,3)"

Require-Capability: osgi.ee;filter="(&(osgi.ee=JavaSE)(version=1.8))"

Tool: Bnd-3.3.0.201609221906

Found in path(s):

* /opt/cola/permits/1274705522_1648836004.08/0/metrics-json-4-0-5-jar/META-INF/MANIFEST.MF

1.225 fastutil 8.2.3

1.225.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2002-2017 Sebastiano Vigna

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

package PACKAGE;

import java.util.Iterator;

import java.util.ListIterator;

import java.util.NoSuchElementException;

import java.util.Objects;

/** A class providing static methods and objects that do useful things with type-specific iterators.

```

*
* @see Iterator
*/

public final class ITERATORS {

    private ITERATORS() {}

    /** A class returning no elements and a type-specific iterator interface.
    *
    * <p>This class may be useful to implement your own in case you subclass
    * a type-specific iterator.
    */

    public static class EmptyIterator KEY_GENERIC implements KEY_LIST_ITERATOR KEY_GENERIC,
    java.io.Serializable, Cloneable {

        private static final long serialVersionUID = -7046029254386353129L;

        protected EmptyIterator() {}

        @Override
        public boolean hasNext() { return false; }

        @Override
        public boolean hasPrevious() { return false; }

        @Override
        public KEY_GENERIC_TYPE NEXT_KEY() { throw new NoSuchElementException(); }

        @Override
        public KEY_GENERIC_TYPE PREV_KEY() { throw new NoSuchElementException(); }

        @Override
        public int nextIndex() { return 0; }

        @Override
        public int previousIndex() { return -1; }

        @Override
        public int skip(int n) { return 0; };

        @Override
        public int back(int n) { return 0; };

        @Override
        public Object clone() { return EMPTY_ITERATOR; }
    }
}

```

```

private Object readResolve() { return EMPTY_ITERATOR; }
}

/** An empty iterator. It is serializable and cloneable.
 *
 * <p>The class of this objects represent an abstract empty iterator
 * that can iterate as a type-specific (list) iterator.
 */

SUPPRESS_WARNINGS_KEY_RAWTYPES
public static final EmptyIterator EMPTY_ITERATOR = new EmptyIterator();

#if KEYS_REFERENCE
/** Returns an empty iterator. It is serializable and cloneable.
 *
 * <p>The class of the object returned represent an abstract empty iterator
 * that can iterate as a type-specific (list) iterator.
 *
 * <p>This method provides a typesafe access to { @link #EMPTY_ITERATOR}.
 * @return an empty iterator.
 */
@SuppressWarnings("unchecked")
public static KEY_GENERIC KEY_ITERATOR KEY_GENERIC emptyIterator() { return EMPTY_ITERATOR; }
#endif

/** An iterator returning a single element. */

private static class SingletonIterator KEY_GENERIC implements KEY_LIST_ITERATOR KEY_GENERIC {
private final KEY_GENERIC_TYPE element;
private int curr;

public SingletonIterator(final KEY_GENERIC_TYPE element) {
this.element = element;
}

@Override
public boolean hasNext() { return curr == 0; }

@Override
public boolean hasPrevious() { return curr == 1; }

@Override
public KEY_GENERIC_TYPE NEXT_KEY() {
if (! hasNext()) throw new NoSuchElementException();
curr = 1;
return element;
}
}

```

```

@Override
public KEY_GENERIC_TYPE PREV_KEY() {
    if (!hasPrevious()) throw new NoSuchElementException();
    curr = 0;
    return element;
}

@Override
public int nextIndex() {
    return curr;
}

@Override
public int previousIndex() {
    return curr - 1;
}
}

/** Returns an immutable iterator that iterates just over the given element.
 *
 * @param element the only element to be returned by a type-specific list iterator.
 * @return an immutable iterator that iterates just over { @code element }.
 */
public static KEY_GENERIC KEY_LIST_ITERATOR KEY_GENERIC singleton(final KEY_GENERIC_TYPE
element) {
    return new SingletonIterator KEY_GENERIC_DIAMOND(element);
}

/** A class to wrap arrays in iterators. */

private static class ArrayIterator KEY_GENERIC implements KEY_LIST_ITERATOR KEY_GENERIC {
    private final KEY_GENERIC_TYPE[] array;
    private final int offset, length;
    private int curr;

    public ArrayIterator(final KEY_GENERIC_TYPE[] array, final int offset, final int length) {
        this.array = array;
        this.offset = offset;
        this.length = length;
    }

    @Override
    public boolean hasNext() { return curr < length; }

    @Override

```

```

public boolean hasPrevious() { return curr > 0; }

@Override
public KEY_GENERIC_TYPE NEXT_KEY() {
    if (! hasNext()) throw new NoSuchElementException();
    return array[offset + curr++];
}

@Override
public KEY_GENERIC_TYPE PREV_KEY() {
    if (! hasPrevious()) throw new NoSuchElementException();
    return array[offset + --curr];
}

@Override
public int skip(int n) {
    if (n <= length - curr) {
        curr += n;
        return n;
    }
    n = length - curr;
    curr = length;
    return n;
}

@Override
public int back(int n) {
    if (n <= curr) {
        curr -= n;
        return n;
    }
    n = curr;
    curr = 0;
    return n;
}

@Override
public int nextIndex() {
    return curr;
}

@Override
public int previousIndex() {
    return curr - 1;
}
}

```

```

/** Wraps the given part of an array into a type-specific list iterator.
 *
 * <p>The type-specific list iterator returned by this method will iterate
 * { @code length } times, returning consecutive elements of the given
 * array starting from the one with index { @code offset }.
 *
 * @param array an array to wrap into a type-specific list iterator.
 * @param offset the first element of the array to be returned.
 * @param length the number of elements to return.
 * @return an iterator that will return { @code length } elements of { @code array } starting at position { @code
offset }.
 */
public static KEY_GENERIC KEY_LIST_ITERATOR KEY_GENERIC wrap(final KEY_GENERIC_TYPE[]
array, final int offset, final int length) {
    ARRAYS.ensureOffsetLength(array, offset, length);
    return new ArrayIterator KEY_GENERIC_DIAMOND(array, offset, length);
}

/** Wraps the given array into a type-specific list iterator.
 *
 * <p>The type-specific list iterator returned by this method will return
 * all elements of the given array.
 *
 * @param array an array to wrap into a type-specific list iterator.
 * @return an iterator that will the elements of { @code array }.
 */
public static KEY_GENERIC KEY_LIST_ITERATOR KEY_GENERIC wrap(final KEY_GENERIC_TYPE[]
array) {
    return new ArrayIterator KEY_GENERIC_DIAMOND(array, 0, array.length);
}

/** Unwraps an iterator into an array starting at a given offset for a given number of elements.
 *
 * <p>This method iterates over the given type-specific iterator and stores the elements
 * returned, up to a maximum of { @code length }, in the given array starting at { @code offset }.
 * The number of actually unwrapped elements is returned (it may be less than { @code max } if
 * the iterator emits less than { @code max } elements).
 *
 * @param i a type-specific iterator.
 * @param array an array to contain the output of the iterator.
 * @param offset the first element of the array to be returned.
 * @param max the maximum number of elements to unwrap.
 * @return the number of elements unwrapped.
 */
public static KEY_GENERIC int unwrap(final STD_KEY_ITERATOR KEY_EXTENDS_GENERIC i, final
KEY_GENERIC_TYPE array[], int offset, final int max) {
    if (max < 0) throw new IllegalArgumentException("The maximum number of elements (" + max + ") is negative");

```

```

if (offset < 0 || offset + max > array.length) throw new IllegalArgumentException();
int j = max;
while(j-- != 0 && i.hasNext()) array[offset++] = i.NEXT_KEY();
return max - j - 1;
}

/** Unwraps an iterator into an array.
 *
 * <p>This method iterates over the given type-specific iterator and stores the
 * elements returned in the given array. The iteration will stop when the
 * iterator has no more elements or when the end of the array has been reached.
 *
 * @param i a type-specific iterator.
 * @param array an array to contain the output of the iterator.
 * @return the number of elements unwrapped.
 */
public static KEY_GENERIC int unwrap(final STD_KEY_ITERATOR KEY_EXTENDS_GENERIC i, final
KEY_GENERIC_TYPE array[]) {
    return unwrap(i, array, 0, array.length);
}

/** Unwraps an iterator, returning an array, with a limit on the number of elements.
 *
 * <p>This method iterates over the given type-specific iterator and returns an array
 * containing the elements returned by the iterator. At most { @code max } elements
 * will be returned.
 *
 * @param i a type-specific iterator.
 * @param max the maximum number of elements to be unwrapped.
 * @return an array containing the elements returned by the iterator (at most { @code max }).
 */
SUPPRESS_WARNINGS_KEY_UNCHECKED
public static KEY_GENERIC KEY_GENERIC_TYPE[] unwrap(final STD_KEY_ITERATOR
KEY_EXTENDS_GENERIC i, int max) {
    if (max < 0) throw new IllegalArgumentException("The maximum number of elements (" + max + ") is negative");
    KEY_GENERIC_TYPE array[] = KEY_GENERIC_ARRAY_CAST new KEY_TYPE[16];
    int j = 0;

    while(max-- != 0 && i.hasNext()) {
        if (j == array.length) array = ARRAYS.grow(array, j + 1);
        array[j++] = i.NEXT_KEY();
    }

    return ARRAYS.trim(array, j);
}

/** Unwraps an iterator, returning an array.

```

```

*
* <p>This method iterates over the given type-specific iterator and returns an array
* containing the elements returned by the iterator.
*
* @param i a type-specific iterator.
* @return an array containing the elements returned by the iterator.
*/

public static KEY_GENERIC KEY_GENERIC_TYPE[] unwrap(final STD_KEY_ITERATOR
KEY_EXTENDS_GENERIC i) {
    return unwrap(i, Integer.MAX_VALUE);
}

/** Unwraps an iterator into a type-specific collection, with a limit on the number of elements.
*
* <p>This method iterates over the given type-specific iterator and stores the elements
* returned, up to a maximum of { @code max }, in the given type-specific collection.
* The number of actually unwrapped elements is returned (it may be less than { @code max } if
* the iterator emits less than { @code max } elements).
*
* @param i a type-specific iterator.
* @param c a type-specific collection array to contain the output of the iterator.
* @param max the maximum number of elements to unwrap.
* @return the number of elements unwrapped. Note that
* this is the number of elements returned by the iterator, which is not necessarily the number
* of elements that have been added to the collection (because of duplicates).
*/

public static KEY_GENERIC int unwrap(final STD_KEY_ITERATOR KEY_GENERIC i, final COLLECTION
KEY_SUPER_GENERIC c, final int max) {
    if (max < 0) throw new IllegalArgumentException("The maximum number of elements (" + max + ") is negative");
    int j = max;
    while(j-- != 0 && i.hasNext()) c.add(i.NEXT_KEY());
    return max - j - 1;
}

/** Unwraps an iterator into a type-specific collection.
*
* <p>This method iterates over the given type-specific iterator and stores the
* elements returned in the given type-specific collection. The returned count on the number
* unwrapped elements is a long, so that it will work also with very large collections.
*
* @param i a type-specific iterator.
* @param c a type-specific collection to contain the output of the iterator.
* @return the number of elements unwrapped. Note that
* this is the number of elements returned by the iterator, which is not necessarily the number
* of elements that have been added to the collection (because of duplicates).
*/

```

```

public static KEY_GENERIC long unwrap(final STD_KEY_ITERATOR KEY_GENERIC i, final COLLECTION
KEY_SUPER_GENERIC c) {
    long n = 0;
    while(i.hasNext()) {
        c.add(i.NEXT_KEY());
        n++;
    }
    return n;
}

```

```

/** Pours an iterator into a type-specific collection, with a limit on the number of elements.

```

```

*

```

```

* <p>This method iterates over the given type-specific iterator and adds

```

```

* the returned elements to the given collection (up to { @code max }).

```

```

*

```

```

* @param i a type-specific iterator.

```

```

* @param s a type-specific collection.

```

```

* @param max the maximum number of elements to be poured.

```

```

* @return the number of elements poured. Note that

```

```

* this is the number of elements returned by the iterator, which is not necessarily the number

```

```

* of elements that have been added to the collection (because of duplicates).

```

```

*/

```

```

public static KEY_GENERIC int pour(final STD_KEY_ITERATOR KEY_GENERIC i, final COLLECTION
KEY_SUPER_GENERIC s, final int max) {
    if (max < 0) throw new IllegalArgumentException("The maximum number of elements (" + max + ") is negative");
    int j = max;
    while(j-- != 0 && i.hasNext()) s.add(i.NEXT_KEY());
    return max - j - 1;
}

```

```

/** Pours an iterator into a type-specific collection.

```

```

*

```

```

* <p>This method iterates over the given type-specific iterator and adds

```

```

* the returned elements to the given collection.

```

```

*

```

```

* @param i a type-specific iterator.

```

```

* @param s a type-specific collection.

```

```

* @return the number of elements poured. Note that

```

```

* this is the number of elements returned by the iterator, which is not necessarily the number

```

```

* of elements that have been added to the collection (because of duplicates).

```

```

*/

```

```

public static KEY_GENERIC int pour(final STD_KEY_ITERATOR KEY_GENERIC i, final COLLECTION
KEY_SUPER_GENERIC s) {
    return pour(i, s, Integer.MAX_VALUE);
}

```

```

/** Pours an iterator, returning a type-specific list, with a limit on the number of elements.
 *
 * <p>This method iterates over the given type-specific iterator and returns
 * a type-specific list containing the returned elements (up to { @code max }). Iteration
 * on the returned list is guaranteed to produce the elements in the same order
 * in which they appeared in the iterator.
 *
 *
 * @param i a type-specific iterator.
 * @param max the maximum number of elements to be poured.
 * @return a type-specific list containing the returned elements, up to { @code max }.
 */

public static KEY_GENERIC LIST KEY_GENERIC pour(final STD_KEY_ITERATOR KEY_GENERIC i, int
max) {
    final ARRAY_LIST KEY_GENERIC l = new ARRAY_LIST KEY_GENERIC_DIAMOND();
    pour(i, l, max);
    l.trim();
    return l;
}

/** Pours an iterator, returning a type-specific list.
 *
 * <p>This method iterates over the given type-specific iterator and returns
 * a list containing the returned elements. Iteration
 * on the returned list is guaranteed to produce the elements in the same order
 * in which they appeared in the iterator.
 *
 *
 * @param i a type-specific iterator.
 * @return a type-specific list containing the returned elements.
 */

public static KEY_GENERIC LIST KEY_GENERIC pour(final STD_KEY_ITERATOR KEY_GENERIC i) {
    return pour(i, Integer.MAX_VALUE);
}

private static class IteratorWrapper KEY_GENERIC implements KEY_ITERATOR KEY_GENERIC {
    final Iterator<KEY_GENERIC_CLASS> i;

    public IteratorWrapper(final Iterator<KEY_GENERIC_CLASS> i) {
        this.i = i;
    }

    @Override
    public boolean hasNext() { return i.hasNext(); }

    @Override
    public void remove() { i.remove(); }
}

```

```

@Override
public KEY_GENERIC_TYPE NEXT_KEY() { return KEY_CLASS2TYPE(i.next()); }
}

/** Wraps a standard iterator into a type-specific iterator.
 *
 * <p>This method wraps a standard iterator into a type-specific one which will handle the
 * type conversions for you. Of course, any attempt to wrap an iterator returning the
 * instances of the wrong class will generate a {@link ClassCastException}. The
 * returned iterator is backed by {@code i}: changes to one of the iterators
 * will affect the other, too.
 *
 * <p>If {@code i} is already type-specific, it will returned and no new object
 * will be generated.
 *
 * @param i an iterator.
 * @return a type-specific iterator backed by {@code i}.
 */
#if KEYS_PRIMITIVE
@SuppressWarnings({"unchecked","rawtypes"})
#endif
public static KEY_GENERIC KEY_ITERATOR KEY_GENERIC AS_KEY_ITERATOR(final Iterator
KEY_GENERIC i) {
    if (i instanceof KEY_ITERATOR) return (KEY_ITERATOR KEY_GENERIC)i;
    return new IteratorWrapper KEY_GENERIC_DIAMOND(i);
}

private static class ListIteratorWrapper KEY_GENERIC implements KEY_LIST_ITERATOR KEY_GENERIC {
    final ListIterator<KEY_GENERIC_CLASS> i;

    public ListIteratorWrapper(final ListIterator<KEY_GENERIC_CLASS> i) {
        this.i = i;
    }

    @Override
    public boolean hasNext() { return i.hasNext(); }

    @Override
    public boolean hasPrevious() { return i.hasPrevious(); }

    @Override
    public int nextIndex() { return i.nextIndex(); }

    @Override
    public int previousIndex() { return i.previousIndex(); }

    @Override

```

```

public void set(KEY_GENERIC_TYPE k) { i.set(KEY2OBJ(k)); }

@Override
public void add(KEY_GENERIC_TYPE k) { i.add(KEY2OBJ(k)); }

@Override
public void remove() { i.remove(); }

@Override
public KEY_GENERIC_TYPE NEXT_KEY() { return KEY_CLASS2TYPE(i.next()); }

@Override
public KEY_GENERIC_TYPE PREV_KEY() { return KEY_CLASS2TYPE(i.previous()); }
}

/** Wraps a standard list iterator into a type-specific list iterator.
 *
 * <p>This method wraps a standard list iterator into a type-specific one
 * which will handle the type conversions for you. Of course, any attempt
 * to wrap an iterator returning the instances of the wrong class will
 * generate a {@link ClassCastException}. The
 * returned iterator is backed by {@code i}: changes to one of the iterators
 * will affect the other, too.
 *
 * <p>If {@code i} is already type-specific, it will returned and no new object
 * will be generated.
 *
 * @param i a list iterator.
 * @return a type-specific list iterator backed by {@code i}.
 */
#if KEYS_PRIMITIVE
    @SuppressWarnings({"unchecked","rawtypes"})
#endif
public static KEY_GENERIC KEY_LIST_ITERATOR KEY_GENERIC AS_KEY_ITERATOR(final ListIterator
KEY_GENERIC i) {
    if (i instanceof KEY_LIST_ITERATOR) return (KEY_LIST_ITERATOR KEY_GENERIC)i;
    return new ListIteratorWrapper KEY_GENERIC_DIAMOND(i);
}

#ifdef JDK_PRIMITIVE_PREDICATE
    public static boolean any(final KEY_ITERATOR iterator, final JDK_PRIMITIVE_PREDICATE predicate) {
#else
    public static KEY_GENERIC boolean any(final KEY_ITERATOR KEY_GENERIC iterator, final
java.util.function.Predicate<? super KEY_GENERIC_CLASS> predicate) {
#endif
    return indexOf(iterator, predicate) != -1;
}

```

```

#ifdef JDK_PRIMITIVE_PREDICATE
    public static boolean all(final KEY_ITERATOR iterator, final JDK_PRIMITIVE_PREDICATE predicate) {
#else
    public static KEY_GENERIC boolean all(final KEY_ITERATOR KEY_GENERIC iterator, final
java.util.function.Predicate<? super KEY_GENERIC_CLASS> predicate) {
#endif
    Objects.requireNonNull(predicate);

    do {
        if (!iterator.hasNext()) return true;
#ifdef KEY_CLASS_Boolean
    } while (predicate.test(Boolean.valueOf(iterator.nextBoolean())));
#else
    } while (predicate.test(iterator.NEXT_KEY()));
#endif
    return false;
    }

#ifdef JDK_PRIMITIVE_PREDICATE
    public static int indexOf(final KEY_ITERATOR iterator, final JDK_PRIMITIVE_PREDICATE predicate) {
#else
    public static KEY_GENERIC int indexOf(final KEY_ITERATOR KEY_GENERIC iterator, final
java.util.function.Predicate<? super KEY_GENERIC_CLASS> predicate) {
#endif
    Objects.requireNonNull(predicate);

    for (int i = 0; iterator.hasNext(); ++i) {
#ifdef KEY_CLASS_Boolean
        if (predicate.test(Boolean.valueOf(iterator.nextBoolean()))) return i;
#else
        if (predicate.test(iterator.NEXT_KEY())) return i;
#endif
    }

    return -1;
    }

#ifdef KEY_CLASS_Integer || KEY_CLASS_Byte || KEY_CLASS_Short || KEY_CLASS_Character ||
KEY_CLASS_Long

#ifdef KEY_CLASS_Long
    private static class IntervalIterator implements KEY_BIDI_ITERATOR {
#else
    private static class IntervalIterator implements KEY_LIST_ITERATOR {
#endif
    private final KEY_TYPE from, to;
    KEY_TYPE curr;

```

```

public IntervalIterator(final KEY_TYPE from, final KEY_TYPE to) {
    this.from = this.curr = from;
    this.to = to;
}

@Override
public boolean hasNext() { return curr < to; }

@Override
public boolean hasPrevious() { return curr > from; }

@Override
public KEY_TYPE NEXT_KEY() {
    if (! hasNext()) throw new NoSuchElementException();
    return curr++;
}

@Override
public KEY_TYPE PREV_KEY() {
    if (! hasPrevious()) throw new NoSuchElementException();
    return --curr;
}

#if ! KEY_CLASS_Long
@Override
public int nextIndex() { return curr - from; }

@Override
public int previousIndex() { return curr - from - 1; }
#endif

@Override
public int skip(int n) {
    if (curr + n <= to) {
        curr += n;
        return n;
    }
#if ! KEY_CLASS_Long
    n = to - curr;
#else
    n = (int)(to - curr);
#endif
    curr = to;
    return n;
}

@Override
public int back(int n) {

```

```

    if (curr - n >= from) {
        curr -= n;
        return n;
    }
#if ! KEY_CLASS_Long
    n = curr - from ;
#else
    n = (int)(curr - from);
#endif
    curr = from;
    return n;
}
}

#if KEY_CLASS_Long
/** Creates a type-specific bidirectional iterator over an interval.
 *
 * <p>The type-specific bidirectional iterator returned by this method will return the
 * elements { @code from}, { @code from+1},&hellip;, { @code to-1}.
 *
 * <p>Note that all other type-specific interval iterator are <em>list</em>
 * iterators. Of course, this is not possible with longs as the index
 * returned by { @link java.util.ListIterator#nextIndex() nextIndex()}/{ @link
 * java.util.ListIterator#previousIndex() previousIndex()} would exceed an integer.
 *
 * @param from the starting element (inclusive).
 * @param to the ending element (exclusive).
 * @return a type-specific bidirectional iterator enumerating the elements from { @code from} to { @code to}.
 */
public static KEY_BIDI_ITERATOR fromTo(final KEY_TYPE from, final KEY_TYPE to) {
    return new IntervalIterator(from, to);
}
#else

/** Creates a type-specific list iterator over an interval.
 *
 * <p>The type-specific list iterator returned by this method will return the
 * elements { @code from}, { @code from+1},&hellip;, { @code to-1}.
 *
 * @param from the starting element (inclusive).
 * @param to the ending element (exclusive).
 * @return a type-specific list iterator enumerating the elements from { @code from} to { @code to}.
 */
public static KEY_LIST_ITERATOR fromTo(final KEY_TYPE from, final KEY_TYPE to) {
    return new IntervalIterator(from, to);
}

#endif

```

```

#endif

private static class IteratorConcatenator KEY_GENERIC implements KEY_ITERATOR KEY_GENERIC {
    final KEY_ITERATOR KEY_EXTENDS_GENERIC a[];
    int offset, length, lastOffset = -1;

    public IteratorConcatenator(final KEY_ITERATOR KEY_EXTENDS_GENERIC a[], int offset, int length) {
        this.a = a;
        this.offset = offset;
        this.length = length;
        advance();
    }

    private void advance() {
        while(length != 0) {
            if (a[offset].hasNext()) break;
            length--;
            offset++;
        }

        return;
    }

    @Override
    public boolean hasNext() {
        return length > 0;
    }

    @Override
    public KEY_GENERIC_TYPE NEXT_KEY() {
        if (! hasNext()) throw new NoSuchElementException();
        KEY_GENERIC_TYPE next = a[lastOffset = offset].NEXT_KEY();
        advance();
        return next;
    }

    @Override
    public void remove() {
        if (lastOffset == -1) throw new IllegalStateException();
        a[lastOffset].remove();
    }

    @Override
    public int skip(int n) {
        lastOffset = -1;

        int skipped = 0;

```

```

while(skipped < n && length != 0) {
    skipped += a[offset].skip(n - skipped);
    if (a[offset].hasNext()) break;
    length--;
    offset++;
}

return skipped;
}
}

```

```

/** Concatenates all iterators contained in an array.

```

```

*

```

```

* <p>This method returns an iterator that will enumerate in order the elements returned

```

```

* by all iterators contained in the given array.

```

```

*

```

```

* @param a an array of iterators.

```

```

* @return an iterator obtained by concatenation.

```

```

*/

```

```

public static KEY_GENERIC KEY_ITERATOR KEY_GENERIC concat(final KEY_ITERATOR
KEY_EXTENDS_GENERIC a[]) {
    return concat(a, 0, a.length);
}

```

```

/** Concatenates a sequence of iterators contained in an array.

```

```

*

```

```

* <p>This method returns an iterator that will enumerate in order the elements returned

```

```

* by { @code a[offset] }, then those returned

```

```

* by { @code a[offset + 1] }, and so on up to

```

```

* { @code a[offset + length - 1] }.

```

```

*

```

```

* @param a an array of iterators.

```

```

* @param offset the index of the first iterator to concatenate.

```

```

* @param length the number of iterators to concatenate.

```

```

* @return an iterator obtained by concatenation of { @code length } elements of { @code a } starting at { @code
offset }.

```

```

*/

```

```

public static KEY_GENERIC KEY_ITERATOR KEY_GENERIC concat(final KEY_ITERATOR
KEY_EXTENDS_GENERIC a[], final int offset, final int length) {
    return new IteratorConcatenator KEY_GENERIC_DIAMOND(a, offset, length);
}

```

```

/** An unmodifiable wrapper class for iterators. */

public static class UnmodifiableIterator KEY_GENERIC implements KEY_ITERATOR KEY_GENERIC {
    protected final KEY_ITERATOR KEY_GENERIC i;

    public UnmodifiableIterator(final KEY_ITERATOR KEY_GENERIC i) {
        this.i = i;
    }

    @Override
    public boolean hasNext() { return i.hasNext(); }

    @Override
    public KEY_GENERIC_TYPE NEXT_KEY() { return i.NEXT_KEY(); }
}

/** Returns an unmodifiable iterator backed by the specified iterator.
 *
 * @param i the iterator to be wrapped in an unmodifiable iterator.
 * @return an unmodifiable view of the specified iterator.
 */
public static KEY_GENERIC KEY_ITERATOR KEY_GENERIC unmodifiable(final KEY_ITERATOR
KEY_GENERIC i) { return new UnmodifiableIterator KEY_GENERIC_DIAMOND(i); }

/** An unmodifiable wrapper class for bidirectional iterators. */

public static class UnmodifiableBidirectionalIterator KEY_GENERIC implements KEY_BIDI_ITERATOR
KEY_GENERIC {
    protected final KEY_BIDI_ITERATOR KEY_GENERIC i;

    public UnmodifiableBidirectionalIterator(final KEY_BIDI_ITERATOR KEY_GENERIC i) {
        this.i = i;
    }

    @Override
    public boolean hasNext() { return i.hasNext(); }

    @Override
    public boolean hasPrevious() { return i.hasPrevious(); }

    @Override
    public KEY_GENERIC_TYPE NEXT_KEY() { return i.NEXT_KEY(); }

    @Override

```

```

public KEY_GENERIC_TYPE PREV_KEY() { return i.PREV_KEY(); }
}

/** Returns an unmodifiable bidirectional iterator backed by the specified bidirectional iterator.
 *
 * @param i the bidirectional iterator to be wrapped in an unmodifiable bidirectional iterator.
 * @return an unmodifiable view of the specified bidirectional iterator.
 */
public static KEY_GENERIC KEY_BIDI_ITERATOR KEY_GENERIC unmodifiable(final
KEY_BIDI_ITERATOR KEY_GENERIC i) { return new UnmodifiableBidirectionalIterator
KEY_GENERIC_DIAMOND(i); }

/** An unmodifiable wrapper class for list iterators. */

public static class UnmodifiableListIterator KEY_GENERIC implements KEY_LIST_ITERATOR
KEY_GENERIC {
protected final KEY_LIST_ITERATOR KEY_GENERIC i;

public UnmodifiableListIterator(final KEY_LIST_ITERATOR KEY_GENERIC i) {
this.i = i;
}

@Override
public boolean hasNext() { return i.hasNext(); }

@Override
public boolean hasPrevious() { return i.hasPrevious(); }

@Override
public KEY_GENERIC_TYPE NEXT_KEY() { return i.NEXT_KEY(); }

@Override
public KEY_GENERIC_TYPE PREV_KEY() { return i.PREV_KEY(); }

@Override
public int nextIndex() { return i.nextIndex(); }

@Override
public int previousIndex() { return i.previousIndex(); }
}

/** Returns an unmodifiable list iterator backed by the specified list iterator.
 *
 * @param i the list iterator to be wrapped in an unmodifiable list iterator.
 * @return an unmodifiable view of the specified list iterator.
 */

```

```

public static KEY_GENERIC KEY_LIST_ITERATOR KEY_GENERIC unmodifiable(final
KEY_LIST_ITERATOR KEY_GENERIC i) { return new UnmodifiableListIterator
KEY_GENERIC_DIAMOND(i); }

#if SMALL_TYPES && (KEY_CLASS_Short || KEY_CLASS_Integer || KEY_CLASS_Long ||
KEY_CLASS_Float || KEY_CLASS_Double)

/** A wrapper promoting the results of a ByteIterator. */

protected static class ByteIteratorWrapper implements KEY_ITERATOR {
final it.unimi.dsi.fastutil.bytes.ByteIterator iterator;

public ByteIteratorWrapper(final it.unimi.dsi.fastutil.bytes.ByteIterator iterator) {
this.iterator = iterator;
}

@Override
public boolean hasNext() { return iterator.hasNext(); }

@Deprecated
@Override
public KEY_GENERIC_CLASS next() { return KEY_GENERIC_CLASS.valueOf(iterator.nextByte()); }

@Override
public KEY_TYPE NEXT_KEY() { return iterator.nextByte(); }

@Override
public void remove() { iterator.remove(); }

@Override
public int skip(final int n) { return iterator.skip(n); }
}

/** Returns an iterator backed by the specified byte iterator.
* @param iterator a byte iterator.
* @return an iterator backed by the specified byte iterator.
*/
public static KEY_ITERATOR wrap(final it.unimi.dsi.fastutil.bytes.ByteIterator iterator) {
return new ByteIteratorWrapper(iterator);
}
#endif

#if SMALL_TYPES && (KEY_CLASS_Integer || KEY_CLASS_Long || KEY_CLASS_Float ||
KEY_CLASS_Double)

/** A wrapper promoting the results of a ShortIterator. */

protected static class ShortIteratorWrapper implements KEY_ITERATOR {

```

```

final it.unimi.dsi.fastutil.shorts.ShortIterator iterator;

public ShortIteratorWrapper(final it.unimi.dsi.fastutil.shorts.ShortIterator iterator) {
    this.iterator = iterator;
}

@Override
public boolean hasNext() { return iterator.hasNext(); }

@Deprecated
@Override
public KEY_GENERIC_CLASS next() { return KEY_GENERIC_CLASS.valueOf(iterator.nextShort()); }

@Override
public KEY_TYPE NEXT_KEY() { return iterator.nextShort(); }

@Override
public void remove() { iterator.remove(); }

@Override
public int skip(final int n) { return iterator.skip(n); }
}

/** Returns an iterator backed by the specified short iterator.
 * @param iterator a short iterator.
 * @return an iterator backed by the specified short iterator.
 */
public static KEY_ITERATOR wrap(final it.unimi.dsi.fastutil.shorts.ShortIterator iterator) {
    return new ShortIteratorWrapper(iterator);
}

#endif

#if KEY_CLASS_Long || KEY_CLASS_Double

/** A wrapper promoting the results of an IntIterator. */

protected static class IntIteratorWrapper implements KEY_ITERATOR {
    final it.unimi.dsi.fastutil.ints.IntIterator iterator;

    public IntIteratorWrapper(final it.unimi.dsi.fastutil.ints.IntIterator iterator) {
        this.iterator = iterator;
    }

    @Override
    public boolean hasNext() { return iterator.hasNext(); }

    @Deprecated

```

```

@Override
public KEY_GENERIC_CLASS next() { return KEY_GENERIC_CLASS.valueOf(iterator.nextInt()); }

@Override
public KEY_TYPE NEXT_KEY() { return iterator.nextInt(); }

@Override
public void remove() { iterator.remove(); }

@Override
public int skip(final int n) { return iterator.skip(n); }
}

/** Returns an iterator backed by the specified integer iterator.
 * @param iterator an integer iterator.
 * @return an iterator backed by the specified integer iterator.
 */

public static KEY_ITERATOR wrap(final it.unimi.dsi.fastutil.ints.IntIterator iterator) {
    return new IntIteratorWrapper(iterator);
}

#endif

#if SMALL_TYPES && KEY_CLASS_Double

/** A wrapper promoting the results of a FloatIterator. */

protected static class FloatIteratorWrapper implements KEY_ITERATOR {
    final it.unimi.dsi.fastutil.floats.FloatIterator iterator;

    public FloatIteratorWrapper(final it.unimi.dsi.fastutil.floats.FloatIterator iterator) {
        this.iterator = iterator;
    }

    @Override
    public boolean hasNext() { return iterator.hasNext(); }

    @Deprecated
    @Override
    public KEY_GENERIC_CLASS next() { return KEY_GENERIC_CLASS.valueOf(iterator.nextFloat()); }

    @Override
    public KEY_TYPE NEXT_KEY() { return iterator.nextFloat(); }

    @Override
    public void remove() { iterator.remove(); }
}

```

```

@Override
public int skip(final int n) { return iterator.skip(n); }
}

/** Returns an iterator backed by the specified float iterator.
 * @param iterator a float iterator.
 * @return an iterator backed by the specified float iterator.
 */
public static KEY_ITERATOR wrap(final it.unimi.dsi.fastutil.floats.FloatIterator iterator) {
    return new FloatIteratorWrapper(iterator);
}
#endif
}

```

Found in path(s):

```

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-
a775574/drv/Iterators.drv

```

No license file was found, but licenses were detected in source scan.

```

/*
 * Copyright (C) 2017 Sebastiano Vigna
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

```

```

package PACKAGE;

```

```

#if KEYS_PRIMITIVE
import java.util.Objects;
#endif
import java.util.function.Consumer;

```

```

/** A type-specific {@link Consumer}; provides methods to consume a primitive type both as object
 * and as primitive.
 *
 * @see Consumer
 * @since 8.0.0

```

```

*/

@FunctionalInterface
#ifdef JDK_PRIMITIVE_KEY_CONSUMER
public interface KEY_CONSUMER KEY_GENERIC extends Consumer<KEY_GENERIC_CLASS>,
JDK_PRIMITIVE_KEY_CONSUMER {
#else
public interface KEY_CONSUMER KEY_GENERIC extends Consumer<KEY_GENERIC_CLASS> {
#endif

#if KEYS_PRIMITIVE
#if !defined JDK_PRIMITIVE_KEY_CONSUMER || KEY_WIDENED
void accept(KEY_TYPE t);
#endif

#if KEY_WIDENED
/** { @inheritDoc }
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default void accept(final KEY_TYPE_WIDENED t) {
accept(KEY_NARROWING(t));
}
#endif

/** { @inheritDoc }
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default void accept(final KEY_CLASS t) {
this.accept(t.KEY_VALUE());
}

#if !defined JDK_PRIMITIVE_KEY_CONSUMER || KEY_WIDENED
default KEY_CONSUMER andThen(final KEY_CONSUMER after) {
Objects.requireNonNull(after);
return (KEY_TYPE t) -> { accept(t); after.accept(t); };
}
#endif

#ifdef JDK_PRIMITIVE_KEY_CONSUMER
#if KEY_WIDENED
/** { @inheritDoc }
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
#endif
@Override
default KEY_CONSUMER andThen(final JDK_PRIMITIVE_KEY_CONSUMER after) {

```

```

Objects.requireNonNull(after);
return (KEY_TYPE t) -> { accept(t); after.accept(t); };
}
#endif

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default Consumer<KEY_GENERIC_CLASS> andThen(final Consumer<? super KEY_GENERIC_CLASS> after)
{
    return Consumer.super.andThen(after);
}
#endif
}

```

Found in path(s):

```

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-
a775574/drv/Consumer.drv

```

No license file was found, but licenses were detected in source scan.

```

/*
 * Copyright (C) 2002-2017 Sebastiano Vigna
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

```

```

package PACKAGE;

```

```

import VALUE_PACKAGE.VALUE_COLLECTION;
import VALUE_PACKAGE.VALUE_ABSTRACT_COLLECTION;
import VALUE_PACKAGE.VALUE_ITERATOR;
import it.unimi.dsi.fastutil.objects.ObjectBidirectionalIterator;

```

```

#if KEYS_REFERENCE
import java.util.Comparator;
#endif

```

```

/** An abstract class providing basic methods for sorted maps implementing a type-specific interface. */

public abstract class ABSTRACT_SORTED_MAP KEY_VALUE_GENERIC extends ABSTRACT_MAP
KEY_VALUE_GENERIC implements SORTED_MAP KEY_VALUE_GENERIC {

    private static final long serialVersionUID = -1773560792952436569L;

    protected ABSTRACT_SORTED_MAP() {}

    /** {@inheritDoc}
     *
     * <p>The view is backed by the sorted set returned by {@link java.util.Map#entrySet()}. Note that
     * <em>no attempt is made at caching the result of this method</em>, as this would
     * require adding some attributes that lightweight implementations would
     * not need. Subclasses may easily override this policy by calling
     * this method and caching the result, but implementors are encouraged to
     * write more efficient ad-hoc implementations.
     *
     * @return a sorted set view of the keys of this map; it may be safely cast to a type-specific interface.
     */
    @Override
    public SORTED_SET KEY_GENERIC keySet() {
        return new KeySet();
    }

    /** A wrapper exhibiting the keys of a map. */

    protected class KeySet extends ABSTRACT_SORTED_SET KEY_GENERIC {
        @Override
        public boolean contains(final KEY_TYPE k) { return containsKey(k); }

        @Override
        public int size() { return ABSTRACT_SORTED_MAP.this.size(); }

        @Override
        public void clear() { ABSTRACT_SORTED_MAP.this.clear(); }

        @Override
        public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() { return
        ABSTRACT_SORTED_MAP.this.comparator(); }

        @Override
        public KEY_GENERIC_TYPE FIRST() { return FIRST_KEY(); }

        @Override
        public KEY_GENERIC_TYPE LAST() { return LAST_KEY(); }
    }

```

```

@Override
public SORTED_SET KEY_GENERIC headSet(final KEY_GENERIC_TYPE to) { return headMap(to).keySet(); }

@Override
public SORTED_SET KEY_GENERIC tailSet(final KEY_GENERIC_TYPE from) { return
tailMap(from).keySet(); }

@Override
public SORTED_SET KEY_GENERIC subSet(final KEY_GENERIC_TYPE from, final KEY_GENERIC_TYPE
to) { return subMap(from, to).keySet(); }

@Override
public KEY_BIDI_ITERATOR KEY_GENERIC iterator(final KEY_GENERIC_TYPE from) { return new
KeySetIterator KEY_VALUE_GENERIC_DIAMOND(ENTRYSET().iterator(new BasicEntry
KEY_VALUE_GENERIC_DIAMOND(from, VALUE_NULL))); }

@Override
public KEY_BIDI_ITERATOR KEY_GENERIC iterator() { return new KeySetIterator
KEY_VALUE_GENERIC_DIAMOND(SORTED_MAPS.fastIterator(ABSTRACT_SORTED_MAP.this)); }
}

/** A wrapper exhibiting a map iterator as an iterator on keys.
 *
 * <p>To provide an iterator on keys, just create an instance of this
 * class using the corresponding iterator on entries.
 */

protected static class KeySetIterator KEY_VALUE_GENERIC implements KEY_BIDI_ITERATOR
KEY_GENERIC {
protected final ObjectBidirectionalIterator<MAP.Entry KEY_VALUE_GENERIC> i;

public KeySetIterator(ObjectBidirectionalIterator<MAP.Entry KEY_VALUE_GENERIC> i) {
this.i = i;
}

@Override
public KEY_GENERIC_TYPE NEXT_KEY() { return i.next().ENTRY_GET_KEY(); };

@Override
public KEY_GENERIC_TYPE PREV_KEY() { return i.previous().ENTRY_GET_KEY(); };

@Override
public boolean hasNext() { return i.hasNext(); }

@Override
public boolean hasPrevious() { return i.hasPrevious(); }
}

```

```

/** {@inheritDoc}
 *
 * <p>The view is backed by the sorted set returned by {@link java.util.Map#entrySet()}. Note that
 * <em>no attempt is made at caching the result of this method</em>, as this would
 * require adding some attributes that lightweight implementations would
 * not need. Subclasses may easily override this policy by calling
 * this method and caching the result, but implementors are encouraged to
 * write more efficient ad-hoc implementations.
 *
 * @return a type-specific collection view of the values contained in this map.
 */
@Override
public VALUE_COLLECTION VALUE_GENERIC values() {
    return new ValuesCollection();
}

/** A wrapper exhibiting the values of a map. */
protected class ValuesCollection extends VALUE_ABSTRACT_COLLECTION VALUE_GENERIC {
    @Override
    public VALUE_ITERATOR VALUE_GENERIC iterator() { return new ValuesIterator
    KEY_VALUE_GENERIC_DIAMOND(SORTED_MAPS.fastIterator(ABSTRACT_SORTED_MAP.this)); }

    @Override
    public boolean contains(final VALUE_TYPE k) { return containsValue(k); }

    @Override
    public int size() { return ABSTRACT_SORTED_MAP.this.size(); }

    @Override
    public void clear() { ABSTRACT_SORTED_MAP.this.clear(); }
}

/** A wrapper exhibiting a map iterator as an iterator on values.
 *
 * <p>To provide an iterator on values, just create an instance of this
 * class using the corresponding iterator on entries.
 */

protected static class ValuesIterator KEY_VALUE_GENERIC implements VALUE_ITERATOR
VALUE_GENERIC {
    protected final ObjectBidirectionalIterator<MAP.Entry KEY_VALUE_GENERIC> i;

    public ValuesIterator(ObjectBidirectionalIterator<MAP.Entry KEY_VALUE_GENERIC> i) {
        this.i = i;
    }

    @Override
    public VALUE_GENERIC_TYPE NEXT_VALUE() { return i.next().ENTRY_GET_VALUE(); };
}

```

```
@Override
public boolean hasNext() { return i.hasNext(); }
}
}
```

Found in path(s):

```
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-
a775574/drv/AbstractSortedMap.drv
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2003-2017 Paolo Boldi and Sebastiano Vigna
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

```
package PACKAGE;
```

```
#if KEY_CLASS_Object
import java.util.Arrays;
import java.util.Comparator;
```

```
import it.unimi.dsi.fastutil.PriorityQueue;
#else
import java.util.Iterator;
#endif
```

```
import java.util.Collection;
import java.util.NoSuchElementException;
```

```
/** A type-specific heap-based priority queue.
 *
 * <p>Instances of this class represent a priority queue using a heap. The heap is enlarged as needed, but
 * it is never shrunk. Use the {@link #trim()} method to reduce its size, if necessary.
 */
```

```

public class HEAP_PRIORITY_QUEUE KEY_GENERIC implements PRIORITY_QUEUE KEY_GENERIC,
java.io.Serializable {
    private static final long serialVersionUID = 1L;

    /** The heap array. */
    SUPPRESS_WARNINGS_KEY_UNCHECKED
    protected transient KEY_GENERIC_TYPE[] heap = KEY_GENERIC_ARRAY_CAST
    ARRAYS.EMPTY_ARRAY;

    /** The number of elements in this queue. */
    protected int size;

    /** The type-specific comparator used in this queue. */
    protected KEY_COMPARATOR KEY_SUPER_GENERIC c;

    /** Creates a new empty queue with a given capacity and comparator.
    *
    * @param capacity the initial capacity of this queue.
    * @param c the comparator used in this queue, or { @code null } for the natural order.
    */
    SUPPRESS_WARNINGS_KEY_UNCHECKED
    public HEAP_PRIORITY_QUEUE(int capacity, KEY_COMPARATOR KEY_SUPER_GENERIC c) {
        if (capacity > 0) this.heap = KEY_GENERIC_ARRAY_CAST new KEY_TYPE[capacity];
        this.c = c;
    }

    /** Creates a new empty queue with a given capacity and using the natural order.
    *
    * @param capacity the initial capacity of this queue.
    */
    public HEAP_PRIORITY_QUEUE(int capacity) {
        this(capacity, null);
    }

    /** Creates a new empty queue with a given comparator.
    *
    * @param c the comparator used in this queue, or { @code null } for the natural order.
    */
    public HEAP_PRIORITY_QUEUE(KEY_COMPARATOR KEY_SUPER_GENERIC c) {
        this(0, c);
    }

    /** Creates a new empty queue using the natural order.
    */
    public HEAP_PRIORITY_QUEUE() {
        this(0, null);
    }
}

```

```

/** Wraps a given array in a queue using a given comparator.
 *
 * <p>The queue returned by this method will be backed by the given array.
 * The first { @code size } element of the array will be rearranged so to form a heap (this is
 * more efficient than enqueueing the elements of { @code a } one by one).
 *
 * @param a an array.
 * @param size the number of elements to be included in the queue.
 * @param c the comparator used in this queue, or { @code null } for the natural order.
 */
public HEAP_PRIORITY_QUEUE(final KEY_GENERIC_TYPE[] a, int size, final KEY_COMPARATOR
KEY_SUPER_GENERIC c) {
    this(c);
    this.heap = a;
    this.size = size;
    HEAPS.makeHeap(a, size, c);
}

```

```

/** Wraps a given array in a queue using a given comparator.
 *
 * <p>The queue returned by this method will be backed by the given array.
 * The elements of the array will be rearranged so to form a heap (this is
 * more efficient than enqueueing the elements of { @code a } one by one).
 *
 * @param a an array.
 * @param c the comparator used in this queue, or { @code null } for the natural order.
 */
public HEAP_PRIORITY_QUEUE(final KEY_GENERIC_TYPE[] a, final KEY_COMPARATOR
KEY_SUPER_GENERIC c) {
    this(a, a.length, c);
}

```

```

/** Wraps a given array in a queue using the natural order.
 *
 * <p>The queue returned by this method will be backed by the given array.
 * The first { @code size } element of the array will be rearranged so to form a heap (this is
 * more efficient than enqueueing the elements of { @code a } one by one).
 *
 * @param a an array.
 * @param size the number of elements to be included in the queue.
 */
public HEAP_PRIORITY_QUEUE(final KEY_GENERIC_TYPE[] a, int size) {
    this(a, size, null);
}

```

```

/** Wraps a given array in a queue using the natural order.
 *
 * <p>The queue returned by this method will be backed by the given array.
 * The elements of the array will be rearranged so to form a heap (this is
 * more efficient than enqueueing the elements of { @code a } one by one).
 *
 * @param a an array.
 */
public HEAP_PRIORITY_QUEUE(final KEY_GENERIC_TYPE[] a) {
    this(a, a.length);
}

#if KEYS_PRIMITIVE

/** Creates a queue using the elements in a type-specific collection using a given comparator.
 *
 * <p>This constructor is more efficient than enqueueing the elements of { @code collection } one by one.
 *
 * @param collection a collection; its elements will be used to initialize the queue.
 * @param c the comparator used in this queue, or { @code null } for the natural order.
 */
public HEAP_PRIORITY_QUEUE(final COLLECTION KEY_EXTENDS_GENERIC collection, final
KEY_COMPARATOR KEY_SUPER_GENERIC c) {
    this(collection.TO_KEY_ARRAY(), c);
}

/** Creates a queue using the elements in a type-specific collection using the natural order.
 *
 * <p>This constructor is
 * more efficient than enqueueing the elements of { @code collection } one by one.
 *
 * @param collection a collection; its elements will be used to initialize the queue.
 */
public HEAP_PRIORITY_QUEUE(final COLLECTION KEY_EXTENDS_GENERIC collection) {
    this(collection, null);
}

/** Creates a queue using the elements in a collection using a given comparator.
 *
 * <p>This constructor is more efficient than enqueueing the elements of { @code collection } one by one.
 *
 * @param collection a collection; its elements will be used to initialize the queue.
 * @param c the comparator used in this queue, or { @code null } for the natural order.
 */
public HEAP_PRIORITY_QUEUE(final Collection<? extends KEY_GENERIC_CLASS> collection, final
KEY_COMPARATOR KEY_SUPER_GENERIC c) {
    this(collection.size(), c);
    final Iterator<? extends KEY_GENERIC_CLASS> iterator = collection.iterator();

```

```

final int size = collection.size();
for(int i = 0 ; i < size; i++) heap[i] = KEY_OBJ2TYPE(iterator.next());
}

/** Creates a queue using the elements in a collection using the natural order.
 *
 * <p>This constructor is
 * more efficient than enqueueing the elements of { @code collection } one by one.
 *
 * @param collection a collection; its elements will be used to initialize the queue.
 */
public HEAP_PRIORITY_QUEUE(final Collection<? extends KEY_GENERIC_CLASS> collection) {
    this(collection, null);
}
#else
/** Creates a queue using the elements in a collection using a given comparator.
 *
 * <p>This constructor is more efficient than enqueueing the elements of { @code collection } one by one.
 *
 * @param collection a collection; its elements will be used to initialize the queue.
 * @param c the comparator used in this queue, or { @code null } for the natural order.
 */
SUPPRESS_WARNINGS_KEY_UNCHECKED
public HEAP_PRIORITY_QUEUE(final Collection<? extends KEY_GENERIC_CLASS> collection, final
KEY_COMPARATOR KEY_SUPER_GENERIC c) {
    this(KEY_GENERIC_ARRAY_CAST collection.toArray(), c);
}

/** Creates a queue using the elements in a collection using the natural order.
 *
 * <p>This constructor is
 * more efficient than enqueueing the elements of { @code collection } one by one.
 *
 * @param collection a collection; its elements will be used to initialize the queue.
 */
public HEAP_PRIORITY_QUEUE(final Collection<? extends KEY_GENERIC_CLASS> collection) {
    this(collection, null);
}
#endif

@Override
public void enqueue(KEY_GENERIC_TYPE x) {
    if (size == heap.length) heap = ARRAYS.grow(heap, size + 1);

    heap[size++] = x;
    HEAPS.upHeap(heap, size, size - 1, c);
}

```

```

@Override
public KEY_GENERIC_TYPE DEQUEUE() {
    if (size == 0) throw new NoSuchElementException();

    final KEY_GENERIC_TYPE result = heap[0];
    heap[0] = heap[--size];
    #if KEY_CLASS_Object
        heap[size] = null;
    #endif
    if (size != 0) HEAPS.downHeap(heap, size, 0, c);
    return result;
}

@Override
public KEY_GENERIC_TYPE FIRST() {
    if (size == 0) throw new NoSuchElementException();
    return heap[0];
}

@Override
public void changed() {
    HEAPS.downHeap(heap, size, 0, c);
}

@Override
public int size() { return size; }

@Override
public void clear() {
    #if KEY_CLASS_Object
        Arrays.fill(heap, 0, size, null);
    #endif
    size = 0;
}

/** Trims the underlying heap array so that it has exactly {@link #size()} elements. */

public void trim() {
    heap = ARRAYS.trim(heap, size);
}

@Override
public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() { return c; }

private void writeObject(java.io.ObjectOutputStream s) throws java.io.IOException {
    s.defaultWriteObject();
    s.writeInt(heap.length);
    for(int i = 0; i < size; i++) s.WRITE_KEY(heap[i]);
}

```

```

}

SUPPRESS_WARNINGS_KEY_UNCHECKED
private void readObject(java.io.ObjectInputStream s) throws java.io.IOException, ClassNotFoundException {
    s.defaultReadObject();
    heap = KEY_GENERIC_ARRAY_CAST new KEY_TYPE[s.readInt()];
    for(int i = 0; i < size; i++) heap[i] = KEY_GENERIC_CAST s.READ_KEY();
}

#ifdef TEST

private static long seed = System.currentTimeMillis();
private static java.util.Random r = new java.util.Random(seed);

private static KEY_TYPE genKey() {
#ifdef KEY_CLASS_Byte || KEY_CLASS_Short || KEY_CLASS_Character
    return (KEY_TYPE)(r.nextInt());
#elif KEYS_PRIMITIVE
    return r.NEXT_KEY();
#elif KEY_CLASS_Object
    return Integer.toBinaryString(r.nextInt());
#else
    return new java.io.Serializable() {};
#endif
}

private static java.text.NumberFormat format = new java.text.DecimalFormat("#,###.00");
private static java.text.FieldPosition p = new java.text.FieldPosition(0);

private static String format(double d) {
    StringBuffer s = new StringBuffer();
    return format.format(d, s, p).toString();
}

private static void speedTest(int n, boolean comp) {
    System.out.println("There are presently no speed tests for this class.");
}

private static void fatal(String msg) {
    System.out.println(msg);
    System.exit(1);
}

private static void ensure(boolean cond, String msg) {
    if (cond) return;
    fatal(msg);
}

```

```

}

private static boolean heapEqual(KEY_TYPE[] a, KEY_TYPE[] b, int sizea, int sizeb) {
    if (sizea != sizeb) return false;
    KEY_TYPE[] aa = (KEY_TYPE[])a.clone();
    KEY_TYPE[] bb = (KEY_TYPE[])b.clone();
    java.util.Arrays.sort(aa, 0, sizea);
    java.util.Arrays.sort(bb, 0, sizeb);
    while(sizea-- != 0) if (! KEY_EQUALS(aa[sizea], bb[sizea])) return false;
    return true;
}

private static KEY_TYPE k[];

protected static void runTest(int n) {
    long ms;
    Exception mThrowsIllegal, tThrowsIllegal, mThrowsOutOfBounds, tThrowsOutOfBounds, mThrowsNoElement,
    tThrowsNoElement;
    KEY_TYPE rm = KEY_NULL, rt = KEY_NULL;
    k = new KEY_TYPE[n];

    for(int i = 0; i < n; i++) k[i] = genKey();

    HEAP_PRIORITY_QUEUE m = new
    HEAP_PRIORITY_QUEUE(COMPARATORS.NATURAL_COMPARATOR);
    ARRAY_PRIORITY_QUEUE t = new
    ARRAY_PRIORITY_QUEUE(COMPARATORS.NATURAL_COMPARATOR);

    /* We add pairs to t. */
    for(int i = 0; i < n / 2; i++) {
        t.enqueue(k[i]);
        m.enqueue(k[i]);
    }

    ensure(heapEqual(m.heap, t.array, m.size(), t.size()), "Error (" + seed + "): m and t differ after creation (" + m + ", "
    + t + ")");

    if (m.size() != 0) {
        ensure(KEY_EQUALS(m.FIRST(), t.FIRST()), "Error (" + seed + "): m and t differ in first element after creation ("
        + m.FIRST() + ", " + t.FIRST() + ")");
    }

    /* Now we add and remove random data in m and t, checking that the result is the same. */

    for(int i=0; i<2*n; i++) {

        if (r.nextDouble() < 0.01) {
            t.clear();

```

```

m.clear();
for(int j = 0; j < n / 2; j++) {
    t.enqueue(k[j]);
    m.enqueue(k[j]);
}
}

KEY_TYPE T = genKey();

mThrowsNoElement = tThrowsNoElement = mThrowsOutOfBounds = tThrowsOutOfBounds = mThrowsIllegal =
tThrowsIllegal = null;

try {
    m.enqueue(T);
}
catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }
catch (IllegalArgumentException e) { mThrowsIllegal = e; }

try {
    t.enqueue(T);
}
catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }
catch (IllegalArgumentException e) { tThrowsIllegal = e; }

ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + seed + "): enqueue()
divergence in IndexOutOfBoundsException for " + T + " (" + mThrowsOutOfBounds + ", " + tThrowsOutOfBounds
+ ")");
ensure((mThrowsIllegal == null) == (tThrowsIllegal == null), "Error (" + seed + "): enqueue() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ")");

ensure(heapEqual(m.heap, t.array, m.size(), t.size()), "Error (" + seed + "): m and t differ after enqueue (" + m + ", "
+ t + ")");

if (m.size() != 0) {
    ensure(KEY_EQUALS(m.FIRST(), t.FIRST()), "Error (" + seed + "): m and t differ in first element after enqueue ("
+ m.FIRST() + ", " + t.FIRST() + ")");
}

mThrowsNoElement = tThrowsNoElement = mThrowsOutOfBounds = tThrowsOutOfBounds = mThrowsIllegal =
tThrowsIllegal = null;

try {
    rm = m.DEQUEUE();
}
catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }
catch (IllegalArgumentException e) { mThrowsIllegal = e; }
catch (NoSuchElementException e) { mThrowsNoElement = e; }

```

```

try {
    rt = t.DEQUEUE();
}
catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }
catch (IllegalArgumentException e) { tThrowsIllegal = e; }
catch (NoSuchElementException e) { tThrowsNoElement = e; }

ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + seed + "): dequeue()
divergence in IndexOutOfBoundsException (" + mThrowsOutOfBounds + ", " + tThrowsOutOfBounds + ")");
ensure((mThrowsIllegal == null) == (tThrowsIllegal == null), "Error (" + seed + "): dequeue() divergence in
IllegalArgumentException (" + mThrowsIllegal + ", " + tThrowsIllegal + ")");
ensure((mThrowsNoElement == null) == (tThrowsNoElement == null), "Error (" + seed + "): dequeue() divergence
in NoSuchElementException (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");
if (mThrowsOutOfBounds == null) ensure(KEY_EQUALS(rt, rm), "Error (" + seed + "): divergence in dequeue()
between t and m (" + rt + ", " + rm + ")");

ensure(heapEqual(m.heap, t.array, m.size(), t.size()), "Error (" + seed + "): m and t differ after dequeue (" + m + ", "
+ t + ")");

if (m.size() != 0) {
    ensure(KEY_EQUALS(m.FIRST(), t.FIRST()), "Error (" + seed + "): m and t differ in first element after dequeue ("
+ m.FIRST() + ", " + t.FIRST() + ")");
}

/* Now we save and read m. */

try {
    java.io.File ff = new java.io.File("it.unimi.dsi.fastutil.test");
    java.io.OutputStream os = new java.io.FileOutputStream(ff);
    java.io.ObjectOutputStream oos = new java.io.ObjectOutputStream(os);

    oos.writeObject(m);
    oos.close();

    java.io.InputStream is = new java.io.FileInputStream(ff);
    java.io.ObjectInputStream ois = new java.io.ObjectInputStream(is);

    m = (HEAP_PRIORITY_QUEUE)ois.readObject();
    ois.close();
    ff.delete();
}
catch (Exception e) {
    e.printStackTrace();
    System.exit(1);
}

ensure(heapEqual(m.heap, t.array, m.size(), t.size()), "Error (" + seed + "): m and t differ after save/read");

```

```

HEAP_PRIORITY_QUEUE m2 = new HEAP_PRIORITY_QUEUE(t.array, t.size());
ARRAY_PRIORITY_QUEUE t2 = new ARRAY_PRIORITY_QUEUE(m.heap, m.size());
m = m2;
t = t2;

ensure(heapEqual(m.heap, t.array, m.size(), t.size()), "Error (" + seed + "): m and t differ after wrap (" + m + ", " + t
+ ")");

if (m.size() != 0) {
    ensure(KEY_EQUALS(m.FIRST(), t.FIRST()), "Error (" + seed + "): m and t differ in first element after wrap (" +
m.FIRST() + ", " + t.FIRST() + ")");
}

if (m.size() != 0 && ((new OPEN_HASH_SET(m.heap, 0, m.size())).size() == m.size())) {

    int j = t.size(), M = --j;
#ifdef KEYS_PRIMITIVE
    while(j-- != 0) if (KEY_LESS(t.array[j], t.array[M])) M = j;
#else
    while(j-- != 0) if (((Comparable)t.array[j]).compareTo(t.array[M]) < 0) M = j;
#endif

    m.heap[0] = t.array[M] = genKey();

    m.changed();
    t.changed();

    ensure(heapEqual(m.heap, t.array, m.size(), t.size()), "Error (" + seed + "): m and t differ after change (" + m + ", " +
t + ")");

    if (m.size() != 0) {
        ensure(KEY_EQUALS(m.FIRST(), t.FIRST()), "Error (" + seed + "): m and t differ in first element after change ("
+ m.FIRST() + ", " + t.FIRST() + ")");
    }
}

/* Now we check that m actually holds the same data. */

m.clear();
ensure(m.isEmpty(), "Error (" + seed + "): m is not empty after clear()");

System.out.println("Test OK");
}

```

```

public static void main(String args[]) {
    int n = Integer.parseInt(args[1]);
    if (args.length > 2) r = new java.util.Random(seed = Long.parseLong(args[2]));

    try {
        if ("speedTest".equals(args[0]) || "speedComp".equals(args[0])) speedTest(n, "speedComp".equals(args[0]));
        else if ("test".equals(args[0])) runTest(n);
    } catch(Throwable e) {
        e.printStackTrace(System.err);
        System.err.println("seed: " + seed);
    }
}

#endif

}

```

Found in path(s):

```
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-
a775574/drv/HeapPriorityQueue.drv
```

No license file was found, but licenses were detected in source scan.

```
/*
```

```
* Copyright (C) 2009-2017 Sebastiano Vigna
```

```
*
```

```
* Licensed under the Apache License, Version 2.0 (the "License");
```

```
* you may not use this file except in compliance with the License.
```

```
* You may obtain a copy of the License at
```

```
*
```

```
* http://www.apache.org/licenses/LICENSE-2.0
```

```
*
```

```
* Unless required by applicable law or agreed to in writing, software
```

```
* distributed under the License is distributed on an "AS IS" BASIS,
```

```
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
```

```
* See the License for the specific language governing permissions and
```

```
* limitations under the License.
```

```
*
```

```
*
```

```
*
```

```
* Copyright (C) 1999 CERN - European Organization for Nuclear Research.
```

```
*
```

```
* Permission to use, copy, modify, distribute and sell this software and
```

```
* its documentation for any purpose is hereby granted without fee,
```

```
* provided that the above copyright notice appear in all copies and that
```

```
* both that copyright notice and this permission notice appear in
```

```
* supporting documentation. CERN makes no representations about the
```

```

* suitability of this software for any purpose. It is provided "as is"
* without expressed or implied warranty.
*/

package PACKAGE;

import java.util.Arrays;
import java.util.Random;

import it.unimi.dsi.fastutil.BigArrays;
import it.unimi.dsi.fastutil.Hash;
import static it.unimi.dsi.fastutil.BigArrays.ensureLength;
import static it.unimi.dsi.fastutil.BigArrays.start;
import static it.unimi.dsi.fastutil.BigArrays.segment;
import static it.unimi.dsi.fastutil.BigArrays.displacement;
import static it.unimi.dsi.fastutil.BigArrays.SEGMENT_MASK;
import static it.unimi.dsi.fastutil.BigArrays.SEGMENT_SHIFT;
import static it.unimi.dsi.fastutil.BigArrays.SEGMENT_SIZE;

#if KEYS_PRIMITIVE

#if ! KEY_CLASS_Byte && ! KEY_CLASS_Boolean
import it.unimi.dsi.fastutil.bytes.ByteBigArrays;
#endif

/** A class providing static methods and objects that do useful things with {@linkplain BigArrays big arrays}.
 *
 * <p>In particular, the {@code forceCapacity()}, {@code ensureCapacity()}, {@code grow()},
 * {@code trim()} and {@code setLength()} methods allow to handle
 * big arrays much like array lists.
 *
 * <p>Note that {@link it.unimi.dsi.fastutil.io.BinIO} and {@link it.unimi.dsi.fastutil.io.TextIO}
 * contain several methods that make it possible to load and save big arrays of primitive types as sequences
 * of elements in {@link java.io.DataInput} format (i.e., not as objects) or as sequences of lines of text.
 *
 * @see BigArrays
 */

public final class BIG_ARRAYS {

#else

import java.util.Comparator;

/** A class providing static methods and objects that do useful things with {@linkplain BigArrays big arrays}.
 *
 * <p>In particular, the {@code ensureCapacity()}, {@code grow()},
 * {@code trim()} and {@code setLength()} methods allow to handle

```

* arrays much like array lists.
 *
 * <p>Note that { @link it.unimi.dsi.fastutil.io.BinIO } and { @link it.unimi.dsi.fastutil.io.TextIO }
 * contain several methods make it possible to load and save big arrays of primitive types as sequences
 * of elements in { @link java.io.DataInput } format (i.e., not as objects) or as sequences of lines of text.
 *
 * <p>Warning: creating arrays
 * using { @linkplain java.lang.reflect.Array#newInstance(Class,int) reflection }, as it
 * happens in { @link #ensureCapacity(Object[][],long,long) } and { @link #grow(Object[][],long,long) },
 * is significantly slower than using { @code new }. This phenomenon is particularly
 * evident in the first growth phases of an array reallocated with doubling (or similar) logic.
 *
 * @see BigArrays
 */

```
public final class BIG_ARRAYS {

#endif
private BIG_ARRAYS() {}

/** A static, final, empty big array. */
public static final KEY_TYPE[][] EMPTY_BIG_ARRAY = {};

/** A static, final, empty big array to be used as default big array in allocations. An
 * object distinct from { @link #EMPTY_BIG_ARRAY } makes it possible to have different
 * behaviors depending on whether the user required an empty allocation, or we are
 * just lazily delaying allocation.
 *
 * @see java.util.ArrayList
 */
public static final KEY_TYPE[][] DEFAULT_EMPTY_BIG_ARRAY = {};

/** Returns the element of the given big array of specified index.
 *
 * @param array a big array.
 * @param index a position in the big array.
 * @return the element of the big array at the specified position.
 */
public static KEY_GENERIC KEY_GENERIC_TYPE get(final KEY_GENERIC_TYPE[][] array, final long index)
{
return array[segment(index)][displacement(index)];
}

/** Sets the element of the given big array of specified index.
 *
 * @param array a big array.
 * @param index a position in the big array.
 * @param value the new value for the array element at the specified position.
 */
```

```

*/
public static KEY_GENERIC void set(final KEY_GENERIC_TYPE[][] array, final long index,
KEY_GENERIC_TYPE value) {
    array[segment(index)][displacement(index)] = value;
}

/** Swaps the element of the given big array of specified indices.
*
* @param array a big array.
* @param first a position in the big array.
* @param second a position in the big array.
*/
public static KEY_GENERIC void swap(final KEY_GENERIC_TYPE[][] array, final long first, final long second)
{
    final KEY_GENERIC_TYPE t = array[segment(first)][displacement(first)];
    array[segment(first)][displacement(first)] = array[segment(second)][displacement(second)];
    array[segment(second)][displacement(second)] = t;
}

#if KEYS_PRIMITIVE && !KEY_CLASS_Boolean
/** Adds the specified increment the element of the given big array of specified index.
*
* @param array a big array.
* @param index a position in the big array.
* @param incr the increment
*/
public static void add(final KEY_GENERIC_TYPE[][] array, final long index, KEY_GENERIC_TYPE incr) {
    array[segment(index)][displacement(index)] += incr;
}

/** Multiplies by the specified factor the element of the given big array of specified index.
*
* @param array a big array.
* @param index a position in the big array.
* @param factor the factor
*/
public static void mul(final KEY_GENERIC_TYPE[][] array, final long index, KEY_GENERIC_TYPE factor) {
    array[segment(index)][displacement(index)] *= factor;
}

/** Increments the element of the given big array of specified index.
*
* @param array a big array.
* @param index a position in the big array.
*/
public static void incr(final KEY_GENERIC_TYPE[][] array, final long index) {
    array[segment(index)][displacement(index)]++;
}

```

```

/** Decrements the element of the given big array of specified index.
 *
 * @param array a big array.
 * @param index a position in the big array.
 */
public static void decr(final KEY_GENERIC_TYPE[][] array, final long index) {
    array[segment(index)][displacement(index)]--;
}

#endif

/** Returns the length of the given big array.
 *
 * @param array a big array.
 * @return the length of the given big array.
 */
public static KEY_GENERIC long length(final KEY_GENERIC_TYPE[][] array) {
    final int length = array.length;
    return length == 0 ? 0 : start(length - 1) + array[length - 1].length;
}

/** Copies a big array from the specified source big array, beginning at the specified position, to the specified
position of the destination big array.
 * Handles correctly overlapping regions of the same big array.
 *
 * @param srcArray the source big array.
 * @param srcPos the starting position in the source big array.
 * @param destArray the destination big array.
 * @param destPos the starting position in the destination data.
 * @param length the number of elements to be copied.
 */
public static KEY_GENERIC void copy(final KEY_GENERIC_TYPE[][] srcArray, final long srcPos, final
KEY_GENERIC_TYPE[][] destArray, final long destPos, long length) {
    if (destPos <= srcPos) {
        int srcSegment = segment(srcPos);
        int destSegment = segment(destPos);
        int srcDispl = displacement(srcPos);
        int destDispl = displacement(destPos);
        int l;
        while (length > 0) {
            l = (int) Math.min(length, Math.min(srcArray[srcSegment].length - srcDispl, destArray[destSegment].length -
destDispl));
            System.arraycopy(srcArray[srcSegment], srcDispl, destArray[destSegment], destDispl, l);
            if ((srcDispl += l) == SEGMENT_SIZE) {
                srcDispl = 0;
            }
        }
    }
}

```

```

    srcSegment++;
}
if ((destDispl += 1) == SEGMENT_SIZE) {
    destDispl = 0;
    destSegment++;
}
length -= 1;
}
}
else {
    int srcSegment = segment(srcPos + length);
    int destSegment = segment(destPos + length);
    int srcDispl = displacement(srcPos + length);
    int destDispl = displacement(destPos + length);
    int l;
    while(length > 0) {
        if (srcDispl == 0) {
            srcDispl = SEGMENT_SIZE;
            srcSegment--;
        }
        if (destDispl == 0) {
            destDispl = SEGMENT_SIZE;
            destSegment--;
        }
        l = (int)Math.min(length, Math.min(srcDispl, destDispl));
        System.arraycopy(srcArray[srcSegment], srcDispl - l, destArray[destSegment], destDispl - l, l);
        srcDispl -= l;
        destDispl -= l;
        length -= l;
    }
}
}

/** Copies a big array from the specified source big array, beginning at the specified position, to the specified
position of the destination array.
*
* @param srcArray the source big array.
* @param srcPos the starting position in the source big array.
* @param destArray the destination array.
* @param destPos the starting position in the destination data.
* @param length the number of elements to be copied.
*/
public static KEY_GENERIC void copyFromBig(final KEY_GENERIC_TYPE[][] srcArray, final long srcPos, final
KEY_GENERIC_TYPE[] destArray, int destPos, int length) {
    int srcSegment = segment(srcPos);
    int srcDispl = displacement(srcPos);
    int l;
    while(length > 0) {

```

```

    l = Math.min(srcArray[srcSegment].length - srcDispl, length);
    System.arraycopy(srcArray[srcSegment], srcDispl, destArray, destPos, l);
    if ((srcDispl += l) == SEGMENT_SIZE) {
        srcDispl = 0;
        srcSegment++;
    }
    destPos += l;
    length -= l;
}
}

/** Copies an array from the specified source array, beginning at the specified position, to the specified position of
the destination big array.
*
* @param srcArray the source array.
* @param srcPos the starting position in the source array.
* @param destArray the destination big array.
* @param destPos the starting position in the destination data.
* @param length the number of elements to be copied.
*/
public static KEY_GENERIC void copyToBig(final KEY_GENERIC_TYPE[] srcArray, int srcPos, final
KEY_GENERIC_TYPE[][] destArray, final long destPos, long length) {
    int destSegment = segment(destPos);
    int destDispl = displacement(destPos);
    int l;
    while(length > 0) {
        l = (int)Math.min(destArray[destSegment].length - destDispl, length);
        System.arraycopy(srcArray, srcPos, destArray[destSegment], destDispl, l);
        if ((destDispl += l) == SEGMENT_SIZE) {
            destDispl = 0;
            destSegment++;
        }
        srcPos += l;
        length -= l;
    }
}

#if KEY_CLASS_Object
/** Creates a new big array using the given one as prototype.
*
* <p>This method returns a new big array of the given length whose element
* are of the same class as of those of { @code prototype}. In case
* of an empty big array, it tries to return { @link #EMPTY_BIG_ARRAY}, if possible.
*
* @param prototype a big array that will be used to type the new one.
* @param length the length of the new big array.
* @return a new big array of given type and length.
*/

```

```

SUPPRESS_WARNINGS_KEY_UNCHECKED
public static <K> K[][] newBigArray(final K[][] prototype, final long length) {
    return (K[][])newBigArray(prototype.getClass().getComponentType(), length);
}

/** Creates a new big array using a the given one as component type.
 *
 * <p>This method returns a new big array whose segments
 * are of class { @code componentType}. In case
 * of an empty big array, it tries to return { @link #EMPTY_BIG_ARRAY}, if possible.
 *
 * @param componentType a class representing the type of segments of the array to be created.
 * @param length the length of the new big array.
 * @return a new big array of given type and length.
 */

private static Object[][] newBigArray(Class<?> componentType, final long length) {
    if (length == 0 && componentType == Object[].class) return EMPTY_BIG_ARRAY;
    ensureLength(length);
    final int baseLength = (int)((length + SEGMENT_MASK) >>> SEGMENT_SHIFT);
    Object[][] base = (Object[][])java.lang.reflect.Array.newInstance(componentType, baseLength);
    final int residual = (int)(length & SEGMENT_MASK);
    if (residual != 0) {
        for(int i = 0; i < baseLength - 1; i++) base[i] =
            (Object[])java.lang.reflect.Array.newInstance(componentType.getComponentType(), SEGMENT_SIZE);
        base[baseLength - 1] = (Object[])java.lang.reflect.Array.newInstance(componentType.getComponentType(),
            residual);
    }
    else for(int i = 0; i < baseLength; i++) base[i] =
        (Object[])java.lang.reflect.Array.newInstance(componentType.getComponentType(), SEGMENT_SIZE);

    return base;
}
#endif

/** Creates a new big array.
 *
 * @param length the length of the new big array.
 * @return a new big array of given length.
 */

public static KEY_TYPE[][] newBigArray(final long length) {
    if (length == 0) return EMPTY_BIG_ARRAY;
    ensureLength(length);
    final int baseLength = (int)((length + SEGMENT_MASK) >>> SEGMENT_SHIFT);
    KEY_TYPE[][] base = new KEY_TYPE[baseLength][];
    final int residual = (int)(length & SEGMENT_MASK);

```

```

if (residual != 0) {
    for(int i = 0; i < baseLength - 1; i++) base[i] = new KEY_TYPE[SEGMENT_SIZE];
    base[baseLength - 1] = new KEY_TYPE[residual];
}
else for(int i = 0; i < baseLength; i++) base[i] = new KEY_TYPE[SEGMENT_SIZE];

return base;
}

#if KEY_CLASS_Object
/** Turns a standard array into a big array.
 *
 * <p>Note that the returned big array might contain as a segment the original array.
 *
 * @param array an array.
 * @return a new big array with the same length and content of { @code array }.
 */

SUPPRESS_WARNINGS_KEY_UNCHECKED
public static <K> K[][] wrap(final K[] array) {
    if (array.length == 0 && array.getClass() == Object[].class) return KEY_GENERIC_BIG_ARRAY_CAST
EMPTY_BIG_ARRAY;
    if (array.length <= SEGMENT_SIZE) {
        final K[][] bigArray = (K[][]).java.lang.reflect.Array.newInstance(array.getClass(), 1);
        bigArray[0] = array;
        return bigArray;
    }
    final K[][] bigArray = (K[][]).newBigArray(array.getClass(), array.length);
    for(int i = 0; i < bigArray.length; i++) System.arraycopy(array, (int)start(i), bigArray[i], 0, bigArray[i].length);
    return bigArray;
}

#else
/** Turns a standard array into a big array.
 *
 * <p>Note that the returned big array might contain as a segment the original array.
 *
 * @param array an array.
 * @return a new big array with the same length and content of { @code array }.
 */

public static KEY_TYPE[][] wrap(final KEY_TYPE[] array) {
    if (array.length == 0) return EMPTY_BIG_ARRAY;
    if (array.length <= SEGMENT_SIZE) return new KEY_TYPE[][] { array };
    final KEY_TYPE[][] bigArray = newBigArray(array.length);
    for(int i = 0; i < bigArray.length; i++) System.arraycopy(array, (int)start(i), bigArray[i], 0, bigArray[i].length);
    return bigArray;
}

```

```

#endif
/** Ensures that a big array can contain the given number of entries.
 *
 * <p>If you cannot foresee whether this big array will need again to be
 * enlarged, you should probably use { @code grow() } instead.
 *
 * <p><strong>Warning:</strong> the returned array might use part of the segments of the original
 * array, which must be considered read-only after calling this method.
 *
 * @param array a big array.
 * @param length the new minimum length for this big array.
 * @return { @code array }, if it contains { @code length } entries or more; otherwise,
 * a big array with { @code length } entries whose first { @code length(array) }
 * entries are the same as those of { @code array }.
 */
public static KEY_GENERIC KEY_GENERIC_TYPE[][] ensureCapacity(final KEY_GENERIC_TYPE[][] array,
final long length) {
    return ensureCapacity(array, length, length(array));
}

#if KEY_CLASS_Object

/** Forces a big array to contain the given number of entries, preserving just a part of the big array.
 *
 * <p>This method returns a new big array of the given length whose element
 * are of the same class as of those of { @code array }.
 *
 * <p><strong>Warning:</strong> the returned array might use part of the segments of the original
 * array, which must be considered read-only after calling this method.
 *
 * @param array a big array.
 * @param length the new minimum length for this big array.
 * @param preserve the number of elements of the big array that must be preserved in case a new allocation is
 * necessary.
 * @return a big array with { @code length } entries whose first { @code preserve }
 * entries are the same as those of { @code array }.
 */
SUPPRESS_WARNINGS_KEY_UNCHECKED
public static KEY_GENERIC KEY_GENERIC_TYPE[][] forceCapacity(final KEY_GENERIC_TYPE[][] array,
final long length, final long preserve) {
    ensureLength(length);
    final int valid = array.length - (array.length == 0 || array.length > 0 && array[array.length - 1].length ==
SEGMENT_SIZE ? 0 : 1);
    final int baseLength = (int)((length + SEGMENT_MASK) >>> SEGMENT_SHIFT);
    final KEY_GENERIC_TYPE[][] base = Arrays.copyOf(array, baseLength);
    final Class<?> componentType = array.getClass().getComponentType();
    final int residual = (int)(length & SEGMENT_MASK);
    if (residual != 0) {

```

```

    for(int i = valid; i < baseLength - 1; i++) base[i] =
(KEY_GENERIC_TYPE[])java.lang.reflect.Array.newInstance(componentType.getComponentType(),
SEGMENT_SIZE);
    base[baseLength - 1] =
(KEY_GENERIC_TYPE[])java.lang.reflect.Array.newInstance(componentType.getComponentType(), residual);
    }
    else for(int i = valid; i < baseLength; i++) base[i] =
(KEY_GENERIC_TYPE[])java.lang.reflect.Array.newInstance(componentType.getComponentType(),
SEGMENT_SIZE);

    if (preserve - (valid * (long)SEGMENT_SIZE) > 0) copy(array, valid * (long)SEGMENT_SIZE, base, valid *
(long)SEGMENT_SIZE, preserve - (valid * (long)SEGMENT_SIZE));
    return base;
}

/** Ensures that a big array can contain the given number of entries, preserving just a part of the big array.
 *
 * <p>This method returns a new big array of the given length whose element
 * are of the same class as of those of { @code array }.
 *
 * <p><strong>Warning:</strong> the returned array might use part of the segments of the original
 * array, which must be considered read-only after calling this method.
 *
 * @param array a big array.
 * @param length the new minimum length for this big array.
 * @param preserve the number of elements of the big array that must be preserved in case a new allocation is
necessary.
 * @return { @code array }, if it can contain { @code length } entries or more; otherwise,
 * a big array with { @code length } entries whose first { @code preserve }
 * entries are the same as those of { @code array }.
 */
SUPPRESS_WARNINGS_KEY_UNCHECKED
public static KEY_GENERIC KEY_GENERIC_TYPE[][] ensureCapacity(final KEY_GENERIC_TYPE[][] array,
final long length, final long preserve) {
    return length > length(array) ? forceCapacity(array, length, preserve) : array;
}

#else

/** Forces a big array to contain the given number of entries, preserving just a part of the big array.
 *
 * <p><strong>Warning:</strong> the returned array might use part of the segments of the original
 * array, which must be considered read-only after calling this method.
 *
 * @param array a big array.
 * @param length the new minimum length for this big array.
 * @param preserve the number of elements of the big array that must be preserved in case a new allocation is
necessary.

```

```

* @return a big array with { @code length } entries whose first { @code preserve }
* entries are the same as those of { @code array }.
*/
public static KEY_TYPE[][] forceCapacity(final KEY_TYPE[][] array, final long length, final long preserve) {
    ensureLength(length);
    final int valid = array.length - (array.length == 0 || array.length > 0 && array[array.length - 1].length ==
SEGMENT_SIZE ? 0 : 1);
    final int baseLength = (int)((length + SEGMENT_MASK) >>> SEGMENT_SHIFT);
    final KEY_TYPE[][] base = Arrays.copyOf(array, baseLength);
    final int residual = (int)(length & SEGMENT_MASK);
    if (residual != 0) {
        for(int i = valid; i < baseLength - 1; i++) base[i] = new KEY_TYPE[SEGMENT_SIZE];
        base[baseLength - 1] = new KEY_TYPE[residual];
    }
    else for(int i = valid; i < baseLength; i++) base[i] = new KEY_TYPE[SEGMENT_SIZE];

    if (preserve - (valid * (long)SEGMENT_SIZE) > 0) copy(array, valid * (long)SEGMENT_SIZE, base, valid *
(long)SEGMENT_SIZE, preserve - (valid * (long)SEGMENT_SIZE));
    return base;
}

/** Ensures that a big array can contain the given number of entries, preserving just a part of the big array.
*
* <p><strong>Warning:</strong> the returned array might use part of the segments of the original
* array, which must be considered read-only after calling this method.
*
* @param array a big array.
* @param length the new minimum length for this big array.
* @param preserve the number of elements of the big array that must be preserved in case a new allocation is
necessary.
* @return { @code array }, if it can contain { @code length } entries or more; otherwise,
* a big array with { @code length } entries whose first { @code preserve }
* entries are the same as those of { @code array }.
*/
public static KEY_TYPE[][] ensureCapacity(final KEY_TYPE[][] array, final long length, final long preserve) {
    return length > length(array) ? forceCapacity(array, length, preserve) : array;
}

#endif

/** Grows the given big array to the maximum between the given length and
* the current length increased by 50%, provided that the given
* length is larger than the current length.
*
* <p>If you want complete control on the big array growth, you
* should probably use { @code ensureCapacity() } instead.
*
* <p><strong>Warning:</strong> the returned array might use part of the segments of the original

```

```

* array, which must be considered read-only after calling this method.
*
* @param array a big array.
* @param length the new minimum length for this big array.
* @return { @code array }, if it can contain { @code length }
* entries; otherwise, a big array with
* max({ @code length },{@code length(array)}/&phi;) entries whose first
* { @code length(array) } entries are the same as those of { @code array }.
* */

```

```

public static KEY_GENERIC KEY_GENERIC_TYPE[][] grow(final KEY_GENERIC_TYPE[][] array, final long
length) {
    final long oldLength = length(array);
    return length > oldLength ? grow(array, length, oldLength) : array;
}

```

```

/** Grows the given big array to the maximum between the given length and
* the current length increased by 50%, provided that the given
* length is larger than the current length, preserving just a part of the big array.
*
* <p>If you want complete control on the big array growth, you
* should probably use { @code ensureCapacity() } instead.
*
* <p><strong>Warning:</strong> the returned array might use part of the segments of the original
* array, which must be considered read-only after calling this method.
*
* @param array a big array.
* @param length the new minimum length for this big array.
* @param preserve the number of elements of the big array that must be preserved in case a new allocation is
necessary.
* @return { @code array }, if it can contain { @code length }
* entries; otherwise, a big array with
* max({ @code length },{@code length(array)}/&phi;) entries whose first
* { @code preserve } entries are the same as those of { @code array }.
* */

```

```

public static KEY_GENERIC KEY_GENERIC_TYPE[][] grow(final KEY_GENERIC_TYPE[][] array, final long
length, final long preserve) {
    final long oldLength = length(array);
    return length > oldLength ? ensureCapacity(array, Math.max(oldLength + (oldLength >> 1), length), preserve) :
array;
}

```

```

#if KEY_CLASS_Object

```

```

/** Trims the given big array to the given length.
*
* <p><strong>Warning:</strong> the returned array might use part of the segments of the original

```

```

* array, which must be considered read-only after calling this method.
*
* @param array a big array.
* @param length the new maximum length for the big array.
* @return { @code array }, if it contains { @code length }
* entries or less; otherwise, a big array with
* { @code length } entries whose entries are the same as
* the first { @code length } entries of { @code array }.
*
*/

public static KEY_GENERIC KEY_GENERIC_TYPE[][] trim(final KEY_GENERIC_TYPE[][] array, final long
length) {
    ensureLength(length);
    final long oldLength = length(array);
    if (length >= oldLength) return array;
    final int baseLength = (int)((length + SEGMENT_MASK) >>> SEGMENT_SHIFT);
    final KEY_GENERIC_TYPE[][] base = Arrays.copyOf(array, baseLength);
    final int residual = (int)(length & SEGMENT_MASK);
    if (residual != 0) base[baseLength - 1] = ARRAYS.trim(base[baseLength - 1], residual);
    return base;
}

#else

/** Trims the given big array to the given length.
*
* <p><strong>Warning:</strong> the returned array might use part of the segments of the original
* array, which must be considered read-only after calling this method.
*
* @param array a big array.
* @param length the new maximum length for the big array.
* @return { @code array }, if it contains { @code length }
* entries or less; otherwise, a big array with
* { @code length } entries whose entries are the same as
* the first { @code length } entries of { @code array }.
*
*/

public static KEY_GENERIC KEY_GENERIC_TYPE[][] trim(final KEY_GENERIC_TYPE[][] array, final long
length) {
    ensureLength(length);
    final long oldLength = length(array);
    if (length >= oldLength) return array;
    final int baseLength = (int)((length + SEGMENT_MASK) >>> SEGMENT_SHIFT);
    final KEY_GENERIC_TYPE[][] base = Arrays.copyOf(array, baseLength);
    final int residual = (int)(length & SEGMENT_MASK);
    if (residual != 0) base[baseLength - 1] = ARRAYS.trim(base[baseLength - 1], residual);
}

```

```
return base;
}
```

```
#endif
```

```
/** Sets the length of the given big array.
```

```
*
```

```
* <p><strong>Warning:</strong> the returned array might use part of the segments of the original  
* array, which must be considered read-only after calling this method.
```

```
*
```

```
* @param array a big array.
```

```
* @param length the new length for the big array.
```

```
* @return { @code array }, if it contains exactly { @code length }
```

```
* entries; otherwise, if it contains <em>more</em> than
```

```
* { @code length } entries, a big array with { @code length } entries
```

```
* whose entries are the same as the first { @code length } entries of
```

```
* { @code array }; otherwise, a big array with { @code length } entries
```

```
* whose first { @code length(array) } entries are the same as those of
```

```
* { @code array }.
```

```
*
```

```
*/
```

```
public static KEY_GENERIC KEY_GENERIC_TYPE[][] setLength(final KEY_GENERIC_TYPE[][] array, final  
long length) {
```

```
    final long oldLength = length(array);
```

```
    if (length == oldLength) return array;
```

```
    if (length < oldLength) return trim(array, length);
```

```
    return ensureCapacity(array, length);
```

```
}
```

```
/** Returns a copy of a portion of a big array.
```

```
*
```

```
* @param array a big array.
```

```
* @param offset the first element to copy.
```

```
* @param length the number of elements to copy.
```

```
* @return a new big array containing { @code length } elements of { @code array } starting at { @code offset }.
```

```
*/
```

```
public static KEY_GENERIC KEY_GENERIC_TYPE[][] copy(final KEY_GENERIC_TYPE[][] array, final long  
offset, final long length) {
```

```
    ensureOffsetLength(array, offset, length);
```

```
    final KEY_GENERIC_TYPE[][] a =
```

```
#if KEY_CLASS_Object
```

```
    newBigArray(array, length);
```

```
#else
```

```
    newBigArray(length);
```

```
#endif
```

```
    copy(array, offset, a, 0, length);
```

```

return a;
}

/** Returns a copy of a big array.
 *
 * @param array a big array.
 * @return a copy of { @code array }.
 */

public static KEY_GENERIC KEY_GENERIC_TYPE[][] copy(final KEY_GENERIC_TYPE[][] array) {
    final KEY_GENERIC_TYPE[][] base = array.clone();
    for(int i = base.length; i-- != 0;) base[i] = array[i].clone();
    return base;
}

/** Fills the given big array with the given value.
 *
 * <p>This method uses a backward loop. It is significantly faster than the corresponding
 * method in { @link java.util.Arrays }.
 *
 * @param array a big array.
 * @param value the new value for all elements of the big array.
 */

public static KEY_GENERIC void fill(final KEY_GENERIC_TYPE[][] array, final KEY_GENERIC_TYPE value)
{
    for(int i = array.length; i-- != 0;) Arrays.fill(array[i], value);
}

/** Fills a portion of the given big array with the given value.
 *
 * <p>If possible (i.e., { @code from } is 0) this method uses a
 * backward loop. In this case, it is significantly faster than the
 * corresponding method in { @link java.util.Arrays }.
 *
 * @param array a big array.
 * @param from the starting index of the portion to fill.
 * @param to the end index of the portion to fill.
 * @param value the new value for all elements of the specified portion of the big array.
 */

public static KEY_GENERIC void fill(final KEY_GENERIC_TYPE[][] array, final long from, long to, final
KEY_GENERIC_TYPE value) {
    final long length = length(array);
    BigArrays.ensureFromTo(length, from, to);
    if (length == 0) return; // To avoid addressing array[0]
    int fromSegment = segment(from);
    int toSegment = segment(to);

```

```

int fromDispl = displacement(from);
int toDispl = displacement(to);
if (fromSegment == toSegment) {
    Arrays.fill(array[fromSegment], fromDispl, toDispl, value);
    return;
}

if (toDispl != 0) Arrays.fill(array[toSegment], 0, toDispl, value);
while(--toSegment > fromSegment) Arrays.fill(array[toSegment], value);
Arrays.fill(array[fromSegment], fromDispl, SEGMENT_SIZE, value);
}

/** Returns true if the two big arrays are elementwise equal.
 *
 * <p>This method uses a backward loop. It is significantly faster than the corresponding
 * method in { @link java.util.Arrays }.
 *
 * @param a1 a big array.
 * @param a2 another big array.
 * @return true if the two big arrays are of the same length, and their elements are equal.
 */

public static KEY_GENERIC boolean equals(final KEY_GENERIC_TYPE[][] a1, final KEY_GENERIC_TYPE
a2[][]) {
    if (length(a1) != length(a2)) return false;
    int i = a1.length, j;
    KEY_GENERIC_TYPE[] t, u;
    while(i-- != 0) {
        t = a1[i];
        u = a2[i];
        j = t.length;
        while(j-- != 0) if (! KEY_EQUALS(t[j], u[j])) return false;
    }
    return true;
}

/** Returns a string representation of the contents of the specified big array.
 *
 * The string representation consists of a list of the big array's elements, enclosed in square brackets ("[]"). Adjacent
elements are separated by the characters ", " (a comma followed by a space). Returns "null" if { @code a } is null.
 * @param a the big array whose string representation to return.
 * @return the string representation of { @code a }.
 */

public static KEY_GENERIC String toString(final KEY_GENERIC_TYPE[][] a) {
    if (a == null) return "null";
    final long last = length(a) - 1;

```

```

if (last == - 1) return "";
final StringBuilder b = new StringBuilder();
b.append('[');
for (long i = 0; ; i++) {
    b.append(String.valueOf(get(a, i)));
    if (i == last) return b.append(']').toString();
    b.append(", ");
}
}

```

/** Ensures that a range given by its first (inclusive) and last (exclusive) elements fits a big array.

*

* <p>This method may be used whenever a big array range check is needed.

*

* @param a a big array.

* @param from a start index (inclusive).

* @param to an end index (inclusive).

* @throws IllegalArgumentException if { @code from } is greater than { @code to }.

* @throws ArrayIndexOutOfBoundsException if { @code from } or { @code to } are greater than the big array length or negative.

*/

```

public static KEY_GENERIC void ensureFromTo(final KEY_GENERIC_TYPE[][] a, final long from, final long to)
{
    BigArrays.ensureFromTo(length(a), from, to);
}

```

/** Ensures that a range given by an offset and a length fits a big array.

*

* <p>This method may be used whenever a big array range check is needed.

*

* @param a a big array.

* @param offset a start index.

* @param length a length (the number of elements in the range).

* @throws IllegalArgumentException if { @code length } is negative.

* @throws ArrayIndexOutOfBoundsException if { @code offset } is negative or { @code offset }+{ @code length } is greater than the big array length.

*/

```

public static KEY_GENERIC void ensureOffsetLength(final KEY_GENERIC_TYPE[][] a, final long offset, final
long length) {
    BigArrays.ensureOffsetLength(length(a), offset, length);
}

```

/** A type-specific content-based hash strategy for big arrays. */

```

private static final class BigArrayHashStrategy KEY_GENERIC implements
Hash.Strategy<KEY_GENERIC_TYPE[][]>, java.io.Serializable {

```

```

private static final long serialVersionUID = -7046029254386353129L;

@Override
public int hashCode(final KEY_GENERIC_TYPE[][] o) { return java.util.Arrays.deepHashCode(o); }

@Override
public boolean equals(final KEY_GENERIC_TYPE[][] a, final KEY_GENERIC_TYPE[][] b) { return
BIG_ARRAYS.equals(a, b); }
}

/** A type-specific content-based hash strategy for big arrays.
 *
 * <p>This hash strategy may be used in custom hash collections whenever keys are
 * big arrays, and they must be considered equal by content. This strategy
 * will handle { @code null } correctly, and it is serializable.
 */

@SuppressWarnings({"rawtypes"})
public static final Hash.Strategy HASH_STRATEGY = new BigArrayHashStrategy();

private static final int SMALL = 7;
private static final int MEDIUM = 40;

private static KEY_GENERIC void vecSwap(final KEY_GENERIC_TYPE[][] x, long a, long b, final long n) {
for(int i = 0; i < n; i++, a++, b++) swap(x, a, b);
}

private static KEY_GENERIC long med3(final KEY_GENERIC_TYPE x[][], final long a, final long b, final long c,
KEY_COMPARATOR KEY_GENERIC comp) {
int ab = comp.compare(get(x, a), get(x, b));
int ac = comp.compare(get(x, a), get(x, c));
int bc = comp.compare(get(x, b), get(x, c));
return (ab < 0 ?
(bc < 0 ? b : ac < 0 ? c : a) :
(bc > 0 ? b : ac > 0 ? c : a));
}

private static KEY_GENERIC void selectionSort(final KEY_GENERIC_TYPE[][] a, final long from, final long to,
final KEY_COMPARATOR KEY_GENERIC comp) {
for(long i = from; i < to - 1; i++) {
long m = i;
for(long j = i + 1; j < to; j++) if (comp.compare(BIG_ARRAYS.get(a, j), BIG_ARRAYS.get(a, m)) < 0) m = j;
if (m != i) swap(a, i, m);
}
}

/** Sorts the specified range of elements according to the order induced by the specified
 * comparator using quicksort.

```

```

*
* <p>The sorting algorithm is a tuned quicksort adapted from Jon L. Bentley and M. Douglas
* McIlroy, &ldquo;Engineering a Sort Function&rdquo;, <i>Software: Practice and Experience</i>, 23(11), pages
* 1249&minus;1265, 1993.
*
* @param x the big array to be sorted.
* @param from the index of the first element (inclusive) to be sorted.
* @param to the index of the last element (exclusive) to be sorted.
* @param comp the comparator to determine the sorting order.
*/
public static KEY_GENERIC void quickSort(final KEY_GENERIC_TYPE[][] x, final long from, final long to,
final KEY_COMPARATOR KEY_GENERIC comp) {
    final long len = to - from;

    // Selection sort on smallest arrays
    if (len < SMALL) {
        selectionSort(x, from, to, comp);
        return;
    }

    // Choose a partition element, v
    long m = from + len / 2; // Small arrays, middle element
    if (len > SMALL) {
        long l = from;
        long n = to - 1;
        if (len > MEDIUM) { // Big arrays, pseudomedian of 9
            long s = len / 8;
            l = med3(x, l, l + s, l + 2 * s, comp);
            m = med3(x, m - s, m, m + s, comp);
            n = med3(x, n - 2 * s, n - s, n, comp);
        }
        m = med3(x, l, m, n, comp); // Mid-size, med of 3
    }

    final KEY_GENERIC_TYPE v = get(x, m);

    // Establish Invariant: v* (<v)* (>v)* v*
    long a = from, b = a, c = to - 1, d = c;
    while(true) {
        int comparison;
        while (b <= c && (comparison = comp.compare(get(x, b), v)) <= 0) {
            if (comparison == 0) swap(x, a++, b);
            b++;
        }
        while (c >= b && (comparison = comp.compare(get(x, c), v)) >= 0) {
            if (comparison == 0) swap(x, c, d--);
            c--;
        }
    }
}

```

```

    if (b > c) break;
    swap(x, b++, c--);
}

// Swap partition elements back to middle
long s, n = to;
s = Math.min(a - from, b - a);
vecSwap(x, from, b - s, s);
s = Math.min(d - c, n - d - 1);
vecSwap(x, b, n - s, s);

// Recursively sort non-partition-elements
if ((s = b - a) > 1) quickSort(x, from, from + s, comp);
if ((s = d - c) > 1) quickSort(x, n - s, n, comp);

}

SUPPRESS_WARNINGS_KEY_UNCHECKED
private static KEY_GENERIC long med3(final KEY_GENERIC_TYPE x[], final long a, final long b, final long c)
{
    int ab = KEY_CMP(get(x, a), get(x, b));
    int ac = KEY_CMP(get(x, a), get(x, c));
    int bc = KEY_CMP(get(x, b), get(x, c));
    return (ab < 0 ?
        (bc < 0 ? b : ac < 0 ? c : a) :
        (bc > 0 ? b : ac > 0 ? c : a));
}

SUPPRESS_WARNINGS_KEY_UNCHECKED
private static KEY_GENERIC void selectionSort(final KEY_GENERIC_TYPE[][] a, final long from, final long to)
{
    for(long i = from; i < to - 1; i++) {
        long m = i;
        for(long j = i + 1; j < to; j++) if (KEY_LESS(BIG_ARRAYS.get(a, j), BIG_ARRAYS.get(a, m))) m = j;
        if (m != i) swap(a, i, m);
    }
}

/** Sorts the specified big array according to the order induced by the specified
 * comparator using quicksort.
 *
 * <p>The sorting algorithm is a tuned quicksort adapted from Jon L. Bentley and M. Douglas
 * McIlroy, "Engineering a Sort Function", Software: Practice and Experience, 23(11), pages
 * 1249-1265, 1993.
 *
 * @param x the big array to be sorted.
 * @param comp the comparator to determine the sorting order.

```

```

*
*/
public static KEY_GENERIC void quickSort(final KEY_GENERIC_TYPE[][] x, final KEY_COMPARATOR
KEY_GENERIC comp) {
    quickSort(x, 0, BIG_ARRAYS.length(x), comp);
}

/** Sorts the specified range of elements according to the natural ascending order using quicksort.
*
* <p>The sorting algorithm is a tuned quicksort adapted from Jon L. Bentley and M. Douglas
* McIlroy, &ldquo;Engineering a Sort Function&rdquo;, <i>Software: Practice and Experience</i>, 23(11), pages
* 1249&minus;1265, 1993.
*
* @param x the big array to be sorted.
* @param from the index of the first element (inclusive) to be sorted.
* @param to the index of the last element (exclusive) to be sorted.
*/

SUPPRESS_WARNINGS_KEY_UNCHECKED
public static KEY_GENERIC void quickSort(final KEY_GENERIC_TYPE[][] x, final long from, final long to) {
    final long len = to - from;

    // Selection sort on smallest arrays
    if (len < SMALL) {
        selectionSort(x, from, to);
        return;
    }

    // Choose a partition element, v
    long m = from + len / 2; // Small arrays, middle element
    if (len > SMALL) {
        long l = from;
        long n = to - 1;
        if (len > MEDIUM) { // Big arrays, pseudomedian of 9
            long s = len / 8;
            l = med3(x, l, l + s, l + 2 * s);
            m = med3(x, m - s, m, m + s);
            n = med3(x, n - 2 * s, n - s, n);
        }
        m = med3(x, l, m, n); // Mid-size, med of 3
    }

    final KEY_GENERIC_TYPE v = get(x, m);

    // Establish Invariant: v* (<v)* (>v)* v*
    long a = from, b = a, c = to - 1, d = c;
    while(true) {
        int comparison;

```

```

while (b <= c && (comparison = KEY_CMP(get(x, b), v)) <= 0) {
    if (comparison == 0) swap(x, a++, b);
    b++;
}
while (c >= b && (comparison = KEY_CMP(get(x, c), v)) >=0) {
    if (comparison == 0) swap(x, c, d--);
    c--;
}
if (b > c) break;
swap(x, b++, c--);
}

// Swap partition elements back to middle
long s, n = to;
s = Math.min(a - from, b - a);
vecSwap(x, from, b - s, s);
s = Math.min(d - c, n - d - 1);
vecSwap(x, b, n - s, s);

// Recursively sort non-partition-elements
if ((s = b - a) > 1) quickSort(x, from, from + s);
if ((s = d - c) > 1) quickSort(x, n - s, n);
}

/** Sorts the specified big array according to the natural ascending order using quicksort.
 *
 * <p>The sorting algorithm is a tuned quicksort adapted from Jon L. Bentley and M. Douglas
 * McIlroy, &ldquo;Engineering a Sort Function&rdquo;, <i>Software: Practice and Experience</i>, 23(11), pages
 * 1249&minus;1265, 1993.
 *
 * @param x the big array to be sorted.
 */

public static KEY_GENERIC void quickSort(final KEY_GENERIC_TYPE[][] x) {
    quickSort(x, 0, BIG_ARRAYS.length(x));
}

#if ! KEY_CLASS_Boolean

/**
 * Searches a range of the specified big array for the specified value using
 * the binary search algorithm. The range must be sorted prior to making this call.
 * If it is not sorted, the results are undefined. If the range contains multiple elements with

```

* the specified value, there is no guarantee which one will be found.
 *
 * @param a the big array to be searched.
 * @param from the index of the first element (inclusive) to be searched.
 * @param to the index of the last element (exclusive) to be searched.
 * @param key the value to be searched for.
 * @return index of the search key, if it is contained in the big array;
 * otherwise, <tt>-(*insertion point* - 1)</tt>. The *insertion point* is defined as the the point at which the value would be inserted into the big array: the index of the first element greater than the key, or the length of the big array, if all elements in the big array are less than the specified key. Note that this guarantees that the return value will be >= 0 if and only if the key is found.
 * @see java.util.Arrays
 */

SUPPRESS_WARNINGS_KEY_UNCHECKED

```
public static KEY_GENERIC long binarySearch(final KEY_GENERIC_TYPE[][] a, long from, long to, final
KEY_GENERIC_TYPE key) {
    KEY_GENERIC_TYPE midVal;
    to--;
    while (from <= to) {
        final long mid = (from + to) >>> 1;
        midVal = get(a, mid);
    #if KEYS_PRIMITIVE
        if (midVal < key) from = mid + 1;
        else if (midVal > key) to = mid - 1;
        else return mid;
    #else
        final int cmp = ((Comparable KEY_SUPER_GENERIC)midVal).compareTo(key);
        if (cmp < 0) from = mid + 1;
        else if (cmp > 0) to = mid - 1;
        else return mid;
    #endif
    }
    return -(from + 1);
}
```

/**

* Searches a big array for the specified value using
 * the binary search algorithm. The range must be sorted prior to making this call.
 * If it is not sorted, the results are undefined. If the range contains multiple elements with
 * the specified value, there is no guarantee which one will be found.
 *
 * @param a the big array to be searched.
 * @param key the value to be searched for.
 * @return index of the search key, if it is contained in the big array;
 * otherwise, <tt>-(*insertion point* - 1)</tt>. The *insertion*

```

*      point</i> is defined as the the point at which the value would
*      be inserted into the big array: the index of the first
*      element greater than the key, or the length of the big array, if all
*      elements in the big array are less than the specified key. Note
*      that this guarantees that the return value will be &gt;= 0 if
*      and only if the key is found.
* @see java.util.Arrays
*/
public static KEY_GENERIC long binarySearch(final KEY_GENERIC_TYPE[][] a, final KEY_TYPE key) {
    return binarySearch(a, 0, BIG_ARRAYS.length(a), key);
}

/**
 * Searches a range of the specified big array for the specified value using
 * the binary search algorithm and a specified comparator. The range must be sorted following the comparator prior
 * to making this call.
 * If it is not sorted, the results are undefined. If the range contains multiple elements with
 * the specified value, there is no guarantee which one will be found.
 *
 * @param a the big array to be searched.
 * @param from the index of the first element (inclusive) to be searched.
 * @param to the index of the last element (exclusive) to be searched.
 * @param key the value to be searched for.
 * @param c a comparator.
 * @return index of the search key, if it is contained in the big array;
 *         otherwise, <tt>-(<i>insertion point</i> - 1)</tt>. The <i>insertion
 *         point</i> is defined as the the point at which the value would
 *         be inserted into the big array: the index of the first
 *         element greater than the key, or the length of the big array, if all
 *         elements in the big array are less than the specified key. Note
 *         that this guarantees that the return value will be &gt;= 0 if
 *         and only if the key is found.
 * @see java.util.Arrays
 */
public static KEY_GENERIC long binarySearch(final KEY_GENERIC_TYPE[][] a, long from, long to, final
KEY_GENERIC_TYPE key, final KEY_COMPARATOR KEY_GENERIC c) {
    KEY_GENERIC_TYPE midVal;
    to--;
    while (from <= to) {
        final long mid = (from + to) >>> 1;
        midVal = get(a, mid);
        final int cmp = c.compare(midVal, key);
        if (cmp < 0) from = mid + 1;
        else if (cmp > 0) to = mid - 1;
        else return mid; // key found
    }
    return -(from + 1);
}

```

```

/**
 * Searches a big array for the specified value using
 * the binary search algorithm and a specified comparator. The range must be sorted following the comparator prior
to making this call.
 * If it is not sorted, the results are undefined. If the range contains multiple elements with
 * the specified value, there is no guarantee which one will be found.
 *
 * @param a the big array to be searched.
 * @param key the value to be searched for.
 * @param c a comparator.
 * @return index of the search key, if it is contained in the big array;
 * otherwise, <tt>-(<i>insertion point</i> - 1)</tt>. The <i>insertion
 * point</i> is defined as the the point at which the value would
 * be inserted into the big array: the index of the first
 * element greater than the key, or the length of the big array, if all
 * elements in the big array are less than the specified key. Note
 * that this guarantees that the return value will be >= 0 if
 * and only if the key is found.
 * @see java.util.Arrays
 */
public static KEY_GENERIC long binarySearch(final KEY_GENERIC_TYPE[][] a, final KEY_GENERIC_TYPE
key, final KEY_COMPARATOR KEY_GENERIC c) {
    return binarySearch(a, 0, BIG_ARRAYS.length(a), key, c);
}

#if KEYS_PRIMITIVE
/** The size of a digit used during radix sort (must be a power of 2). */
private static final int DIGIT_BITS = 8;
/** The mask to extract a digit of { @link #DIGIT_BITS } bits. */
private static final int DIGIT_MASK = (1 << DIGIT_BITS) - 1;
/** The number of digits per element. */
private static final int DIGITS_PER_ELEMENT = KEY_CLASS.SIZE / DIGIT_BITS;

/** This method fixes negative numbers so that the combination exponent/significand is lexicographically sorted. */
#if KEY_CLASS_Double
private static final long fixDouble(final double d) {
    final long l = Double.doubleToRawLongBits(d);
    return l >= 0 ? l : l ^ 0x7FFFFFFFFFFFFFFFL;
}
#elif KEY_CLASS_Float
private static final long fixFloat(final float f) {
    final long i = Float.floatToRawIntBits(f);
    return i >= 0 ? i : i ^ 0x7FFFFFFF;
}
#endif

```

```

/** Sorts the specified big array using radix sort.
 *
 * <p>The sorting algorithm is a tuned radix sort adapted from Peter M. McIlroy, Keith Bostic and M. Douglas
 * McIlroy, &ldquo;Engineering radix sort&rdquo;, <i>Computing Systems</i>, 6(1), pages 5&minus;27 (1993),
 * and further improved using the digit-oracle idea described by
 * Juha K&auml;rkk&auml;inen and Tommi Rantala in &ldquo;Engineering radix sort for strings&rdquo;,
 * <i>String Processing and Information Retrieval, 15th International Symposium</i>, volume 5280 of
 * Lecture Notes in Computer Science, pages 3&minus;14, Springer (2008).
 *
 * <p>This implementation is significantly faster than quicksort
 * already at small sizes (say, more than 10000 elements), but it can only
 * sort in ascending order.
 * It will allocate a support array of bytes with the same number of elements as the array to be sorted.
 *
 * @param a the big array to be sorted.
 */
public static void radixSort(final KEY_TYPE[][] a) {
    radixSort(a, 0, BIG_ARRAYS.length(a));
}

/** Sorts the specified big array using radix sort.
 *
 * <p>The sorting algorithm is a tuned radix sort adapted from Peter M. McIlroy, Keith Bostic and M. Douglas
 * McIlroy, &ldquo;Engineering radix sort&rdquo;, <i>Computing Systems</i>, 6(1), pages 5&minus;27 (1993),
 * and further improved using the digit-oracle idea described by
 * Juha K&auml;rkk&auml;inen and Tommi Rantala in &ldquo;Engineering radix sort for strings&rdquo;,
 * <i>String Processing and Information Retrieval, 15th International Symposium</i>, volume 5280 of
 * Lecture Notes in Computer Science, pages 3&minus;14, Springer (2008).
 *
 * <p>This implementation is significantly faster than quicksort
 * already at small sizes (say, more than 10000 elements), but it can only
 * sort in ascending order.
 * It will allocate a support array of bytes with the same number of elements as the array to be sorted.
 *
 * @param a the big array to be sorted.
 * @param from the index of the first element (inclusive) to be sorted.
 * @param to the index of the last element (exclusive) to be sorted.
 */
public static void radixSort(final KEY_TYPE[][] a, final long from, final long to) {
    final int maxLevel = DIGITS_PER_ELEMENT - 1;

    final int stackSize = ((1 << DIGIT_BITS) - 1) * (DIGITS_PER_ELEMENT - 1) + 1;
    final long[] offsetStack = new long[stackSize];
    int offsetPos = 0;
    final long[] lengthStack = new long[stackSize];
    int lengthPos = 0;
    final int[] levelStack = new int[stackSize];

```

```

int levelPos = 0;

offsetStack[offsetPos++] = from;
lengthStack[lengthPos++] = to - from;
levelStack[levelPos++] = 0;

final long[] count = new long[1 << DIGIT_BITS];
final long[] pos = new long[1 << DIGIT_BITS];
final byte[][] digit = ByteBigArrays.newBigArray(to - from);

while(offsetPos > 0) {
    final long first = offsetStack[--offsetPos];
    final long length = lengthStack[--lengthPos];
    final int level = levelStack[--levelPos];
    #if KEY_CLASS_Character
        final int signMask = 0;
    #else
        final int signMask = level % DIGITS_PER_ELEMENT == 0 ? 1 << DIGIT_BITS - 1 : 0;
    #endif

    if (length < MEDIUM) {
        selectionSort(a, first, first + length);
        continue;
    }

    final int shift = (DIGITS_PER_ELEMENT - 1 - level % DIGITS_PER_ELEMENT) * DIGIT_BITS; // This is the
    shift that extract the right byte from a key

    // Count keys.

    for(long i = length; i-- != 0;) ByteBigArrays.set(digit, i, (byte)((((KEY2LEXINT(BIG_ARRAYS.get(a, first + i))
    >>> shift) & DIGIT_MASK) ^ signMask)));
    for(long i = length; i-- != 0;) count[ByteBigArrays.get(digit, i) & 0xFF]++;
    // Compute cumulative distribution and push non-singleton keys on stack.
    int lastUsed = -1;

    long p = 0;
    for(int i = 0; i < 1 << DIGIT_BITS; i++) {
        if (count[i] != 0) {
            lastUsed = i;
            if (level < maxLevel && count[i] > 1){
                //System.err.println(" Pushing " + new StackEntry(first + pos[i - 1], first + pos[i], level + 1));
                offsetStack[offsetPos++] = p + first;
                lengthStack[lengthPos++] = count[i];
                levelStack[levelPos++] = level + 1;
            }
        }
        pos[i] = (p += count[i]);
    }
}

```

```

}

// When all slots are OK, the last slot is necessarily OK.
final long end = length - count[lastUsed];
count[lastUsed] = 0;

// i moves through the start of each block
int c = -1;
for(long i = 0, d; i < end; i += count[c], count[c] = 0) {
    KEY_TYPE t = BIG_ARRAYS.get(a, i + first);
    c = ByteBigArrays.get(digit, i) & 0xFF;
    while((d = --pos[c]) > i) {
        final KEY_TYPE z = t;
        final int zz = c;
        t = BIG_ARRAYS.get(a, d + first);
        c = ByteBigArrays.get(digit, d) & 0xFF;
        BIG_ARRAYS.set(a, d + first, z);
        ByteBigArrays.set(digit, d, (byte)zz);
    }

    BIG_ARRAYS.set(a, i + first, t);
}
}
}

private static void selectionSort(final KEY_TYPE[][] a, final KEY_TYPE[][] b, final long from, final long to) {
    for(long i = from; i < to - 1; i++) {
        long m = i;
        for(long j = i + 1; j < to; j++)
            if (KEY_LESS(BIG_ARRAYS.get(a, j), BIG_ARRAYS.get(a, m)) || KEY_CMP_EQ(BIG_ARRAYS.get(a, j),
BIG_ARRAYS.get(a, m)) && KEY_LESS(BIG_ARRAYS.get(b, j), BIG_ARRAYS.get(b, m))) m = j;

        if (m != i) {
            KEY_TYPE t = BIG_ARRAYS.get(a, i);
            BIG_ARRAYS.set(a, i, BIG_ARRAYS.get(a, m));
            BIG_ARRAYS.set(a, m, t);
            t = BIG_ARRAYS.get(b, i);
            BIG_ARRAYS.set(b, i, BIG_ARRAYS.get(b, m));
            BIG_ARRAYS.set(b, m, t);
        }
    }
}

/** Sorts the specified pair of big arrays lexicographically using radix sort.
 * <p>The sorting algorithm is a tuned radix sort adapted from Peter M. McIlroy, Keith Bostic and M. Douglas
 * McIlroy, &ldquo;Engineering radix sort&rdquo;, <i>Computing Systems</i>, 6(1), pages 5&minus;27 (1993),
 * and further improved using the digit-oracle idea described by

```

```

* Juha K&auml;rkk&auml;inen and Tommi Rantala in &ldquo;Engineering radix sort for strings&rdquo;,
* <i>String Processing and Information Retrieval, 15th International Symposium</i>, volume 5280 of
* Lecture Notes in Computer Science, pages 3&minus;14, Springer (2008).
*
* <p>This method implements a <em>lexicographical</em> sorting of the arguments. Pairs of elements
* in the same position in the two provided arrays will be considered a single key, and permuted
* accordingly. In the end, either { @code a[i] &lt; a[i + 1]} or { @code a[i] == a[i + 1]} and { @code b[i] &lt;= b[i +
1]}.
*
* <p>This implementation is significantly faster than quicksort
* already at small sizes (say, more than 10000 elements), but it can only
* sort in ascending order. It will allocate a support array of bytes with the same number of elements as the arrays to
be sorted.
*
* @param a the first big array to be sorted.
* @param b the second big array to be sorted.
*/

public static void radixSort(final KEY_TYPE[][] a, final KEY_TYPE[][] b) {
    radixSort(a, b, 0, BIG_ARRAYS.length(a));
}

/** Sorts the specified pair of big arrays lexicographically using radix sort.
*
* <p>The sorting algorithm is a tuned radix sort adapted from Peter M. McIlroy, Keith Bostic and M. Douglas
* McIlroy, &ldquo;Engineering radix sort&rdquo;, <i>Computing Systems</i>, 6(1), pages 5&minus;27 (1993),
* and further improved using the digit-oracle idea described by
* Juha K&auml;rkk&auml;inen and Tommi Rantala in &ldquo;Engineering radix sort for strings&rdquo;,
* <i>String Processing and Information Retrieval, 15th International Symposium</i>, volume 5280 of
* Lecture Notes in Computer Science, pages 3&minus;14, Springer (2008).
*
* <p>This method implements a <em>lexicographical</em> sorting of the arguments. Pairs of elements
* in the same position in the two provided arrays will be considered a single key, and permuted
* accordingly. In the end, either { @code a[i] &lt; a[i + 1]} or { @code a[i] == a[i + 1]} and { @code b[i] &lt;= b[i +
1]}.
*
* <p>This implementation is significantly faster than quicksort
* already at small sizes (say, more than 10000 elements), but it can only
* sort in ascending order. It will allocate a support array of bytes with the same number of elements as the arrays to
be sorted.
*
* @param a the first big array to be sorted.
* @param b the second big array to be sorted.
* @param from the index of the first element (inclusive) to be sorted.
* @param to the index of the last element (exclusive) to be sorted.
*/

public static void radixSort(final KEY_TYPE[][] a, final KEY_TYPE[][] b, final long from, final long to) {
    final int layers = 2;

```

```

if (BIG_ARRAYS.length(a) != BIG_ARRAYS.length(b)) throw new IllegalArgumentException("Array size
mismatch.");
final int maxLevel = DIGITS_PER_ELEMENT * layers - 1;

final int stackSize = ((1 << DIGIT_BITS) - 1) * (layers * DIGITS_PER_ELEMENT - 1) + 1;
final long[] offsetStack = new long[stackSize];
int offsetPos = 0;
final long[] lengthStack = new long[stackSize];
int lengthPos = 0;
final int[] levelStack = new int[stackSize];
int levelPos = 0;

offsetStack[offsetPos++] = from;
lengthStack[lengthPos++] = to - from;
levelStack[levelPos++] = 0;

final long[] count = new long[1 << DIGIT_BITS];
final long[] pos = new long[1 << DIGIT_BITS];
final byte[][] digit = ByteBigArrays.newBigArray(to - from);

while(offsetPos > 0) {
    final long first = offsetStack[--offsetPos];
    final long length = lengthStack[--lengthPos];
    final int level = levelStack[--levelPos];
#ifdef KEY_CLASS_Character
    final int signMask = 0;
#else
    final int signMask = level % DIGITS_PER_ELEMENT == 0 ? 1 << DIGIT_BITS - 1 : 0;
#endif
    if (length < MEDIUM) {
        selectionSort(a, b, first, first + length);
        continue;
    }

    final KEY_TYPE[][] k = level < DIGITS_PER_ELEMENT ? a : b; // This is the key array
    final int shift = (DIGITS_PER_ELEMENT - 1 - level % DIGITS_PER_ELEMENT) * DIGIT_BITS; // This is the
    shift that extract the right byte from a key

    // Count keys.
    for(long i = length; i-- != 0;) ByteBigArrays.set(digit, i, (byte)((((KEY2LEXINT(BIG_ARRAYS.get(k, first + i))
>>> shift) & DIGIT_MASK) ^ signMask));
    for(long i = length; i-- != 0;) count[ByteBigArrays.get(digit, i) & 0xFF]++;
    // Compute cumulative distribution and push non-singleton keys on stack.
    int lastUsed = -1;

    long p = 0;
    for(int i = 0; i < 1 << DIGIT_BITS; i++) {

```

```

if (count[i] != 0) {
    lastUsed = i;
    if (level < maxLevel && count[i] > 1){
        offsetStack[offsetPos++] = p + first;
        lengthStack[lengthPos++] = count[i];
        levelStack[levelPos++] = level + 1;
    }
}
pos[i] = (p += count[i]);
}

// When all slots are OK, the last slot is necessarily OK.
final long end = length - count[lastUsed];
count[lastUsed] = 0;

// i moves through the start of each block
int c = -1;
for(long i = 0, d; i < end; i += count[c], count[c] = 0) {
    KEY_TYPE t = BIG_ARRAYS.get(a, i + first);
    KEY_TYPE u = BIG_ARRAYS.get(b, i + first);
    c = ByteBigArrays.get(digit, i) & 0xFF;
    while((d = --pos[c]) > i) {
        KEY_TYPE z = t;
        final int zz = c;
        t = BIG_ARRAYS.get(a, d + first);
        BIG_ARRAYS.set(a, d + first, z);
        z = u;
        u = BIG_ARRAYS.get(b, d + first);
        BIG_ARRAYS.set(b, d + first, z);
        c = ByteBigArrays.get(digit, d) & 0xFF;
        ByteBigArrays.set(digit, d, (byte)zz);
    }

    BIG_ARRAYS.set(a, i + first, t);
    BIG_ARRAYS.set(b, i + first, u);
}
}
}

#endif

#endif

/** Shuffles the specified big array fragment using the specified pseudorandom number generator.
 *
 * @param a the big array to be shuffled.
 * @param from the index of the first element (inclusive) to be shuffled.

```

```

* @param to the index of the last element (exclusive) to be shuffled.
* @param random a pseudorandom number generator.
* @return { @code a}.
*/
public static KEY_GENERIC KEY_GENERIC_TYPE[][] shuffle(final KEY_GENERIC_TYPE[][] a, final long
from, final long to, final Random random) {
    for(long i = to - from; i-- != 0;) {
        final long p = (random.nextLong() & 0x7FFFFFFFFFFFFFFFL) % (i + 1);
        final KEY_GENERIC_TYPE t = get(a, from + i);
        set(a, from + i, get(a, from + p));
        set(a, from + p, t);
    }
    return a;
}

/** Shuffles the specified big array using the specified pseudorandom number generator.
*
* @param a the big array to be shuffled.
* @param random a pseudorandom number generator.
* @return { @code a}.
*/
public static KEY_GENERIC KEY_GENERIC_TYPE[][] shuffle(final KEY_GENERIC_TYPE[][] a, final Random
random) {
    for(long i = length(a); i-- != 0;) {
        final long p = (random.nextLong() & 0x7FFFFFFFFFFFFFFFL) % (i + 1);
        final KEY_GENERIC_TYPE t = get(a, i);
        set(a, i, get(a, p));
        set(a, p, t);
    }
    return a;
}

#if KEY_CLASS_Integer
#ifdef TEST

private static long seed = System.currentTimeMillis();
private static java.util.Random r = new java.util.Random(seed);

private static KEY_TYPE genKey() {
#if KEY_CLASS_Byte || KEY_CLASS_Short || KEY_CLASS_Character
    return (KEY_TYPE)(r.nextInt());
#elif KEYS_PRIMITIVE
    return r.NEXT_KEY();
#elif KEY_CLASS_Object
    return Integer.toBinaryString(r.nextInt());
#else
    return new java.io.Serializable() {};
#endif
}

```

```

#endif
}

private static Object[] k, v, nk;
private static KEY_TYPE kt[];
private static KEY_TYPE nkt[];
private static BIG_ARRAY_BIG_LIST topList;

protected static void speedTest(int n, boolean b) {}

protected static void runTest(int n) {
    KEY_TYPE[][] a = BIG_ARRAYS.newBigArray(n);
    for(int i = 0; i < n; i++) set(a, i, i);
    BIG_ARRAYS.copy(a, 0, a, 1, n - 2);
    assert a[0][0] == 0;
    for(int i = 0; i < n - 2; i++) assert get(a, i + 1) == i;

    for(int i = 0; i < n; i++) set(a, i, i);
    BIG_ARRAYS.copy(a, 1, a, 0, n - 1);
    for(int i = 0; i < n - 1; i++) assert get(a, i) == i + 1;

    for(int i = 0; i < n; i++) set(a, i, i);
    KEY_TYPE[] b = new KEY_TYPE[n];
    for(int i = 0; i < n; i++) b[i] = i;

    assert equals(wrap(b), a);

    System.out.println("Test OK");
    return;
}

public static void main(String args[]) {
    int n = Integer.parseInt(args[1]);
    if (args.length > 2) r = new java.util.Random(seed = Long.parseLong(args[2]));

    try {
        if ("speedTest".equals(args[0]) || "speedComp".equals(args[0])) speedTest(n, "speedComp".equals(args[0]));
        else if ("test".equals(args[0])) runTest(n);
    } catch(Throwable e) {
        e.printStackTrace(System.err);
        System.err.println("seed: " + seed);
    }
}

#endif
#endif

```

```
}
```

Found in path(s):

```
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-  
a775574/drv/BigArrays.drv
```

No license file was found, but licenses were detected in source scan.

```
/*
```

```
* Copyright (C) 2007-2017 Sebastiano Vigna
```

```
*
```

```
* Licensed under the Apache License, Version 2.0 (the "License");
```

```
* you may not use this file except in compliance with the License.
```

```
* You may obtain a copy of the License at
```

```
*
```

```
* http://www.apache.org/licenses/LICENSE-2.0
```

```
*
```

```
* Unless required by applicable law or agreed to in writing, software
```

```
* distributed under the License is distributed on an "AS IS" BASIS,
```

```
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
```

```
* See the License for the specific language governing permissions and
```

```
* limitations under the License.
```

```
*/
```

```
package PACKAGE;
```

```
import java.util.Map;
```

```
import java.util.NoSuchElementException;
```

```
import it.unimi.dsi.fastutil.objects.AbstractObjectSet;
```

```
import it.unimi.dsi.fastutil.objects.ObjectIterator;
```

```
import VALUE_PACKAGE.VALUE_COLLECTION;
```

```
import VALUE_PACKAGE.VALUE_ABSTRACT_COLLECTION;
```

```
import VALUE_PACKAGE.VALUE_ARRAYS;
```

```
/** A simple, brute-force implementation of a map based on two parallel backing arrays.
```

```
*
```

```
* <p>The main purpose of this
```

```
* implementation is that of wrapping cleanly the brute-force approach to the storage of a very
```

```
* small number of pairs: just put them into two parallel arrays and scan linearly to find an item.
```

```
*/
```

```
public class ARRAY_MAP KEY_VALUE_GENERIC extends ABSTRACT_MAP KEY_VALUE_GENERIC  
implements java.io.Serializable, Cloneable {
```

```
private static final long serialVersionUID = 1L;
```

```
/** The keys (valid up to {@link #size}, excluded). */
```

```

private transient KEY_TYPE[] key;
/** The values (parallel to {@link #key}). */
private transient VALUE_TYPE[] value;
/** The number of valid entries in {@link #key} and {@link #value}. */
private int size;

/** Creates a new empty array map with given key and value backing arrays. The resulting map will have as many
entries as the given arrays.
*
* <p>It is responsibility of the caller that the elements of {@code key} are distinct.
*
* @param key the key array.
* @param value the value array (it <em>must</em> have the same length as {@code key}).
*/
public ARRAY_MAP(final KEY_TYPE[] key, final VALUE_TYPE[] value) {
    this.key = key;
    this.value = value;
    size = key.length;
    if(key.length != value.length) throw new IllegalArgumentException("Keys and values have different lengths (" +
key.length + ", " + value.length + ")");
}

/** Creates a new empty array map.
*/
public ARRAY_MAP() {
    this.key = ARRAYS.EMPTY_ARRAY;
    this.value = VALUE_ARRAYS.EMPTY_ARRAY;
}

/** Creates a new empty array map of given capacity.
*
* @param capacity the initial capacity.
*/
public ARRAY_MAP(final int capacity) {
    this.key = new KEY_TYPE[capacity];
    this.value = new VALUE_TYPE[capacity];
}

/** Creates a new empty array map copying the entries of a given map.
*
* @param m a map.
*/
public ARRAY_MAP(final MAP KEY_VALUE_GENERIC m) {
    this(m.size());
    putAll(m);
}

/** Creates a new empty array map copying the entries of a given map.

```

```

*
* @param m a map.
*/
public ARRAY_MAP(final Map<? extends KEY_GENERIC_CLASS, ? extends VALUE_GENERIC_CLASS> m)
{
    this(m.size());
    putAll(m);
}

/** Creates a new array map with given key and value backing arrays, using the given number of elements.
*
* <p>It is responsibility of the caller that the first { @code size } elements of { @code key } are distinct.
*
* @param key the key array.
* @param value the value array (it <em>must</em> have the same length as { @code key }).
* @param size the number of valid elements in { @code key } and { @code value }.
*/
public ARRAY_MAP(final KEY_TYPE[] key, final VALUE_TYPE[] value, final int size) {
    this.key = key;
    this.value = value;
    this.size = size;
    if(key.length != value.length) throw new IllegalArgumentException("Keys and values have different lengths (" +
key.length + ", " + value.length + ")");
    if (size > key.length) throw new IllegalArgumentException("The provided size (" + size + ") is larger than or equal
to the backing-arrays size (" + key.length + ")");
}

private final class EntrySet extends AbstractObjectSet<MAP.Entry KEY_VALUE_GENERIC> implements
FastEntrySet KEY_VALUE_GENERIC {

    @Override
    public ObjectIterator<MAP.Entry KEY_VALUE_GENERIC> iterator() {
        return new ObjectIterator<MAP.Entry KEY_VALUE_GENERIC>() {
            int curr = -1, next = 0;

            @Override
            public boolean hasNext() { return next < size; }

            @Override
            SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
            public Entry KEY_VALUE_GENERIC next() {
                if (! hasNext()) throw new NoSuchElementException();
                return new ABSTRACT_MAP.BasicEntry KEY_VALUE_GENERIC_DIAMOND(KEY_GENERIC_CAST
key[curr = next], VALUE_GENERIC_CAST value[next++]);
            }

            @Override
            public void remove() {

```



```

}

@Override
public int size() { return size; }

@Override
SUPPRESS_WARNINGS_KEY_UNCHECKED
public boolean contains(Object o) {
    if (!(o instanceof Map.Entry)) return false;
    final Map.Entry<?,?> e = (Map.Entry<?,?>)o;
    #if KEYS_PRIMITIVE
        if (e.getKey() == null || !(e.getKey() instanceof KEY_CLASS)) return false;
    #endif
    #if VALUES_PRIMITIVE
        if (e.getValue() == null || !(e.getValue() instanceof VALUE_CLASS)) return false;
    #endif
    final KEY_GENERIC_TYPE k = KEY_OBJ2TYPE(KEY_GENERIC_CAST e.getKey());
    return ARRAY_MAP.this.containsKey(k) && VALUE_EQUALS(ARRAY_MAP.this.GET_VALUE(k),
VALUE_OBJ2TYPE(e.getValue()));
}

@Override
SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
public boolean remove(final Object o) {
    if (!(o instanceof Map.Entry)) return false;
    final Map.Entry<?,?> e = (Map.Entry<?,?>)o;
    #if KEYS_PRIMITIVE
        if (e.getKey() == null || !(e.getKey() instanceof KEY_CLASS)) return false;
    #endif
    #if VALUES_PRIMITIVE
        if (e.getValue() == null || !(e.getValue() instanceof VALUE_CLASS)) return false;
    #endif
    final KEY_GENERIC_TYPE k = KEY_OBJ2TYPE(KEY_GENERIC_CAST e.getKey());
    final VALUE_GENERIC_TYPE v = VALUE_OBJ2TYPE(VALUE_GENERIC_CAST e.getValue());

    final int oldPos = ARRAY_MAP.this.findKey(k);
    if (oldPos == -1 || !VALUE_EQUALS(v, ARRAY_MAP.this.value[oldPos])) return false;
    final int tail = size - oldPos - 1;
    System.arraycopy(ARRAY_MAP.this.key, oldPos + 1, ARRAY_MAP.this.key, oldPos, tail);
    System.arraycopy(ARRAY_MAP.this.value, oldPos + 1, ARRAY_MAP.this.value, oldPos, tail);
    ARRAY_MAP.this.size--;
    #if KEYS_REFERENCE
        ARRAY_MAP.this.key[size] = null;
    #endif
    #if VALUES_REFERENCE
        ARRAY_MAP.this.value[size] = null;
    #endif
    return true;
}

```

```

    }
}

@Override
public FastEntrySet KEY_VALUE_GENERIC ENTRYSET() { return new EntrySet(); }

private int findKey(final KEY_TYPE k) {
    final KEY_TYPE[] key = this.key;
    for(int i = size; i-- != 0;) if (KEY_EQUALS(key[i], k)) return i;
    return -1;
}

@Override
SUPPRESS_WARNINGS_VALUE_UNCHECKED
public VALUE_GENERIC_TYPE GET_VALUE(final KEY_TYPE k) {
    final KEY_TYPE[] key = this.key;
    for(int i = size; i-- != 0;) if (KEY_EQUALS(key[i], k)) return VALUE_GENERIC_CAST value[i];
    return defRetValue;
}

@Override
public int size() { return size; }

@Override
public void clear() {
    #if KEYS_REFERENCE || VALUES_REFERENCE
        for(int i = size; i-- != 0;) {
            #if KEYS_REFERENCE
                key[i] = null;
            #endif
            #if VALUES_REFERENCE
                value[i] = null;
            #endif
        }
    #endif
    size = 0;
}

@Override
public boolean containsKey(final KEY_TYPE k) { return findKey(k) != -1; }

@Override
public boolean containsValue(VALUE_TYPE v) {
    for(int i = size; i-- != 0;) if (VALUE_EQUALS(value[i], v)) return true;
    return false;
}

@Override

```

```

public boolean isEmpty() { return size == 0; }

@Override
SUPPRESS_WARNINGS_VALUE_UNCHECKED
public VALUE_GENERIC_TYPE put(KEY_GENERIC_TYPE k, VALUE_GENERIC_TYPE v) {
    final int oldKey = findKey(k);
    if (oldKey != -1) {
        final VALUE_GENERIC_TYPE oldValue = VALUE_GENERIC_CAST value[oldKey];
        value[oldKey] = v;
        return oldValue;
    }
    if (size == key.length) {
        final KEY_TYPE[] newKey = new KEY_TYPE[size == 0 ? 2 : size * 2];
        final VALUE_TYPE[] newValue = new VALUE_TYPE[size == 0 ? 2 : size * 2];
        for(int i = size; i-- != 0;) {
            newKey[i] = key[i];
            newValue[i] = value[i];
        }
        key = newKey;
        value = newValue;
    }
    key[size] = k;
    value[size] = v;
    size++;
    return defRetValue;
}

@Override
SUPPRESS_WARNINGS_VALUE_UNCHECKED
public VALUE_GENERIC_TYPE REMOVE_VALUE(final KEY_TYPE k) {
    final int oldPos = findKey(k);
    if (oldPos == -1) return defRetValue;
    final VALUE_GENERIC_TYPE oldValue = VALUE_GENERIC_CAST value[oldPos];
    final int tail = size - oldPos - 1;
    System.arraycopy(key, oldPos + 1, key, oldPos, tail);
    System.arraycopy(value, oldPos + 1, value, oldPos, tail);
    size--;
#ifdef KEYS_REFERENCE
    key[size] = null;
#endif
#ifdef VALUES_REFERENCE
    value[size] = null;
#endif
    return oldValue;
}

@Override
public SET KEY_GENERIC keySet() {

```

```

return new ABSTRACT_SET KEY_GENERIC() {
    @Override
    public boolean contains(final KEY_TYPE k) {
        return findKey(k) != -1;
    }

    @Override
    public boolean remove(final KEY_TYPE k) {
        final int oldPos = findKey(k);
        if (oldPos == -1) return false;
        final int tail = size - oldPos - 1;
        System.arraycopy(key, oldPos + 1, key, oldPos, tail);
        System.arraycopy(value, oldPos + 1, value, oldPos, tail);
        size--;
        return true;
    }

    @Override
    public KEY_ITERATOR KEY_GENERIC iterator() {
        return new KEY_ITERATOR KEY_GENERIC() {
            int pos = 0;
            @Override
            public boolean hasNext() {
                return pos < size;
            }
        }

        @Override
        SUPPRESS_WARNINGS_KEY_UNCHECKED
        public KEY_GENERIC_TYPE NEXT_KEY() {
            if (! hasNext()) throw new NoSuchElementException();
            return KEY_GENERIC_CAST key[pos++];
        }

        @Override
        public void remove() {
            if (pos == 0) throw new IllegalStateException();
            final int tail = size - pos;
            System.arraycopy(key, pos, key, pos - 1, tail);
            System.arraycopy(value, pos, value, pos - 1, tail);
            size--;
        }
    };
}

    @Override
    public int size() {
        return size;
    }
}

```

```

@Override
public void clear() {
    ARRAY_MAP.this.clear();
}
};
}

```

```

@Override
public VALUE_COLLECTION VALUE_GENERIC values() {
    return new VALUE_ABSTRACT_COLLECTION VALUE_GENERIC() {

```

```

@Override
public boolean contains(final VALUE_TYPE v) {
    return containsValue(v);
}

```

```

@Override
public VALUE_PACKAGE.VALUE_ITERATOR VALUE_GENERIC iterator() {
    return new VALUE_PACKAGE.VALUE_ITERATOR VALUE_GENERIC() {
        int pos = 0;
        @Override
        public boolean hasNext() {
            return pos < size;
        }
    }
}

```

```

@Override
SUPPRESS_WARNINGS_VALUE_UNCHECKED
public VALUE_GENERIC_TYPE NEXT_VALUE() {
    if (!hasNext()) throw new NoSuchElementException();
    return VALUE_GENERIC_CAST value[pos++];
}

```

```

@Override
public void remove() {
    if (pos == 0) throw new IllegalStateException();
    final int tail = size - pos;
    System.arraycopy(key, pos, key, pos - 1, tail);
    System.arraycopy(value, pos, value, pos - 1, tail);
    size--;
}
};
}

```

```

@Override
public int size() {
    return size;
}

```

```

@Override
public void clear() {
    ARRAY_MAP.this.clear();
}
};
}

/** Returns a deep copy of this map.
 *
 * <p>This method performs a deep copy of this hash map; the data stored in the
 * map, however, is not cloned. Note that this makes a difference only for object keys.
 *
 * @return a deep copy of this map.
 */
@Override
SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
public ARRAY_MAP KEY_VALUE_GENERIC clone() {
    ARRAY_MAP KEY_VALUE_GENERIC c;
    try {
        c = (ARRAY_MAP KEY_VALUE_GENERIC)super.clone();
    }
    catch(CloneNotSupportedException cantHappen) {
        throw new InternalError();
    }
    c.key = key.clone();
    c.value = value.clone();
    return c;
}

private void writeObject(java.io.ObjectOutputStream s) throws java.io.IOException {
    s.defaultWriteObject();
    for(int i = 0; i < size; i++) {
        s.WRITE_KEY(key[i]);
        s.WRITE_VALUE(value[i]);
    }
}

private void readObject(java.io.ObjectInputStream s) throws java.io.IOException, ClassNotFoundException {
    s.defaultReadObject();
    key = new KEY_TYPE[size];
    value = new VALUE_TYPE[size];
    for(int i = 0; i < size; i++) {
        key[i] = s.READ_KEY();
        value[i] = s.READ_VALUE();
    }
}
}

```

Found in path(s):

```
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-
a775574/drv/ArrayMap.drv
```

No license file was found, but licenses were detected in source scan.

```
/*
```

```
* Copyright (C) 2010-2017 Sebastiano Vigna
```

```
*
```

```
* Licensed under the Apache License, Version 2.0 (the "License");
```

```
* you may not use this file except in compliance with the License.
```

```
* You may obtain a copy of the License at
```

```
*
```

```
* http://www.apache.org/licenses/LICENSE-2.0
```

```
*
```

```
* Unless required by applicable law or agreed to in writing, software
```

```
* distributed under the License is distributed on an "AS IS" BASIS,
```

```
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
```

```
* See the License for the specific language governing permissions and
```

```
* limitations under the License.
```

```
*/
```

```
package PACKAGE;
```

```
#if KEY_CLASS_Object
```

```
import java.util.Arrays;
```

```
import java.util.Comparator;
```

```
#endif
```

```
import java.io.Serializable;
```

```
import it.unimi.dsi.fastutil.HashCommon;
```

```
import it.unimi.dsi.fastutil.PriorityQueue;
```

```
import java.util.NoSuchElementException;
```

```
/** A type-specific array-based FIFO queue, supporting also deque operations.
```

```
*
```

```
* <p>Instances of this class represent a FIFO queue using a backing
```

```
* array in a circular way. The array is enlarged and shrunk as needed. You can use the { @link #trim()} method
```

```
* to reduce its memory usage, if necessary.
```

```
*
```

```
* <p>This class provides additional methods that implement a <em>deque</em> (double-ended queue).
```

```
*/
```

```
public class ARRAY_FIFO_QUEUE KEY_GENERIC implements PRIORITY_QUEUE KEY_GENERIC,
Serializable {
```

```
    private static final long serialVersionUID = 0L;
```

```

/** The standard initial capacity of a queue. */
public static final int INITIAL_CAPACITY = 4;

/** The backing array. */
protected transient KEY_GENERIC_TYPE array[];

/** The current (cached) length of {@link #array}. */
protected transient int length;

/** The start position in {@link #array}. It is always strictly smaller than {@link #length}. */
protected transient int start;

/** The end position in {@link #array}. It is always strictly smaller than {@link #length}.
 * Might be actually smaller than {@link #start} because {@link #array} is used cyclically. */
protected transient int end;

/** Creates a new empty queue with given capacity.
 *
 * @param capacity the initial capacity of this queue.
 */
SUPPRESS_WARNINGS_KEY_UNCHECKED
public ARRAY_FIFO_QUEUE(final int capacity) {
    if (capacity < 0) throw new IllegalArgumentException("Initial capacity (" + capacity + ") is negative");
    array = KEY_GENERIC_ARRAY_CAST new KEY_TYPE[Math.max(1, capacity)]; // Never build a queue with
zero-sized backing array.
    length = array.length;
}

/** Creates a new empty queue with standard {@linkplain #INITIAL_CAPACITY} initial capacity. */
public ARRAY_FIFO_QUEUE() {
    this(INITIAL_CAPACITY);
}

/** {@inheritDoc}
 * <p>This implementation returns {@code null} (FIFO queues have no comparator). */
@Override
public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() {
    return null;
}

@Override
public KEY_GENERIC_TYPE DEQUEUE() {
    if (start == end) throw new NoSuchElementException();
    final KEY_GENERIC_TYPE t = array[start];
#if KEYS_REFERENCE
    array[start] = null; // Clean-up for the garbage collector.
#endif
}

```

```

    if (++start == length) start = 0;
    reduce();
    return t;
}

/** Dequeues the { @linkplain PriorityQueue#last() last } element from the queue.
 *
 * @return the dequeued element.
 * @throws NoSuchElementException if the queue is empty.
 */
public KEY_GENERIC_TYPE DEQUEUE_LAST() {
    if (start == end) throw new NoSuchElementException();
    if (end == 0) end = length;
    final KEY_GENERIC_TYPE t = array[--end];
    #if KEYS_REFERENCE
        array[end] = null; // Clean-up for the garbage collector.
    #endif
    reduce();
    return t;
}

SUPPRESS_WARNINGS_KEY_UNCHECKED
private final void resize(final int size, final int newLength) {
    final KEY_GENERIC_TYPE[] newArray = KEY_GENERIC_ARRAY_CAST new KEY_TYPE[newLength];
    if (start >= end) {
        if (size != 0) {
            System.arraycopy(array, start, newArray, 0, length - start);
            System.arraycopy(array, 0, newArray, length - start, end);
        }
    }
    else System.arraycopy(array, start, newArray, 0, end - start);
    start = 0;
    end = size;
    array = newArray;
    length = newLength;
}

private final void expand() {
    resize(length, (int)Math.min(it.unimi.dsi.fastutil.Arrays.MAX_ARRAY_SIZE, 2L * length));
}

private final void reduce() {
    final int size = size();
    if (length > INITIAL_CAPACITY && size <= length / 4) resize(size, length / 2);
}

@Override
public void enqueue(KEY_GENERIC_TYPE x) {

```

```

array[end++] = x;
if (end == length) end = 0;
if (end == start) expand();
}

/** Enqueues a new element as the first element (in dequeuing order) of the queue.
 * @param x the element to enqueue.
 */
public void enqueueFirst(KEY_GENERIC_TYPE x) {
    if (start == 0) start = length;
    array[--start] = x;
    if (end == start) expand();
}

@Override
public KEY_GENERIC_TYPE FIRST() {
    if (start == end) throw new NoSuchElementException();
    return array[start];
}

@Override
public KEY_GENERIC_TYPE LAST() {
    if (start == end) throw new NoSuchElementException();
    return array[(end == 0 ? length : end) - 1];
}

@Override
public void clear() {
    #if KEYS_REFERENCE
    if (start <= end) Arrays.fill(array, start, end, null);
    else {
        Arrays.fill(array, start, length, null);
        Arrays.fill(array, 0, end, null);
    }
    #endif
    start = end = 0;
}

/** Trims the queue to the smallest possible size. */
SUPPRESS_WARNINGS_KEY_UNCHECKED
public void trim() {
    final int size = size();
    final KEY_GENERIC_TYPE[] newArray =
    #if KEYS_PRIMITIVE
        new KEY_GENERIC_TYPE[size + 1];
    #else
        (KEY_GENERIC_TYPE[])new Object[size + 1];

```

```

#endif
if (start <= end) System.arraycopy(array, start, newArray, 0, end - start);
else {
    System.arraycopy(array, start, newArray, 0, length - start);
    System.arraycopy(array, 0, newArray, length - start, end);
}
start = 0;
length = (end = size) + 1;
array = newArray;
}

@Override
public int size() {
    final int apparentLength = end - start;
    return apparentLength >= 0 ? apparentLength : length + apparentLength;
}

private void writeObject(java.io.ObjectOutputStream s) throws java.io.IOException {
    s.defaultWriteObject();
    int size = size();
    s.writeInt(size);
    for(int i = start; size-- != 0;) {
        s.WRITE_KEY(array[i++]);
        if (i == length) i = 0;
    }
}

SUPPRESS_WARNINGS_KEY_UNCHECKED
private void readObject(java.io.ObjectInputStream s) throws java.io.IOException, ClassNotFoundException {
    s.defaultReadObject();
    end = s.readInt();
    array = KEY_GENERIC_ARRAY_CAST new KEY_TYPE[length = HashCommon.nextPowerOfTwo(end + 1)];
    for(int i = 0; i < end; i++) array[i] = KEY_GENERIC_CAST s.READ_KEY();
}
}
}

```

Found in path(s):

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/drv/ArrayFIFOQueue.drv

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2002-2017 Sebastiano Vigna

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

- * <http://www.apache.org/licenses/LICENSE-2.0>
- *
- * Unless required by applicable law or agreed to in writing, software
- * distributed under the License is distributed on an "AS IS" BASIS,
- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.
- *
- *
- *
- * For the sorting and binary search code:
- *
- * Copyright (C) 1999 CERN - European Organization for Nuclear Research.
- *
- * Permission to use, copy, modify, distribute and sell this software and
- * its documentation for any purpose is hereby granted without fee,
- * provided that the above copyright notice appear in all copies and that
- * both that copyright notice and this permission notice appear in
- * supporting documentation. CERN makes no representations about the
- * suitability of this software for any purpose. It is provided "as is"
- * without expressed or implied warranty.
- */

package PACKAGE;

```
import it.unimi.dsi.fastutil.Arrays;
import it.unimi.dsi.fastutil.Hash;
import java.util.Random;
import java.util.concurrent.ForkJoinPool;
import java.util.concurrent.RecursiveAction;
```

```
#if ! KEY_CLASS_Integer
import it.unimi.dsi.fastutil.ints.IntArrays;
#endif
```

```
#if KEYS_PRIMITIVE
```

```
#if ! KEY_CLASS_Boolean
import java.util.concurrent.ExecutorCompletionService;
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;
import java.util.concurrent.LinkedBlockingQueue;
import java.util.concurrent.atomic.AtomicInteger;
#endif
```

```
/** A class providing static methods and objects that do useful things with type-specific arrays.
```

```
*
```

```
* <p>In particular, the { @code forceCapacity()}, { @code ensureCapacity()}, { @code grow()},
```

```

* {@code trim()} and {@code setLength()} methods allow to handle
* arrays much like array lists. This can be very useful when efficiency (or
* syntactic simplicity) reasons make array lists unsuitable.
*
* <p>Note that {@link it.unimi.dsi.fastutil.io.BinIO} and {@link it.unimi.dsi.fastutil.io.TextIO}
* contain several methods make it possible to load and save arrays of primitive types as sequences
* of elements in {@link java.io.DataInput} format (i.e., not as objects) or as sequences of lines of text.
*
* <h2>Sorting</h2>
*
* <p>There are several sorting methods available. The main theme is that of letting you choose
* the sorting algorithm you prefer (i.e., trading stability of mergesort for no memory allocation in quicksort).
* Several algorithms provide a parallel version, that will use the {@linkplain Runtime#availableProcessors()}
number of cores available}.
* Some algorithms also provide an explicit <em>indirect</em> sorting facility, which makes it possible
* to sort an array using the values in another array as comparator.
*
* <p>All comparison-based algorithm have an implementation based on a type-specific comparator.
*
* <p>As a general rule, sequential radix sort is significantly faster than quicksort or mergesort, in particular
* on random-looking data. In
* the parallel case, up to a few cores parallel radix sort is still the fastest, but at some point quicksort
* exploits parallelism better.
*
* <p>If you are fine with not knowing exactly which algorithm will be run (in particular, not knowing exactly
whether a support array will be allocated),
* the dual-pivot parallel sorts in {@link java.util.Arrays}
* are about 50% faster than the classical single-pivot implementation used here.
*
* <p>In any case, if sorting time is important I suggest that you benchmark your sorting load
* with your data distribution and on your architecture.
*
* @see java.util.Arrays
*/

```

```

public final class ARRAYS {

```

```

#else

```

```

import java.util.Comparator;

```

```

/** A class providing static methods and objects that do useful things with type-specific arrays.

```

```

*

```

```

* In particular, the {@code ensureCapacity()}, {@code grow()},
* {@code trim()} and {@code setLength()} methods allow to handle
* arrays much like array lists. This can be very useful when efficiency (or
* syntactic simplicity) reasons make array lists unsuitable.
*

```

```

*

```

- * `<p>Warning:` if your array is not of type `{ @code Object[] }`,
- * `{ @link #ensureCapacity(Object[],int,int) }` and `{ @link #grow(Object[],int,int) }`
- * will use `{ @linkplain java.lang.reflect.Array#newInstance(Class,int) }` reflection
- * to preserve your array type. Reflection is *significantly slower* than using `{ @code new }`.
- * This phenomenon is particularly
- * evident in the first growth phases of an array reallocated with doubling (or similar) logic.

*

* `<h2>Sorting</h2>`

*

- * `<p>`There are several sorting methods available. The main theme is that of letting you choose
- * the sorting algorithm you prefer (i.e., trading stability of mergesort for no memory allocation in quicksort).
- * Several algorithms provide a parallel version, that will use the `{ @linkplain Runtime#availableProcessors() }` number of cores available}.

*

- * `<p>`All comparison-based algorithm have an implementation based on a type-specific comparator.

*

- * `<p>`If you are fine with not knowing exactly which algorithm will be run (in particular, not knowing exactly whether a support array will be allocated),

- * the dual-pivot parallel sorts in `{ @link java.util.Arrays }`

- * are about 50% faster than the classical single-pivot implementation used here.

*

- * `<p>`In any case, if sorting time is important I suggest that you benchmark your sorting load

- * with your data distribution and on your architecture.

*

- * `@see java.util.Arrays`

*/

```
public final class ARRAYS {
```

```
#endif
```

```
private ARRAYS() {}
```

```
/** A static, final, empty array. */
```

```
public static final KEY_TYPE[] EMPTY_ARRAY = {};
```

```
/** A static, final, empty array to be used as default array in allocations. An
```

```
 * object distinct from { @link #EMPTY_ARRAY } makes it possible to have different
```

```
 * behaviors depending on whether the user required an empty allocation, or we are
```

```
 * just lazily delaying allocation.
```

```
 *
```

```
 * @see java.util.ArrayList
```

```
 */
```

```
public static final KEY_TYPE[] DEFAULT_EMPTY_ARRAY = {};
```

```
#if KEY_CLASS_Object
```

```
/** Creates a new array using a the given one as prototype.
```

```
 *
```

```

* <p>This method returns a new array of the given length whose element
* are of the same class as of those of { @code prototype }. In case
* of an empty array, it tries to return { @link #EMPTY_ARRAY }, if possible.
*
* @param prototype an array that will be used to type the new one.
* @param length the length of the new array.
* @return a new array of given type and length.
*/

```

SUPPRESS_WARNINGS_KEY_UNCHECKED

```

private static <K> K[] newArray(final K[] prototype, final int length) {
    final Class<?> klass = prototype.getClass();
    if (klass == Object[].class) return (K[])(length == 0 ? EMPTY_ARRAY : new Object[length]);
    return (K[])java.lang.reflect.Array.newInstance(klass.getComponentType(), length);
}

```

```

#endifif

```

```

/** Forces an array to contain the given number of entries, preserving just a part of the array.

```

```

*
* @param array an array.
* @param length the new minimum length for this array.
* @param preserve the number of elements of the array that must be preserved in case a new allocation is
necessary.

```

```

* @return an array with { @code length } entries whose first { @code preserve }
* entries are the same as those of { @code array }.

```

```

*/

```

```

public static KEY_GENERIC KEY_GENERIC_TYPE[] forceCapacity(final KEY_GENERIC_TYPE[] array, final
int length, final int preserve) {

```

```

    final KEY_GENERIC_TYPE t[] =

```

```

#if KEY_CLASS_Object
    newArray(array, length);

```

```

#else

```

```

    new KEY_TYPE[length];

```

```

#endifif

```

```

    System.arraycopy(array, 0, t, 0, preserve);

```

```

    return t;

```

```

}

```

```

/** Ensures that an array can contain the given number of entries.

```

```

*
* <p>If you cannot foresee whether this array will need again to be
* enlarged, you should probably use { @code grow() } instead.

```

```

*
* @param array an array.
* @param length the new minimum length for this array.
* @return { @code array }, if it contains { @code length } entries or more; otherwise,
* an array with { @code length } entries whose first { @code array.length }
* entries are the same as those of { @code array }.

```

```

*/
public static KEY_GENERIC KEY_GENERIC_TYPE[] ensureCapacity(final KEY_GENERIC_TYPE[] array,
final int length) {
    return ensureCapacity(array, length, array.length);
}

```

/** Ensures that an array can contain the given number of entries, preserving just a part of the array.

```

*
* @param array an array.
* @param length the new minimum length for this array.
* @param preserve the number of elements of the array that must be preserved in case a new allocation is
necessary.
* @return { @code array}, if it can contain { @code length} entries or more; otherwise,
* an array with { @code length} entries whose first { @code preserve}
* entries are the same as those of { @code array}.
*/

```

```

public static KEY_GENERIC KEY_GENERIC_TYPE[] ensureCapacity(final KEY_GENERIC_TYPE[] array,
final int length, final int preserve) {
    return length > array.length ? forceCapacity(array, length, preserve) : array;
}

```

/** Grows the given array to the maximum between the given length and

```

* the current length increased by 50%, provided that the given
* length is larger than the current length.
*

```

```

* <p>If you want complete control on the array growth, you
* should probably use { @code ensureCapacity()} instead.
*

```

```

* @param array an array.
* @param length the new minimum length for this array.
* @return { @code array}, if it can contain { @code length}
* entries; otherwise, an array with
* max({ @code length},{ @code array.length}/&phi;) entries whose first
* { @code array.length} entries are the same as those of { @code array}.
* */

```

```

public static KEY_GENERIC KEY_GENERIC_TYPE[] grow(final KEY_GENERIC_TYPE[] array, final int
length) {
    return grow(array, length, array.length);
}

```

/** Grows the given array to the maximum between the given length and

```

* the current length increased by 50%, provided that the given
* length is larger than the current length, preserving just a part of the array.
*

```

```

* <p>If you want complete control on the array growth, you
* should probably use { @code ensureCapacity()} instead.
*

```

```

* @param array an array.
* @param length the new minimum length for this array.
* @param preserve the number of elements of the array that must be preserved in case a new allocation is
necessary.
* @return { @code array }, if it can contain { @code length }
* entries; otherwise, an array with
* max({ @code length }, { @code array.length } / &phi;) entries whose first
* { @code preserve } entries are the same as those of { @code array }.
* */

```

```

public static KEY_GENERIC KEY_GENERIC_TYPE[] grow(final KEY_GENERIC_TYPE[] array, final int
length, final int preserve) {
    if (length > array.length) {
        final int newLength = (int) Math.max(Math.min((long) array.length + (array.length >> 1),
Arrays.MAX_ARRAY_SIZE), length);

        final KEY_GENERIC_TYPE t[] =
#if KEY_CLASS_Object
        newArray(array, newLength);
#else
        new KEY_TYPE[newLength];
#endif
        System.arraycopy(array, 0, t, 0, preserve);

        return t;
    }
    return array;
}

```

```

/** Trims the given array to the given length.
*
* @param array an array.
* @param length the new maximum length for the array.
* @return { @code array }, if it contains { @code length }
* entries or less; otherwise, an array with
* { @code length } entries whose entries are the same as
* the first { @code length } entries of { @code array }.
*
* */

```

```

public static KEY_GENERIC KEY_GENERIC_TYPE[] trim(final KEY_GENERIC_TYPE[] array, final int length)
{
    if (length >= array.length) return array;
    final KEY_GENERIC_TYPE t[] =
#if KEY_CLASS_Object
    newArray(array, length);
#else

```

```

    length == 0 ? EMPTY_ARRAY : new KEY_TYPE[length];
#endif
    System.arraycopy(array, 0, t, 0, length);
    return t;
}

/** Sets the length of the given array.
 *
 * @param array an array.
 * @param length the new length for the array.
 * @return { @code array }, if it contains exactly { @code length }
 * entries; otherwise, if it contains <em>more</em> than
 * { @code length } entries, an array with { @code length } entries
 * whose entries are the same as the first { @code length } entries of
 * { @code array }; otherwise, an array with { @code length } entries
 * whose first { @code array.length } entries are the same as those of
 * { @code array }.
 *
 */

public static KEY_GENERIC KEY_GENERIC_TYPE[] setLength(final KEY_GENERIC_TYPE[] array, final int
length) {
    if (length == array.length) return array;
    if (length < array.length) return trim(array, length);
    return ensureCapacity(array, length);
}

/** Returns a copy of a portion of an array.
 *
 * @param array an array.
 * @param offset the first element to copy.
 * @param length the number of elements to copy.
 * @return a new array containing { @code length } elements of { @code array } starting at { @code offset }.
 */

public static KEY_GENERIC KEY_GENERIC_TYPE[] copy(final KEY_GENERIC_TYPE[] array, final int offset,
final int length) {
    ensureOffsetLength(array, offset, length);
    final KEY_GENERIC_TYPE[] a =
#if KEY_CLASS_Object
    newArray(array, length);
#else
    length == 0 ? EMPTY_ARRAY : new KEY_TYPE[length];
#endif
    System.arraycopy(array, offset, a, 0, length);
    return a;
}

```

```

/** Returns a copy of an array.
 *
 * @param array an array.
 * @return a copy of { @code array }.
 */

public static KEY_GENERIC KEY_GENERIC_TYPE[] copy(final KEY_GENERIC_TYPE[] array) {
    return array.clone();
}

/** Fills the given array with the given value.
 *
 * @param array an array.
 * @param value the new value for all elements of the array.
 * @deprecated Please use the corresponding { @link java.util.Arrays } method.
 */

@Deprecated
public static KEY_GENERIC void fill(final KEY_GENERIC_TYPE[] array, final KEY_GENERIC_TYPE value) {
    int i = array.length;
    while(i-- != 0) array[i] = value;
}

/** Fills a portion of the given array with the given value.
 *
 * @param array an array.
 * @param from the starting index of the portion to fill (inclusive).
 * @param to the end index of the portion to fill (exclusive).
 * @param value the new value for all elements of the specified portion of the array.
 * @deprecated Please use the corresponding { @link java.util.Arrays } method.
 */

@Deprecated
public static KEY_GENERIC void fill(final KEY_GENERIC_TYPE[] array, final int from, int to, final
KEY_GENERIC_TYPE value) {
    ensureFromTo(array, from, to);
    if (from == 0) while(to-- != 0) array[to] = value;
    else for(int i = from; i < to; i++) array[i] = value;
}

/** Returns true if the two arrays are elementwise equal.
 *
 * @param a1 an array.
 * @param a2 another array.
 * @return true if the two arrays are of the same length, and their elements are equal.
 * @deprecated Please use the corresponding { @link java.util.Arrays } method, which is intrinsicified in recent JVMs.

```

```
*/
```

```
@Deprecated
```

```
public static KEY_GENERIC boolean equals(final KEY_GENERIC_TYPE[] a1, final KEY_GENERIC_TYPE  
a2[]) {  
    int i = a1.length;  
    if (i != a2.length) return false;  
    while(i-- != 0) if (! KEY_EQUALS(a1[i], a2[i])) return false;  
    return true;  
}
```

```
/** Ensures that a range given by its first (inclusive) and last (exclusive) elements fits an array.
```

```
*
```

```
* <p>This method may be used whenever an array range check is needed.
```

```
*
```

```
* @param a an array.
```

```
* @param from a start index (inclusive).
```

```
* @param to an end index (exclusive).
```

```
* @throws IllegalArgumentException if { @code from } is greater than { @code to }.
```

```
* @throws ArrayIndexOutOfBoundsException if { @code from } or { @code to } are greater than the array length or  
negative.
```

```
*/
```

```
public static KEY_GENERIC void ensureFromTo(final KEY_GENERIC_TYPE[] a, final int from, final int to) {  
    Arrays.ensureFromTo(a.length, from, to);  
}
```

```
/** Ensures that a range given by an offset and a length fits an array.
```

```
*
```

```
* <p>This method may be used whenever an array range check is needed.
```

```
*
```

```
* @param a an array.
```

```
* @param offset a start index.
```

```
* @param length a length (the number of elements in the range).
```

```
* @throws IllegalArgumentException if { @code length } is negative.
```

```
* @throws ArrayIndexOutOfBoundsException if { @code offset } is negative or { @code offset }+{ @code length } is  
greater than the array length.
```

```
*/
```

```
public static KEY_GENERIC void ensureOffsetLength(final KEY_GENERIC_TYPE[] a, final int offset, final int  
length) {  
    Arrays.ensureOffsetLength(a.length, offset, length);  
}
```

```
/** Ensures that two arrays are of the same length.
```

```
*
```

```
* @param a an array.
```

```

* @param b another array.
* @throws IllegalArgumentException if the two argument arrays are not of the same length.
*/
public static KEY_GENERIC void ensureSameLength(final KEY_GENERIC_TYPE[] a, final
KEY_GENERIC_TYPE[] b) {
    if (a.length != b.length) throw new IllegalArgumentException("Array size mismatch: " + a.length + " != " +
b.length);
}

private static final int QUICKSORT_NO_REC = 16;
private static final int PARALLEL_QUICKSORT_NO_FORK = 8192;
private static final int QUICKSORT_MEDIAN_OF_9 = 128;
private static final int MERGESORT_NO_REC = 16;

/** Swaps two elements of an array.
*
* @param x an array.
* @param a a position in { @code x}.
* @param b another position in { @code x}.
*/
public static KEY_GENERIC void swap(final KEY_GENERIC_TYPE x[], final int a, final int b) {
    final KEY_GENERIC_TYPE t = x[a];
    x[a] = x[b];
    x[b] = t;
}

/** Swaps two sequences of elements of an array.
*
* @param x an array.
* @param a a position in { @code x}.
* @param b another position in { @code x}.
* @param n the number of elements to exchange starting at { @code a} and { @code b}.
*/
public static KEY_GENERIC void swap(final KEY_GENERIC_TYPE[] x, int a, int b, final int n) {
    for(int i = 0; i < n; i++, a++, b++) swap(x, a, b);
}

private static KEY_GENERIC int med3(final KEY_GENERIC_TYPE x[], final int a, final int b, final int c,
KEY_COMPARATOR KEY_GENERIC comp) {
    final int ab = comp.compare(x[a], x[b]);
    final int ac = comp.compare(x[a], x[c]);
    final int bc = comp.compare(x[b], x[c]);
    return (ab < 0 ?
(bc < 0 ? b : ac < 0 ? c : a) :
(bc > 0 ? b : ac > 0 ? c : a));
}

private static KEY_GENERIC void selectionSort(final KEY_GENERIC_TYPE[] a, final int from, final int to, final

```

```

KEY_COMPARATOR KEY_GENERIC comp) {
    for(int i = from; i < to - 1; i++) {
        int m = i;
        for(int j = i + 1; j < to; j++) if (comp.compare(a[j], a[m]) < 0) m = j;
        if (m != i) {
            final KEY_GENERIC_TYPE u = a[i];
            a[i] = a[m];
            a[m] = u;
        }
    }
}

private static KEY_GENERIC void insertionSort(final KEY_GENERIC_TYPE[] a, final int from, final int to, final
KEY_COMPARATOR KEY_GENERIC comp) {
    for (int i = from; ++i < to;) {
        KEY_GENERIC_TYPE t = a[i];
        int j = i;
        for (KEY_GENERIC_TYPE u = a[j - 1]; comp.compare(t, u) < 0; u = a[--j - 1]) {
            a[j] = u;
            if (from == j - 1) {
                --j;
                break;
            }
        }
        a[j] = t;
    }
}

/** Sorts the specified range of elements according to the order induced by the specified
 * comparator using quicksort.
 *
 * <p>The sorting algorithm is a tuned quicksort adapted from Jon L. Bentley and M. Douglas
 * McIlroy, &ldquo;Engineering a Sort Function&rdquo;, <i>Software: Practice and Experience</i>, 23(11), pages
 * 1249&minus;1265, 1993.
 *
 * <p>Note that this implementation does not allocate any object, contrarily to the implementation
 * used to sort primitive types in { @link java.util.Arrays }, which switches to mergesort on large inputs.
 *
 * @param x the array to be sorted.
 * @param from the index of the first element (inclusive) to be sorted.
 * @param to the index of the last element (exclusive) to be sorted.
 * @param comp the comparator to determine the sorting order.
 *
 */
public static KEY_GENERIC void quickSort(final KEY_GENERIC_TYPE[] x, final int from, final int to, final
KEY_COMPARATOR KEY_GENERIC comp) {

```

```

final int len = to - from;
// Selection sort on smallest arrays
if (len < QUICKSORT_NO_REC) {
    selectionSort(x, from, to, comp);
    return;
}

// Choose a partition element, v
int m = from + len / 2;
int l = from;
int n = to - 1;
if (len > QUICKSORT_MEDIAN_OF_9) { // Big arrays, pseudomedian of 9
    int s = len / 8;
    l = med3(x, l, l + s, l + 2 * s, comp);
    m = med3(x, m - s, m, m + s, comp);
    n = med3(x, n - 2 * s, n - s, n, comp);
}
m = med3(x, l, m, n, comp); // Mid-size, med of 3

final KEY_GENERIC_TYPE v = x[m];

// Establish Invariant: v* (<v)* (>v)* v*
int a = from, b = a, c = to - 1, d = c;
while(true) {
    int comparison;
    while (b <= c && (comparison = comp.compare(x[b], v)) <= 0) {
        if (comparison == 0) swap(x, a++, b);
        b++;
    }
    while (c >= b && (comparison = comp.compare(x[c], v)) >= 0) {
        if (comparison == 0) swap(x, c, d--);
        c--;
    }
    if (b > c) break;
    swap(x, b++, c--);
}

// Swap partition elements back to middle
int s;
s = Math.min(a - from, b - a);
swap(x, from, b - s, s);
s = Math.min(d - c, to - d - 1);
swap(x, b, to - s, s);

// Recursively sort non-partition-elements
if ((s = b - a) > 1) quickSort(x, from, from + s, comp);
if ((s = d - c) > 1) quickSort(x, to - s, to, comp);
}

```

```

/** Sorts an array according to the order induced by the specified
 * comparator using quicksort.
 *
 * <p>The sorting algorithm is a tuned quicksort adapted from Jon L. Bentley and M. Douglas
 * McIlroy, &ldquo;Engineering a Sort Function&rdquo;, <i>Software: Practice and Experience</i>, 23(11), pages
 * 1249&minus;1265, 1993.
 *
 * <p>Note that this implementation does not allocate any object, contrarily to the implementation
 * used to sort primitive types in { @link java.util.Arrays }, which switches to mergesort on large inputs.
 *
 * @param x the array to be sorted.
 * @param comp the comparator to determine the sorting order.
 */
public static KEY_GENERIC void quickSort(final KEY_GENERIC_TYPE[] x, final KEY_COMPARATOR
KEY_GENERIC comp) {
    quickSort(x, 0, x.length, comp);
}

```

```

protected static class ForkJoinQuickSortComp KEY_GENERIC extends RecursiveAction {
    private static final long serialVersionUID = 1L;
    private final int from;
    private final int to;
    private final KEY_GENERIC_TYPE[] x;
    private final KEY_COMPARATOR KEY_GENERIC comp;

    public ForkJoinQuickSortComp(final KEY_GENERIC_TYPE[] x , final int from , final int to, final
KEY_COMPARATOR KEY_GENERIC comp) {
        this.from = from;
        this.to = to;
        this.x = x;
        this.comp = comp;
    }

```

```

@Override
protected void compute() {
    final KEY_GENERIC_TYPE[] x = this.x;
    final int len = to - from;
    if (len < PARALLEL_QUICKSORT_NO_FORK) {
        quickSort(x, from, to, comp);
        return;
    }
    // Choose a partition element, v
    int m = from + len / 2;
    int l = from;
    int n = to - 1;

```

```

int s = len / 8;
l = med3(x, l, l + s, l + 2 * s, comp);
m = med3(x, m - s, m, m + s, comp);
n = med3(x, n - 2 * s, n - s, n, comp);
m = med3(x, l, m, n, comp);
final KEY_GENERIC_TYPE v = x[m];
// Establish Invariant: v* (<v)* (>v)* v*
int a = from, b = a, c = to - 1, d = c;
while (true) {
    int comparison;
    while (b <= c && (comparison = comp.compare(x[b], v)) <= 0) {
        if (comparison == 0) swap(x, a++, b);
        b++;
    }
    while (c >= b && (comparison = comp.compare(x[c], v)) >= 0) {
        if (comparison == 0) swap(x, c, d--);
        c--;
    }
    if (b > c) break;
    swap(x, b++, c--);
}
// Swap partition elements back to middle
int t;
s = Math.min(a - from, b - a);
swap(x, from, b - s, s);
s = Math.min(d - c, to - d - 1);
swap(x, b, to - s, s);
// Recursively sort non-partition-elements
s = b - a;
t = d - c;
if (s > 1 && t > 1) invokeAll(new ForkJoinQuickSortComp KEY_GENERIC_DIAMOND(x, from, from + s,
comp), new ForkJoinQuickSortComp KEY_GENERIC_DIAMOND(x, to - t, to, comp));
else if (s > 1) invokeAll(new ForkJoinQuickSortComp KEY_GENERIC_DIAMOND(x, from, from + s, comp));
else invokeAll(new ForkJoinQuickSortComp KEY_GENERIC_DIAMOND(x, to - t, to, comp));
}
}

/** Sorts the specified range of elements according to the order induced by the specified
 * comparator using a parallel quicksort.
 *
 * <p>The sorting algorithm is a tuned quicksort adapted from Jon L. Bentley and M. Douglas
 * McIlroy, &ldquo;Engineering a Sort Function&rdquo;, <i>Software: Practice and Experience</i>, 23(11), pages
 * 1249&minus;1265, 1993.
 *
 * <p>This implementation uses a { @link ForkJoinPool} executor service with
 * { @link Runtime#availableProcessors()} parallel threads.
 *
 * @param x the array to be sorted.

```

```

* @param from the index of the first element (inclusive) to be sorted.
* @param to the index of the last element (exclusive) to be sorted.
* @param comp the comparator to determine the sorting order.
*/
public static KEY_GENERIC void parallelQuickSort(final KEY_GENERIC_TYPE[] x, final int from, final int to,
final KEY_COMPARATOR KEY_GENERIC comp) {
    if (to - from < PARALLEL_QUICKSORT_NO_FORK) quickSort(x, from, to, comp);
    else {
        final ForkJoinPool pool = new ForkJoinPool(Runtime.getRuntime().availableProcessors());
        pool.invoke(new ForkJoinQuickSortComp KEY_GENERIC_DIAMOND(x, from, to, comp));
        pool.shutdown();
    }
}

/** Sorts an array according to the order induced by the specified
* comparator using a parallel quicksort.
*
* <p>The sorting algorithm is a tuned quicksort adapted from Jon L. Bentley and M. Douglas
* McIlroy, &ldquo;Engineering a Sort Function&rdquo;, <i>Software: Practice and Experience</i>, 23(11), pages
* 1249&minus;1265, 1993.
*
* <p>This implementation uses a { @link ForkJoinPool } executor service with
* { @link Runtime#availableProcessors() } parallel threads.
*
* @param x the array to be sorted.
* @param comp the comparator to determine the sorting order.
*/
public static KEY_GENERIC void parallelQuickSort(final KEY_GENERIC_TYPE[] x, final
KEY_COMPARATOR KEY_GENERIC comp) {
    parallelQuickSort(x, 0, x.length, comp);
}

```

SUPPRESS_WARNINGS_KEY_UNCHECKED

```

private static KEY_GENERIC int med3(final KEY_GENERIC_TYPE x[], final int a, final int b, final int c) {
    final int ab = KEY_CMP(x[a], x[b]);
    final int ac = KEY_CMP(x[a], x[c]);
    final int bc = KEY_CMP(x[b], x[c]);
    return (ab < 0 ?
        (bc < 0 ? b : ac < 0 ? c : a) :
        (bc > 0 ? b : ac > 0 ? c : a));
}

```

SUPPRESS_WARNINGS_KEY_UNCHECKED

```

private static KEY_GENERIC void selectionSort(final KEY_GENERIC_TYPE[] a, final int from, final int to) {
    for(int i = from; i < to - 1; i++) {
        int m = i;
        for(int j = i + 1; j < to; j++) if (KEY_LESS(a[j], a[m])) m = j;
    }
}

```

```

if (m != i) {
    final KEY_GENERIC_TYPE u = a[i];
    a[i] = a[m];
    a[m] = u;
}
}
}

```

SUPPRESS_WARNINGS_KEY_UNCHECKED

```

private static KEY_GENERIC void insertionSort(final KEY_GENERIC_TYPE[] a, final int from, final int to) {
    for (int i = from; ++i < to;) {
        KEY_GENERIC_TYPE t = a[i];
        int j = i;
        for (KEY_GENERIC_TYPE u = a[j - 1]; KEY_LESS(t, u); u = a[--j - 1]) {
            a[j] = u;
            if (from == j - 1) {
                --j;
                break;
            }
        }
        a[j] = t;
    }
}

```

/** Sorts the specified range of elements according to the natural ascending order using quicksort.

*

* <p>The sorting algorithm is a tuned quicksort adapted from Jon L. Bentley and M. Douglas

* McIlroy, “Engineering a Sort Function”, <i>Software: Practice and Experience</i>, 23(11), pages

* 1249−1265, 1993.

*

* <p>Note that this implementation does not allocate any object, contrarily to the implementation

* used to sort primitive types in { @link java.util.Arrays}, which switches to mergesort on large inputs.

*

* @param x the array to be sorted.

* @param from the index of the first element (inclusive) to be sorted.

* @param to the index of the last element (exclusive) to be sorted.

*/

SUPPRESS_WARNINGS_KEY_UNCHECKED

```

public static KEY_GENERIC void quickSort(final KEY_GENERIC_TYPE[] x, final int from, final int to) {
    final int len = to - from;
    // Selection sort on smallest arrays
    if (len < QUICKSORT_NO_REC) {
        selectionSort(x, from, to);
        return;
    }
}

```

// Choose a partition element, v

```

int m = from + len / 2;
int l = from;
int n = to - 1;
if (len > QUICKSORT_MEDIAN_OF_9) { // Big arrays, pseudomedian of 9
    int s = len / 8;
    l = med3(x, l, l + s, l + 2 * s);
    m = med3(x, m - s, m, m + s);
    n = med3(x, n - 2 * s, n - s, n);
}
m = med3(x, l, m, n); // Mid-size, med of 3

final KEY_GENERIC_TYPE v = x[m];

// Establish Invariant: v* (<v)* (>v)* v*
int a = from, b = a, c = to - 1, d = c;
while(true) {
    int comparison;
    while (b <= c && (comparison = KEY_CMP(x[b], v)) <= 0) {
        if (comparison == 0) swap(x, a++, b);
        b++;
    }
    while (c >= b && (comparison = KEY_CMP(x[c], v)) >= 0) {
        if (comparison == 0) swap(x, c, d--);
        c--;
    }
    if (b > c) break;
    swap(x, b++, c--);
}

// Swap partition elements back to middle
int s;
s = Math.min(a - from, b - a);
swap(x, from, b - s, s);
s = Math.min(d - c, to - d - 1);
swap(x, b, to - s, s);

// Recursively sort non-partition-elements
if ((s = b - a) > 1) quickSort(x, from, from + s);
if ((s = d - c) > 1) quickSort(x, to - s, to);
}

/** Sorts an array according to the natural ascending order using quicksort.
 *
 * <p>The sorting algorithm is a tuned quicksort adapted from Jon L. Bentley and M. Douglas
 * McIlroy, &ldquo;Engineering a Sort Function&rdquo;, <i>Software: Practice and Experience</i>, 23(11), pages
 * 1249&minus;1265, 1993.
 *
 * <p>Note that this implementation does not allocate any object, contrarily to the implementation

```

```

* used to sort primitive types in { @link java.util.Arrays}, which switches to mergesort on large inputs.
*
* @param x the array to be sorted.
*
*/

```

```

public static KEY_GENERIC void quickSort(final KEY_GENERIC_TYPE[] x) {
    quickSort(x, 0, x.length);
}

```

```

protected static class ForkJoinQuickSort KEY_GENERIC extends RecursiveAction {
    private static final long serialVersionUID = 1L;
    private final int from;
    private final int to;
    private final KEY_GENERIC_TYPE[] x;

```

```

    public ForkJoinQuickSort(final KEY_GENERIC_TYPE[] x, final int from, final int to) {
        this.from = from;
        this.to = to;
        this.x = x;
    }

```

```

@Override

```

```

SUPPRESS_WARNINGS_KEY_UNCHECKED

```

```

protected void compute() {
    final KEY_GENERIC_TYPE[] x = this.x;
    final int len = to - from;
    if (len < PARALLEL_QUICKSORT_NO_FORK) {
        quickSort(x, from, to);
        return;
    }
    // Choose a partition element, v
    int m = from + len / 2;
    int l = from;
    int n = to - 1;
    int s = len / 8;
    l = med3(x, l, l + s, l + 2 * s);
    m = med3(x, m - s, m, m + s);
    n = med3(x, n - 2 * s, n - s, n);
    m = med3(x, l, m, n);
    final KEY_GENERIC_TYPE v = x[m];
    // Establish Invariant: v* (<v)* (>v)* v*
    int a = from, b = a, c = to - 1, d = c;
    while (true) {
        int comparison;
        while (b <= c && (comparison = KEY_CMP(x[b], v)) <= 0) {
            if (comparison == 0) swap(x, a++, b);
            b++;
        }
    }
}

```

```

while (c >= b && (comparison = KEY_CMP(x[c], v)) >= 0) {
    if (comparison == 0) swap(x, c, d--);
    c--;
}
if (b > c) break;
swap(x, b++, c--);
}
// Swap partition elements back to middle
int t;
s = Math.min(a - from, b - a);
swap(x, from, b - s, s);
s = Math.min(d - c, to - d - 1);
swap(x, b, to - s, s);
// Recursively sort non-partition-elements
s = b - a;
t = d - c;
if (s > 1 && t > 1) invokeAll(new ForkJoinQuickSort KEY_GENERIC_DIAMOND(x, from, from + s), new
ForkJoinQuickSort KEY_GENERIC_DIAMOND(x, to - t, to));
else if (s > 1) invokeAll(new ForkJoinQuickSort KEY_GENERIC_DIAMOND(x, from, from + s));
else invokeAll(new ForkJoinQuickSort KEY_GENERIC_DIAMOND(x, to - t, to));
}
}

/** Sorts the specified range of elements according to the natural ascending order using a parallel quicksort.
 *
 * <p>The sorting algorithm is a tuned quicksort adapted from Jon L. Bentley and M. Douglas
 * McIlroy, &ldquo;Engineering a Sort Function&rdquo;, <i>Software: Practice and Experience</i>, 23(11), pages
 * 1249&minus;1265, 1993.
 *
 * <p>This implementation uses a { @link ForkJoinPool } executor service with
 * { @link Runtime#availableProcessors() } parallel threads.
 *
 * @param x the array to be sorted.
 * @param from the index of the first element (inclusive) to be sorted.
 * @param to the index of the last element (exclusive) to be sorted.
 */
public static KEY_GENERIC void parallelQuickSort(final KEY_GENERIC_TYPE[] x, final int from, final int to) {
    if (to - from < PARALLEL_QUICKSORT_NO_FORK) quickSort(x, from, to);
    else {
        final ForkJoinPool pool = new ForkJoinPool(Runtime.getRuntime().availableProcessors());
        pool.invoke(new ForkJoinQuickSort KEY_GENERIC_DIAMOND(x, from, to));
        pool.shutdown();
    }
}

/** Sorts an array according to the natural ascending order using a parallel quicksort.
 *
 * <p>The sorting algorithm is a tuned quicksort adapted from Jon L. Bentley and M. Douglas

```

* McIlroy, “Engineering a Sort Function”, <i>Software: Practice and Experience</i>, 23(11), pages
* 1249−1265, 1993.

*

* <p>This implementation uses a { @link ForkJoinPool } executor service with
* { @link Runtime#availableProcessors() } parallel threads.

*

* @param x the array to be sorted.

*

*/

```
public static KEY_GENERIC void parallelQuickSort(final KEY_GENERIC_TYPE[] x) {  
    parallelQuickSort(x, 0, x.length);  
}
```

SUPPRESS_WARNINGS_KEY_UNCHECKED

```
private static KEY_GENERIC int med3Indirect(final int perm[], final KEY_GENERIC_TYPE x[], final int a, final  
int b, final int c) {  
    final KEY_GENERIC_TYPE aa = x[perm[a]];  
    final KEY_GENERIC_TYPE bb = x[perm[b]];  
    final KEY_GENERIC_TYPE cc = x[perm[c]];  
    final int ab = KEY_CMP(aa, bb);  
    final int ac = KEY_CMP(aa, cc);  
    final int bc = KEY_CMP(bb, cc);  
    return (ab < 0 ?  
        (bc < 0 ? b : ac < 0 ? c : a) :  
        (bc > 0 ? b : ac > 0 ? c : a));  
}
```

SUPPRESS_WARNINGS_KEY_UNCHECKED

```
private static KEY_GENERIC void insertionSortIndirect(final int[] perm, final KEY_GENERIC_TYPE[] a, final int  
from, final int to) {  
    for (int i = from; ++i < to;) {  
        int t = perm[i];  
        int j = i;  
        for (int u = perm[j - 1]; KEY_LESS(a[t], a[u]); u = perm[--j - 1]) {  
            perm[j] = u;  
            if (from == j - 1) {  
                --j;  
                break;  
            }  
        }  
        perm[j] = t;  
    }  
}
```

/** Sorts the specified range of elements according to the natural ascending order using indirect quicksort.

*

* <p>The sorting algorithm is a tuned quicksort adapted from Jon L. Bentley and M. Douglas

* McIlroy, “Engineering a Sort Function”, <i>Software: Practice and Experience</i>, 23(11), pages
* 1249−1265, 1993.

*

* <p>This method implement an indirect sort. The elements of { @code perm } (which must
* be exactly the numbers in the interval { @code [0..perm.length) }) will be permuted so that
* { @code x[perm[i]] ≤ x[perm[i + 1]]}.

*

* <p>Note that this implementation does not allocate any object, contrarily to the implementation
* used to sort primitive types in { @link java.util.Arrays}, which switches to mergesort on large inputs.

*

* @param perm a permutation array indexing { @code x }.

* @param x the array to be sorted.

* @param from the index of the first element (inclusive) to be sorted.

* @param to the index of the last element (exclusive) to be sorted.

*/

SUPPRESS_WARNINGS_KEY_UNCHECKED

```
public static KEY_GENERIC void quickSortIndirect(final int[] perm, final KEY_GENERIC_TYPE[] x, final int  
from, final int to) {
```

```
    final int len = to - from;
```

```
    // Selection sort on smallest arrays
```

```
    if (len < QUICKSORT_NO_REC) {
```

```
        insertionSortIndirect(perm, x, from, to);
```

```
        return;
```

```
    }
```

```
    // Choose a partition element, v
```

```
    int m = from + len / 2;
```

```
    int l = from;
```

```
    int n = to - 1;
```

```
    if (len > QUICKSORT_MEDIAN_OF_9) { // Big arrays, pseudomedian of 9
```

```
        int s = len / 8;
```

```
        l = med3Indirect(perm, x, l, l + s, l + 2 * s);
```

```
        m = med3Indirect(perm, x, m - s, m, m + s);
```

```
        n = med3Indirect(perm, x, n - 2 * s, n - s, n);
```

```
    }
```

```
    m = med3Indirect(perm, x, l, m, n); // Mid-size, med of 3
```

```
    final KEY_GENERIC_TYPE v = x[perm[m]];
```

```
    // Establish Invariant: v* (<v)* (>v)* v*
```

```
    int a = from, b = a, c = to - 1, d = c;
```

```
    while(true) {
```

```
        int comparison;
```

```
        while (b <= c && (comparison = KEY_CMP(x[perm[b]], v)) <= 0) {
```

```
            if (comparison == 0) IntArrays.swap(perm, a++, b);
```

```
            b++;
```

```
        }
```

```

while (c >= b && (comparison = KEY_CMP(x[perm[c]], v)) >=0) {
    if (comparison == 0) IntArrays.swap(perm, c, d--);
    c--;
}
if (b > c) break;
IntArrays.swap(perm, b++, c--);
}

// Swap partition elements back to middle
int s;
s = Math.min(a - from, b - a);
IntArrays.swap(perm, from, b - s, s);
s = Math.min(d - c, to - d - 1);
IntArrays.swap(perm, b, to - s, s);

// Recursively sort non-partition-elements
if ((s = b - a) > 1) quickSortIndirect(perm, x, from, from + s);
if ((s = d - c) > 1) quickSortIndirect(perm, x, to - s, to);
}

/** Sorts an array according to the natural ascending order using indirect quicksort.
 *
 * <p>The sorting algorithm is a tuned quicksort adapted from Jon L. Bentley and M. Douglas
 * McIlroy, &ldquo;Engineering a Sort Function&rdquo;, <i>Software: Practice and Experience</i>, 23(11), pages
 * 1249&minus;1265, 1993.
 *
 * <p>This method implement an <em>indirect</em> sort. The elements of { @code perm } (which must
 * be exactly the numbers in the interval { @code [0..perm.length) }) will be permuted so that
 * { @code x[perm[i]] &le; x[perm[i + 1]]}.
 *
 * <p>Note that this implementation does not allocate any object, contrarily to the implementation
 * used to sort primitive types in { @link java.util.Arrays}, which switches to mergesort on large inputs.
 *
 * @param perm a permutation array indexing { @code x}.
 * @param x the array to be sorted.
 */
public static KEY_GENERIC void quickSortIndirect(final int perm[], final KEY_GENERIC_TYPE[] x) {
    quickSortIndirect(perm, x, 0, x.length);
}

protected static class ForkJoinQuickSortIndirect KEY_GENERIC extends RecursiveAction {
    private static final long serialVersionUID = 1L;
    private final int from;
    private final int to;
    private final int[] perm;
    private final KEY_GENERIC_TYPE[] x;

    public ForkJoinQuickSortIndirect(final int perm[], final KEY_GENERIC_TYPE[] x, final int from, final int to) {

```

```

this.from = from;
this.to = to;
this.x = x;
this.perm = perm;
}

```

@Override

```

SUPPRESS_WARNINGS_KEY_UNCHECKED
protected void compute() {
    final KEY_GENERIC_TYPE[] x = this.x;
    final int len = to - from;
    if (len < PARALLEL_QUICKSORT_NO_FORK) {
        quickSortIndirect(perm, x, from, to);
        return;
    }
    // Choose a partition element, v
    int m = from + len / 2;
    int l = from;
    int n = to - 1;
    int s = len / 8;
    l = med3Indirect(perm, x, l, l + s, l + 2 * s);
    m = med3Indirect(perm, x, m - s, m, m + s);
    n = med3Indirect(perm, x, n - 2 * s, n - s, n);
    m = med3Indirect(perm, x, l, m, n);
    final KEY_GENERIC_TYPE v = x[perm[m]];
    // Establish Invariant: v* (<v)* (>v)* v*
    int a = from, b = a, c = to - 1, d = c;
    while (true) {
        int comparison;
        while (b <= c && (comparison = KEY_CMP(x[perm[b]], v)) <= 0) {
            if (comparison == 0) IntArrays.swap(perm, a++, b);
            b++;
        }
        while (c >= b && (comparison = KEY_CMP(x[perm[c]], v)) >= 0) {
            if (comparison == 0) IntArrays.swap(perm, c, d--);
            c--;
        }
        if (b > c) break;
        IntArrays.swap(perm, b++, c--);
    }
    // Swap partition elements back to middle
    int t;
    s = Math.min(a - from, b - a);
    IntArrays.swap(perm, from, b - s, s);
    s = Math.min(d - c, to - d - 1);
    IntArrays.swap(perm, b, to - s, s);
    // Recursively sort non-partition-elements
    s = b - a;

```

```

    t = d - c;
    if (s > 1 && t > 1) invokeAll(new ForkJoinQuickSortIndirect KEY_GENERIC_DIAMOND(perm, x, from, from +
s), new ForkJoinQuickSortIndirect KEY_GENERIC_DIAMOND(perm, x, to - t, to));
    else if (s > 1) invokeAll(new ForkJoinQuickSortIndirect KEY_GENERIC_DIAMOND(perm, x, from, from + s));
    else invokeAll(new ForkJoinQuickSortIndirect KEY_GENERIC_DIAMOND(perm, x, to - t, to));
    }
}

/** Sorts the specified range of elements according to the natural ascending order using a parallel indirect quicksort.
 *
 * <p>The sorting algorithm is a tuned quicksort adapted from Jon L. Bentley and M. Douglas
 * McIlroy, &ldquo;Engineering a Sort Function&rdquo;, <i>Software: Practice and Experience</i>, 23(11), pages
 * 1249&minus;1265, 1993.
 *
 * <p>This method implement an <em>indirect</em> sort. The elements of { @code perm } (which must
 * be exactly the numbers in the interval { @code [0..perm.length) }) will be permuted so that
 * { @code x[perm[i]] &le; x[perm[i + 1]]}.
 *
 * <p>This implementation uses a { @link ForkJoinPool } executor service with
 * { @link Runtime#availableProcessors() } parallel threads.
 *
 * @param perm a permutation array indexing { @code x }.
 * @param x the array to be sorted.
 * @param from the index of the first element (inclusive) to be sorted.
 * @param to the index of the last element (exclusive) to be sorted.
 */
public static KEY_GENERIC void parallelQuickSortIndirect(final int[] perm, final KEY_GENERIC_TYPE[] x,
final int from, final int to) {
    if (to - from < PARALLEL_QUICKSORT_NO_FORK) quickSortIndirect(perm, x, from, to);
    else {
        final ForkJoinPool pool = new ForkJoinPool(Runtime.getRuntime().availableProcessors());
        pool.invoke(new ForkJoinQuickSortIndirect KEY_GENERIC_DIAMOND(perm, x, from, to));
        pool.shutdown();
    }
}

/** Sorts an array according to the natural ascending order using a parallel indirect quicksort.
 *
 * <p>The sorting algorithm is a tuned quicksort adapted from Jon L. Bentley and M. Douglas
 * McIlroy, &ldquo;Engineering a Sort Function&rdquo;, <i>Software: Practice and Experience</i>, 23(11), pages
 * 1249&minus;1265, 1993.
 *
 * <p>This method implement an <em>indirect</em> sort. The elements of { @code perm } (which must
 * be exactly the numbers in the interval { @code [0..perm.length) }) will be permuted so that
 * { @code x[perm[i]] &le; x[perm[i + 1]]}.
 *
 * <p>This implementation uses a { @link ForkJoinPool } executor service with
 * { @link Runtime#availableProcessors() } parallel threads.

```

```

*
* @param perm a permutation array indexing { @code x }.
* @param x the array to be sorted.
*
*/
public static KEY_GENERIC void parallelQuickSortIndirect(final int perm[], final KEY_GENERIC_TYPE[] x) {
    parallelQuickSortIndirect(perm, x, 0, x.length);
}

/** Stabilizes a permutation.
*
* <p>This method can be used to stabilize the permutation generated by an indirect sorting, assuming that
* initially the permutation array was in ascending order (e.g., the identity, as usually happens). This method
* scans the permutation, and for each non-singleton block of elements with the same associated values in { @code
x },
* permutes them in ascending order. The resulting permutation corresponds to a stable sort.
*
* <p>Usually combining an unstable indirect sort and this method is more efficient than using a stable sort,
* as most stable sort algorithms require a support array.
*
* <p>More precisely, assuming that { @code x[perm[i]] &le; x[perm[i + 1]] }, after
* stabilization we will also have that { @code x[perm[i]] = x[perm[i + 1]] } implies
* { @code perm[i] &le; perm[i + 1] }.
*
* @param perm a permutation array indexing { @code x } so that it is sorted.
* @param x the sorted array to be stabilized.
* @param from the index of the first element (inclusive) to be stabilized.
* @param to the index of the last element (exclusive) to be stabilized.
*/
public static KEY_GENERIC void stabilize(final int perm[], final KEY_GENERIC_TYPE[] x, final int from, final
int to) {
    int curr = from;
    for(int i = from + 1; i < to; i++) {
        if (x[perm[i]] != x[perm[curr]]) {
            if (i - curr > 1) IntArrays.parallelQuickSort(perm, curr, i);
            curr = i;
        }
    }
    if (to - curr > 1) IntArrays.parallelQuickSort(perm, curr, to);
}

/** Stabilizes a permutation.
*
* <p>This method can be used to stabilize the permutation generated by an indirect sorting, assuming that
* initially the permutation array was in ascending order (e.g., the identity, as usually happens). This method
* scans the permutation, and for each non-singleton block of elements with the same associated values in { @code
x },
* permutes them in ascending order. The resulting permutation corresponds to a stable sort.

```

```

*
* <p>Usually combining an unstable indirect sort and this method is more efficient than using a stable sort,
* as most stable sort algorithms require a support array.
*
* <p>More precisely, assuming that { @code x[perm[i]] &le; x[perm[i + 1]]}, after
* stabilization we will also have that { @code x[perm[i]] = x[perm[i + 1]]} implies
* { @code perm[i] &le; perm[i + 1]}.
*
* @param perm a permutation array indexing { @code x} so that it is sorted.
* @param x the sorted array to be stabilized.
*/
public static KEY_GENERIC void stabilize(final int perm[], final KEY_GENERIC_TYPE[] x) {
    stabilize(perm, x, 0, perm.length);
}

SUPPRESS_WARNINGS_KEY_UNCHECKED
private static KEY_GENERIC int med3(final KEY_GENERIC_TYPE x[], final KEY_GENERIC_TYPE[] y, final
int a, final int b, final int c) {
    int t;
    final int ab = (t = KEY_CMP(x[a], x[b])) == 0 ? KEY_CMP(y[a], y[b]) : t;
    final int ac = (t = KEY_CMP(x[a], x[c])) == 0 ? KEY_CMP(y[a], y[c]) : t;
    final int bc = (t = KEY_CMP(x[b], x[c])) == 0 ? KEY_CMP(y[b], y[c]) : t;
    return (ab < 0 ?
        (bc < 0 ? b : ac < 0 ? c : a) :
        (bc > 0 ? b : ac > 0 ? c : a));
}

private static KEY_GENERIC void swap(final KEY_GENERIC_TYPE x[], final KEY_GENERIC_TYPE[] y, final
int a, final int b) {
    final KEY_GENERIC_TYPE t = x[a];
    final KEY_GENERIC_TYPE u = y[a];
    x[a] = x[b];
    y[a] = y[b];
    x[b] = t;
    y[b] = u;
}

private static KEY_GENERIC void swap(final KEY_GENERIC_TYPE[] x, final KEY_GENERIC_TYPE[] y, int a,
int b, final int n) {
    for (int i = 0; i < n; i++, a++, b++) swap(x, y, a, b);
}

SUPPRESS_WARNINGS_KEY_UNCHECKED
private static KEY_GENERIC void selectionSort(final KEY_GENERIC_TYPE[] a, final KEY_GENERIC_TYPE[]
b, final int from, final int to) {
    for(int i = from; i < to - 1; i++) {
        int m = i, u;
        for(int j = i + 1; j < to; j++)

```

```

if ((u = KEY_CMP(a[j], a[m])) < 0 || u == 0 && KEY_LESS(b[j], b[m])) m = j;

if (m != i) {
    KEY_GENERIC_TYPE t = a[i];
    a[i] = a[m];
    a[m] = t;
    t = b[i];
    b[i] = b[m];
    b[m] = t;
}
}
}

/** Sorts the specified range of elements of two arrays according to the natural lexicographical
 * ascending order using quicksort.
 *
 * <p>The sorting algorithm is a tuned quicksort adapted from Jon L. Bentley and M. Douglas
 * McIlroy, &ldquo;Engineering a Sort Function&rdquo;, <i>Software: Practice and Experience</i>, 23(11), pages
 * 1249&minus;1265, 1993.
 *
 * <p>This method implements a <em>lexicographical</em> sorting of the arguments. Pairs of
 * elements in the same position in the two provided arrays will be considered a single key, and
 * permuted accordingly. In the end, either {@code x[i] &lt; x[i + 1]} or <code>x[i]
 * == x[i + 1]</code> and {@code y[i] &lt;= y[i + 1]}].
 *
 * @param x the first array to be sorted.
 * @param y the second array to be sorted.
 * @param from the index of the first element (inclusive) to be sorted.
 * @param to the index of the last element (exclusive) to be sorted.
 */

SUPPRESS_WARNINGS_KEY_UNCHECKED
public static KEY_GENERIC void quickSort(final KEY_GENERIC_TYPE[] x, final KEY_GENERIC_TYPE[] y,
final int from, final int to) {
    final int len = to - from;
    if (len < QUICKSORT_NO_REC) {
        selectionSort(x, y, from, to);
        return;
    }
    // Choose a partition element, v
    int m = from + len / 2;
    int l = from;
    int n = to - 1;
    if (len > QUICKSORT_MEDIAN_OF_9) { // Big arrays, pseudomedian of 9
        int s = len / 8;
        l = med3(x, y, l, l + s, l + 2 * s);
        m = med3(x, y, m - s, m, m + s);
        n = med3(x, y, n - 2 * s, n - s, n);
    }
}

```

```

    }
    m = med3(x, y, l, m, n); // Mid-size, med of 3
    final KEY_GENERIC_TYPE v = x[m], w = y[m];
    // Establish Invariant: v* (<v)* (>v)* v*
    int a = from, b = a, c = to - 1, d = c;
    while (true) {
        int comparison, t;
        while (b <= c && (comparison = (t = KEY_CMP(x[b], v)) == 0 ? KEY_CMP(y[b], w) : t) <= 0) {
            if (comparison == 0) swap(x, y, a++, b);
            b++;
        }
        while (c >= b && (comparison = (t = KEY_CMP(x[c], v)) == 0 ? KEY_CMP(y[c], w) : t) >= 0) {
            if (comparison == 0) swap(x, y, c, d--);
            c--;
        }
        if (b > c) break;
        swap(x, y, b++, c--);
    }
    // Swap partition elements back to middle
    int s;
    s = Math.min(a - from, b - a);
    swap(x, y, from, b - s, s);
    s = Math.min(d - c, to - d - 1);
    swap(x, y, b, to - s, s);
    // Recursively sort non-partition-elements
    if ((s = b - a) > 1) quickSort(x, y, from, from + s);
    if ((s = d - c) > 1) quickSort(x, y, to - s, to);
}

/** Sorts two arrays according to the natural lexicographical ascending order using quicksort.
 *
 * <p>The sorting algorithm is a tuned quicksort adapted from Jon L. Bentley and M. Douglas
 * McIlroy, &ldquo;Engineering a Sort Function&rdquo;, <i>Software: Practice and Experience</i>, 23(11), pages
 * 1249&minus;1265, 1993.
 *
 * <p>This method implements a <em>lexicographical</em> sorting of the arguments. Pairs of
 * elements in the same position in the two provided arrays will be considered a single key, and
 * permuted accordingly. In the end, either { @code x[i] &lt; x[i + 1]} or <code>x[i]
 * == x[i + 1]</code> and { @code y[i] &lt; y[i + 1]}.
 *
 * @param x the first array to be sorted.
 * @param y the second array to be sorted.
 */
public static KEY_GENERIC void quickSort(final KEY_GENERIC_TYPE[] x, final KEY_GENERIC_TYPE[] y)
{
    ensureSameLength(x, y);
    quickSort(x, y, 0, x.length);
}

```

```

protected static class ForkJoinQuickSort2 KEY_GENERIC extends RecursiveAction {
    private static final long serialVersionUID = 1L;
    private final int from;
    private final int to;
    private final KEY_GENERIC_TYPE[] x, y;

    public ForkJoinQuickSort2(final KEY_GENERIC_TYPE[] x, final KEY_GENERIC_TYPE[] y, final int from ,
final int to) {
        this.from = from;
        this.to = to;
        this.x = x;
        this.y = y;
    }

    @Override
    SUPPRESS_WARNINGS_KEY_UNCHECKED
    protected void compute() {
        final KEY_GENERIC_TYPE[] x = this.x;
        final KEY_GENERIC_TYPE[] y = this.y;
        final int len = to - from;
        if (len < PARALLEL_QUICKSORT_NO_FORK) {
            quickSort(x, y, from, to);
            return;
        }
        // Choose a partition element, v
        int m = from + len / 2;
        int l = from;
        int n = to - 1;
        int s = len / 8;
        l = med3(x, y, l, l + s, l + 2 * s);
        m = med3(x, y, m - s, m, m + s);
        n = med3(x, y, n - 2 * s, n - s, n);
        m = med3(x, y, l, m, n);
        final KEY_GENERIC_TYPE v = x[m], w = y[m];
        // Establish Invariant: v* (<v)* (>v)* v*
        int a = from, b = a, c = to - 1, d = c;
        while (true) {
            int comparison, t;
            while (b <= c && (comparison = (t = KEY_CMP(x[b], v)) == 0 ? KEY_CMP(y[b], w) : t) <= 0) {
                if (comparison == 0) swap(x, a++, b);
                b++;
            }
            while (c >= b && (comparison = (t = KEY_CMP(x[c], v)) == 0 ? KEY_CMP(y[c], w) : t) >= 0) {
                if (comparison == 0) swap(x, y, c, d--);
                c--;
            }
            if (b > c) break;
        }
    }
}

```

```

    swap(x, y, b++, c--);
}
// Swap partition elements back to middle
int t;
s = Math.min(a - from, b - a);
swap(x, y, from, b - s, s);
s = Math.min(d - c, to - d - 1);
swap(x, y, b, to - s, s);
s = b - a;
t = d - c;
// Recursively sort non-partition-elements
if (s > 1 && t > 1) invokeAll(new ForkJoinQuickSort2 KEY_GENERIC_DIAMOND(x, y, from, from + s), new
ForkJoinQuickSort2 KEY_GENERIC_DIAMOND(x, y, to - t, to));
else if (s > 1) invokeAll(new ForkJoinQuickSort2 KEY_GENERIC_DIAMOND(x, y, from, from + s));
else invokeAll(new ForkJoinQuickSort2 KEY_GENERIC_DIAMOND(x, y, to - t, to));
}
}

/** Sorts the specified range of elements of two arrays according to the natural lexicographical
 * ascending order using a parallel quicksort.
 *
 * <p>The sorting algorithm is a tuned quicksort adapted from Jon L. Bentley and M. Douglas
 * McIlroy, &ldquo;Engineering a Sort Function&rdquo;, <i>Software: Practice and Experience</i>, 23(11), pages
 * 1249&minus;1265, 1993.
 *
 * <p>This method implements a <em>lexicographical</em> sorting of the arguments. Pairs of
 * elements in the same position in the two provided arrays will be considered a single key, and
 * permuted accordingly. In the end, either { @code x[i] &lt; x[i + 1]} or <code>x[i]
 * == x[i + 1]</code> and { @code y[i] &lt;= y[i + 1]}.
 *
 * <p>This implementation uses a { @link ForkJoinPool} executor service with
 * { @link Runtime#availableProcessors()} parallel threads.
 *
 * @param x the first array to be sorted.
 * @param y the second array to be sorted.
 * @param from the index of the first element (inclusive) to be sorted.
 * @param to the index of the last element (exclusive) to be sorted.
 */
public static KEY_GENERIC void parallelQuickSort(final KEY_GENERIC_TYPE[] x, final
KEY_GENERIC_TYPE[] y, final int from, final int to) {
    if (to - from < PARALLEL_QUICKSORT_NO_FORK) quickSort(x, y, from, to);
    final ForkJoinPool pool = new ForkJoinPool(Runtime.getRuntime().availableProcessors());
    pool.invoke(new ForkJoinQuickSort2 KEY_GENERIC_DIAMOND(x, y, from, to));
    pool.shutdown();
}

/** Sorts two arrays according to the natural lexicographical
 * ascending order using a parallel quicksort.

```

```

*
* <p>The sorting algorithm is a tuned quicksort adapted from Jon L. Bentley and M. Douglas
* McIlroy, &ldquo;Engineering a Sort Function&rdquo;, <i>Software: Practice and Experience</i>, 23(11), pages
* 1249&minus;1265, 1993.
*
* <p>This method implements a <em>lexicographical</em> sorting of the arguments. Pairs of
* elements in the same position in the two provided arrays will be considered a single key, and
* permuted accordingly. In the end, either { @code x[i] &lt; x[i + 1]} or <code>x[i]
* == x[i + 1]</code> and { @code y[i] &le; y[i + 1]}.
*
* <p>This implementation uses a { @link ForkJoinPool} executor service with
* { @link Runtime#availableProcessors()} parallel threads.
*
* @param x the first array to be sorted.
* @param y the second array to be sorted.
*/
public static KEY_GENERIC void parallelQuickSort(final KEY_GENERIC_TYPE[] x, final
KEY_GENERIC_TYPE[] y) {
    ensureSameLength(x, y);
    parallelQuickSort(x, y, 0, x.length);
}

/** Sorts the specified range of elements according to the natural ascending order using mergesort, using a given
pre-filled support array.
*
* <p>This sort is guaranteed to be <i>stable</i>: equal elements will not be reordered as a result
* of the sort. Moreover, no support arrays will be allocated.
*
* @param a the array to be sorted.
* @param from the index of the first element (inclusive) to be sorted.
* @param to the index of the last element (exclusive) to be sorted.
* @param supp a support array containing at least { @code to} elements, and whose entries are identical to those
* of { @code a} in the specified range.
*/

SUPPRESS_WARNINGS_KEY_UNCHECKED
public static KEY_GENERIC void mergeSort(final KEY_GENERIC_TYPE a[], final int from, final int to, final
KEY_GENERIC_TYPE supp[]) {
    int len = to - from;

    // Insertion sort on smallest arrays
    if (len < MERGESORT_NO_REC) {
        insertionSort(a, from, to);
        return;
    }
}

```

```

// Recursively sort halves of a into supp
final int mid = (from + to) >>> 1;
mergeSort(supp, from, mid, a);
mergeSort(supp, mid, to, a);

// If list is already sorted, just copy from supp to a. This is an
// optimization that results in faster sorts for nearly ordered lists.
if (KEY_LESSEQ(supp[mid - 1], supp[mid])) {
    System.arraycopy(supp, from, a, from, len);
    return;
}

// Merge sorted halves (now in supp) into a
for(int i = from, p = from, q = mid; i < to; i++) {
    if (q >= to || p < mid && KEY_LESSEQ(supp[p], supp[q])) a[i] = supp[p++];
    else a[i] = supp[q++];
}
}

/** Sorts the specified range of elements according to the natural ascending order using mergesort.
 *
 * <p>This sort is guaranteed to be <i>stable</i>: equal elements will not be reordered as a result
 * of the sort. An array as large as { @code a } will be allocated by this method.
 *
 * @param a the array to be sorted.
 * @param from the index of the first element (inclusive) to be sorted.
 * @param to the index of the last element (exclusive) to be sorted.
 */
public static KEY_GENERIC void mergeSort(final KEY_GENERIC_TYPE a[], final int from, final int to) {
    mergeSort(a, from, to, a.clone());
}

/** Sorts an array according to the natural ascending order using mergesort.
 *
 * <p>This sort is guaranteed to be <i>stable</i>: equal elements will not be reordered as a result
 * of the sort. An array as large as { @code a } will be allocated by this method.
 *
 * @param a the array to be sorted.
 */
public static KEY_GENERIC void mergeSort(final KEY_GENERIC_TYPE a[]) {
    mergeSort(a, 0, a.length);
}

/** Sorts the specified range of elements according to the order induced by the specified
 * comparator using mergesort, using a given pre-filled support array.
 *
 * <p>This sort is guaranteed to be <i>stable</i>: equal elements will not be reordered as a result
 * of the sort. Moreover, no support arrays will be allocated.

```

```

* @param a the array to be sorted.
* @param from the index of the first element (inclusive) to be sorted.
* @param to the index of the last element (exclusive) to be sorted.
* @param comp the comparator to determine the sorting order.
* @param supp a support array containing at least { @code to } elements, and whose entries are identical to those
* of { @code a } in the specified range.
*/
public static KEY_GENERIC void mergeSort(final KEY_GENERIC_TYPE a[], final int from, final int to,
KEY_COMPARATOR KEY_GENERIC comp, final KEY_GENERIC_TYPE supp[]) {
    int len = to - from;

    // Insertion sort on smallest arrays
    if (len < MERGESORT_NO_REC) {
        insertionSort(a, from, to, comp);
        return;
    }

    // Recursively sort halves of a into supp
    final int mid = (from + to) >>> 1;
    mergeSort(supp, from, mid, comp, a);
    mergeSort(supp, mid, to, comp, a);

    // If list is already sorted, just copy from supp to a. This is an
    // optimization that results in faster sorts for nearly ordered lists.
    if (comp.compare(supp[mid - 1], supp[mid]) <= 0) {
        System.arraycopy(supp, from, a, from, len);
        return;
    }

    // Merge sorted halves (now in supp) into a
    for(int i = from, p = from, q = mid; i < to; i++) {
        if (q >= to || p < mid && comp.compare(supp[p], supp[q]) <= 0) a[i] = supp[p++];
        else a[i] = supp[q++];
    }
}

/** Sorts the specified range of elements according to the order induced by the specified
* comparator using mergesort.
*
* <p>This sort is guaranteed to be <i>stable</i>: equal elements will not be reordered as a result
* of the sort. An array as large as { @code a } will be allocated by this method.
*
* @param a the array to be sorted.
* @param from the index of the first element (inclusive) to be sorted.
* @param to the index of the last element (exclusive) to be sorted.
* @param comp the comparator to determine the sorting order.
*/

```

```

public static KEY_GENERIC void mergeSort(final KEY_GENERIC_TYPE a[], final int from, final int to,
KEY_COMPARATOR KEY_GENERIC comp) {
    mergeSort(a, from, to, comp, a.clone());
}

```

```

/** Sorts an array according to the order induced by the specified
 * comparator using mergesort.
 *
 * <p>This sort is guaranteed to be <i>stable</i>: equal elements will not be reordered as a result
 * of the sort. An array as large as { @code a } will be allocated by this method.

```

```

 * @param a the array to be sorted.
 * @param comp the comparator to determine the sorting order.
 */

```

```

public static KEY_GENERIC void mergeSort(final KEY_GENERIC_TYPE a[], KEY_COMPARATOR
KEY_GENERIC comp) {
    mergeSort(a, 0, a.length, comp);
}

```

```

#if ! KEY_CLASS_Boolean

```

```

/**
 * Searches a range of the specified array for the specified value using
 * the binary search algorithm. The range must be sorted prior to making this call.
 * If it is not sorted, the results are undefined. If the range contains multiple elements with
 * the specified value, there is no guarantee which one will be found.
 *
 * @param a the array to be searched.
 * @param from the index of the first element (inclusive) to be searched.
 * @param to the index of the last element (exclusive) to be searched.
 * @param key the value to be searched for.
 * @return index of the search key, if it is contained in the array;
 *         otherwise, { @code (-(<i>insertion point</i>) - 1)}. The <i>insertion
 *         point</i> is defined as the the point at which the value would
 *         be inserted into the array: the index of the first
 *         element greater than the key, or the length of the array, if all
 *         elements in the array are less than the specified key. Note
 *         that this guarantees that the return value will be &ge; 0 if
 *         and only if the key is found.
 * @see java.util.Arrays
 */

```

```

SUPPRESS_WARNINGS_KEY_UNCHECKED

```

```

public static KEY_GENERIC int binarySearch(final KEY_GENERIC_TYPE[] a, int from, int to, final
KEY_GENERIC_TYPE key) {
    KEY_GENERIC_TYPE midVal;
    to--;
    while (from <= to) {
        final int mid = (from + to) >>> 1;

```

```

    midVal = a[mid];
#if KEYS_PRIMITIVE
    if (midVal < key) from = mid + 1;
    else if (midVal > key) to = mid - 1;
    else return mid;
#else
    final int cmp = ((Comparable KEY_SUPER_GENERIC)midVal).compareTo(key);
    if (cmp < 0) from = mid + 1;
    else if (cmp > 0) to = mid - 1;
    else return mid;
#endif
}
return -(from + 1);
}

/**
 * Searches an array for the specified value using
 * the binary search algorithm. The range must be sorted prior to making this call.
 * If it is not sorted, the results are undefined. If the range contains multiple elements with
 * the specified value, there is no guarantee which one will be found.
 *
 * @param a the array to be searched.
 * @param key the value to be searched for.
 * @return index of the search key, if it is contained in the array;
 *         otherwise, { @code (-(<i>insertion point</i>) - 1)}. The <i>insertion
 *         point</i> is defined as the the point at which the value would
 *         be inserted into the array: the index of the first
 *         element greater than the key, or the length of the array, if all
 *         elements in the array are less than the specified key. Note
 *         that this guarantees that the return value will be  $\geq 0$  if
 *         and only if the key is found.
 * @see java.util.Arrays
 */
public static KEY_GENERIC int binarySearch(final KEY_GENERIC_TYPE[] a, final KEY_GENERIC_TYPE
key) {
    return binarySearch(a, 0, a.length, key);
}

/**
 * Searches a range of the specified array for the specified value using
 * the binary search algorithm and a specified comparator. The range must be sorted following the comparator prior
 * to making this call.
 * If it is not sorted, the results are undefined. If the range contains multiple elements with
 * the specified value, there is no guarantee which one will be found.
 *
 * @param a the array to be searched.
 * @param from the index of the first element (inclusive) to be searched.
 * @param to the index of the last element (exclusive) to be searched.

```

```

* @param key the value to be searched for.
* @param c a comparator.
* @return index of the search key, if it is contained in the array;
*         otherwise, { @code (-(<i>insertion point</i>) - 1)}. The <i>insertion
*         point</i> is defined as the the point at which the value would
*         be inserted into the array: the index of the first
*         element greater than the key, or the length of the array, if all
*         elements in the array are less than the specified key. Note
*         that this guarantees that the return value will be  $\geq 0$  if
*         and only if the key is found.
* @see java.util.Arrays
*/
public static KEY_GENERIC int binarySearch(final KEY_GENERIC_TYPE[] a, int from, int to, final
KEY_GENERIC_TYPE key, final KEY_COMPARATOR KEY_GENERIC c) {
    KEY_GENERIC_TYPE midVal;
    to--;
    while (from <= to) {
        final int mid = (from + to) >>> 1;
        midVal = a[mid];
        final int cmp = c.compare(midVal, key);
        if (cmp < 0) from = mid + 1;
        else if (cmp > 0) to = mid - 1;
        else return mid; // key found
    }
    return -(from + 1);
}

/**
* Searches an array for the specified value using
* the binary search algorithm and a specified comparator. The range must be sorted following the comparator prior
to making this call.
* If it is not sorted, the results are undefined. If the range contains multiple elements with
* the specified value, there is no guarantee which one will be found.
*
* @param a the array to be searched.
* @param key the value to be searched for.
* @param c a comparator.
* @return index of the search key, if it is contained in the array;
*         otherwise, { @code (-(<i>insertion point</i>) - 1)}. The <i>insertion
*         point</i> is defined as the the point at which the value would
*         be inserted into the array: the index of the first
*         element greater than the key, or the length of the array, if all
*         elements in the array are less than the specified key. Note
*         that this guarantees that the return value will be  $\geq 0$  if
*         and only if the key is found.
* @see java.util.Arrays
*/
public static KEY_GENERIC int binarySearch(final KEY_GENERIC_TYPE[] a, final KEY_GENERIC_TYPE

```

```

key, final KEY_COMPARATOR KEY_GENERIC c) {
    return binarySearch(a, 0, a.length, key, c);
}

#if KEYS_PRIMITIVE
/** The size of a digit used during radix sort (must be a power of 2). */
private static final int DIGIT_BITS = 8;
/** The mask to extract a digit of { @link #DIGIT_BITS} bits. */
private static final int DIGIT_MASK = (1 << DIGIT_BITS) - 1;
/** The number of digits per element. */
private static final int DIGITS_PER_ELEMENT = KEY_CLASS.SIZE / DIGIT_BITS;
private static final int RADIXSORT_NO_REC = 1024;
private static final int PARALLEL_RADIXSORT_NO_FORK = 1024;

/** This method fixes negative numbers so that the combination exponent/significand is lexicographically sorted. */
#if KEY_CLASS_Double
private static final long fixDouble(final double d) {
    final long l = Double.doubleToLongBits(d);
    return l >= 0 ? l : l ^ 0x7FFFFFFFFFFFFFFFL;
}
#elif KEY_CLASS_Float
private static final int fixFloat(final float f) {
    final int i = Float.floatToIntBits(f);
    return i >= 0 ? i : i ^ 0x7FFFFFFF;
}
#endif

/** Sorts the specified array using radix sort.
 *
 * <p>The sorting algorithm is a tuned radix sort adapted from Peter M. McIlroy, Keith Bostic and M. Douglas
 * McIlroy, &ldquo;Engineering radix sort&rdquo;, <i>Computing Systems</i>, 6(1), pages 5&minus;27 (1993).
 *
 * <p>This implementation is significantly faster than quicksort
 * already at small sizes (say, more than 10000 elements), but it can only
 * sort in ascending order.
 *
 * @param a the array to be sorted.
 */
public static void radixSort(final KEY_TYPE[] a) {
    radixSort(a, 0, a.length);
}

/** Sorts the specified range of an array using radix sort.
 *
 * <p>The sorting algorithm is a tuned radix sort adapted from Peter M. McIlroy, Keith Bostic and M. Douglas
 * McIlroy, &ldquo;Engineering radix sort&rdquo;, <i>Computing Systems</i>, 6(1), pages 5&minus;27 (1993).

```

```

*
* <p>This implementation is significantly faster than quicksort
* already at small sizes (say, more than 10000 elements), but it can only
* sort in ascending order.
*
* @param a the array to be sorted.
* @param from the index of the first element (inclusive) to be sorted.
* @param to the index of the last element (exclusive) to be sorted.
*/
public static void radixSort(final KEY_TYPE[] a, final int from, final int to) {
    if (to - from < RADIXSORT_NO_REC) {
        quickSort(a, from, to);
        return;
    }

    final int maxLevel = DIGITS_PER_ELEMENT - 1;

    final int stackSize = ((1 << DIGIT_BITS) - 1) * (DIGITS_PER_ELEMENT - 1) + 1;
    int stackPos = 0;
    final int[] offsetStack = new int[stackSize];
    final int[] lengthStack = new int[stackSize];
    final int[] levelStack = new int[stackSize];

    offsetStack[stackPos] = from;
    lengthStack[stackPos] = to - from;
    levelStack[stackPos++] = 0;

    final int[] count = new int[1 << DIGIT_BITS];
    final int[] pos = new int[1 << DIGIT_BITS];

    while(stackPos > 0) {
        final int first = offsetStack[--stackPos];
        final int length = lengthStack[stackPos];
        final int level = levelStack[stackPos];
        #if KEY_CLASS_Character
            final int signMask = 0;
        #else
            final int signMask = level % DIGITS_PER_ELEMENT == 0 ? 1 << DIGIT_BITS - 1 : 0;
        #endif
        final int shift = (DIGITS_PER_ELEMENT - 1 - level % DIGITS_PER_ELEMENT) * DIGIT_BITS; // This is the
        shift that extract the right byte from a key

        // Count keys.
        for(int i = first + length; i-- != first;) count[(INT(KEY2LEXINT(a[i]) >>> shift & DIGIT_MASK ^ signMask)]++);
        // Compute cumulative distribution
        int lastUsed = -1;
        for (int i = 0, p = first; i < 1 << DIGIT_BITS; i++) {
            if (count[i] != 0) lastUsed = i;

```

```

    pos[i] = (p += count[i]);
}

final int end = first + length - count[lastUsed];
// i moves through the start of each block
for(int i = first, c = -1, d; i <= end; i += count[c], count[c] = 0) {
    KEY_TYPE t = a[i];
    c = INT(KEY2LEXINT(t) >>> shift & DIGIT_MASK ^ signMask);

    if (i < end) { // When all slots are OK, the last slot is necessarily OK.
        while ((d = --pos[c]) > i) {
            final KEY_TYPE z = t;
            t = a[d];
            a[d] = z;
            c = INT(KEY2LEXINT(t) >>> shift & DIGIT_MASK ^ signMask);
        }
        a[i] = t;
    }

    if (level < maxLevel && count[c] > 1) {
        if (count[c] < RADIXSORT_NO_REC) quickSort(a, i, i + count[c]);
        else {
            offsetStack[stackPos] = i;
            lengthStack[stackPos] = count[c];
            levelStack[stackPos++] = level + 1;
        }
    }
}

protected static final class Segment {
    protected final int offset, length, level;
    protected Segment(final int offset, final int length, final int level) {
        this.offset = offset;
        this.length = length;
        this.level = level;
    }

    @Override
    public String toString() { return "Segment [offset=" + offset + ", length=" + length + ", level=" + level + "]; }
}

protected static final Segment POISON_PILL = new Segment(-1, -1, -1);

/** Sorts the specified range of an array using parallel radix sort.
 *

```

```

* <p>The sorting algorithm is a tuned radix sort adapted from Peter M. McIlroy, Keith Bostic and M. Douglas
* McIlroy, &ldquo;Engineering radix sort&rdquo;, <i>Computing Systems</i>, 6(1), pages 5&minus;27 (1993).
*
* <p>This implementation uses a pool of { @link Runtime#availableProcessors() } threads.
*
* @param a the array to be sorted.
* @param from the index of the first element (inclusive) to be sorted.
* @param to the index of the last element (exclusive) to be sorted.
*/
public static void parallelRadixSort(final KEY_TYPE[] a, final int from, final int to) {
    if (to - from < PARALLEL_RADIXSORT_NO_FORK) {
        quickSort(a, from, to);
        return;
    }
    final int maxLevel = DIGITS_PER_ELEMENT - 1;
    final LinkedBlockingQueue<Segment> queue = new LinkedBlockingQueue<>();
    queue.add(new Segment(from, to - from, 0));
    final AtomicInteger queueSize = new AtomicInteger(1);
    final int numberOfThreads = Runtime.getRuntime().availableProcessors();
    final ExecutorService executorService = Executors.newFixedThreadPool(numberOfThreads,
    Executors.defaultThreadFactory());
    final ExecutorCompletionService<Void> executorCompletionService = new
    ExecutorCompletionService<>(executorService);

    for(int j = numberOfThreads; j-- != 0;) executorCompletionService.submit(() -> {
        final int[] count = new int[1 << DIGIT_BITS];
        final int[] pos = new int[1 << DIGIT_BITS];

        for(;;) {
            if (queueSize.get() == 0) for(int i = numberOfThreads; i-- != 0;) queue.add(POISON_PILL);
            final Segment segment = queue.take();
            if (segment == POISON_PILL) return null;

            final int first = segment.offset;
            final int length = segment.length;
            final int level = segment.level;

            #if KEY_CLASS_Character
                final int signMask = 0;
            #else
                final int signMask = level % DIGITS_PER_ELEMENT == 0 ? 1 << DIGIT_BITS - 1 : 0;
            #endif

            final int shift = (DIGITS_PER_ELEMENT - 1 - level % DIGITS_PER_ELEMENT) * DIGIT_BITS; // This is the
            shift that extract the right byte from a key

            // Count keys.
            for(int i = first + length; i-- != first;) count[INT(KEY2LEXINT(a[i]) >>> shift & DIGIT_MASK ^ signMask)]++;
            // Compute cumulative distribution

```

```

int lastUsed = -1;
for(int i = 0, p = first; i < 1 << DIGIT_BITS; i++) {
    if (count[i] != 0) lastUsed = i;
    pos[i] = (p += count[i]);
}

final int end = first + length - count[lastUsed];
// i moves through the start of each block
for(int i = first, c = -1, d; i <= end; i += count[c], count[c] = 0) {
    KEY_TYPE t = a[i];
    c = INT(KEY2LEXINT(t) >>> shift & DIGIT_MASK ^ signMask);
    if (i < end) {
        while((d = --pos[c]) > i) {
            final KEY_TYPE z = t;
            t = a[d];
            a[d] = z;
            c = INT(KEY2LEXINT(t) >>> shift & DIGIT_MASK ^ signMask);
        }
        a[i] = t;
    }

    if (level < maxLevel && count[c] > 1) {
        if (count[c] < PARALLEL_RADIXSORT_NO_FORK) quickSort(a, i, i + count[c]);
        else {
            queueSize.incrementAndGet();
            queue.add(new Segment(i, count[c], level + 1));
        }
    }
    queueSize.decrementAndGet();
}
});

Throwable problem = null;
for(int i = numberOfThreads; i-- != 0;)
    try {
        executorCompletionService.take().get();
    }
    catch(Exception e) {
        problem = e.getCause(); // We keep only the last one. They will be logged anyway.
    }

executorService.shutdown();
if (problem != null) throw (problem instanceof RuntimeException) ? (RuntimeException)problem : new
RuntimeException(problem);
}

/** Sorts the specified array using parallel radix sort.

```

```

*
* <p>The sorting algorithm is a tuned radix sort adapted from Peter M. McIlroy, Keith Bostic and M. Douglas
* McIlroy, &ldquo;Engineering radix sort&rdquo;, <i>Computing Systems</i>, 6(1), pages 5&minus;27 (1993).
*
* <p>This implementation uses a pool of { @link Runtime#availableProcessors() } threads.
*
* @param a the array to be sorted.
*/
public static void parallelRadixSort(final KEY_TYPE[] a) {
    parallelRadixSort(a, 0, a.length);
}

/** Sorts the specified array using indirect radix sort.
*
* <p>The sorting algorithm is a tuned radix sort adapted from Peter M. McIlroy, Keith Bostic and M. Douglas
* McIlroy, &ldquo;Engineering radix sort&rdquo;, <i>Computing Systems</i>, 6(1), pages 5&minus;27 (1993).
*
* <p>This method implement an <em>indirect</em> sort. The elements of { @code perm } (which must
* be exactly the numbers in the interval { @code [0..perm.length) }) will be permuted so that
* { @code a[perm[i]] &le; a[perm[i + 1]] }.
*
* <p>This implementation will allocate, in the stable case, a support array as large as { @code perm } (note that the
stable
* version is slightly faster).
*
* @param perm a permutation array indexing { @code a }.
* @param a the array to be sorted.
* @param stable whether the sorting algorithm should be stable.
*/
public static void radixSortIndirect(final int[] perm, final KEY_TYPE[] a, final boolean stable) {
    radixSortIndirect(perm, a, 0, perm.length, stable);
}

/** Sorts the specified array using indirect radix sort.
*
* <p>The sorting algorithm is a tuned radix sort adapted from Peter M. McIlroy, Keith Bostic and M. Douglas
* McIlroy, &ldquo;Engineering radix sort&rdquo;, <i>Computing Systems</i>, 6(1), pages 5&minus;27 (1993).
*
* <p>This method implement an <em>indirect</em> sort. The elements of { @code perm } (which must
* be exactly the numbers in the interval { @code [0..perm.length) }) will be permuted so that
* { @code a[perm[i]] &le; a[perm[i + 1]] }.
*
* <p>This implementation will allocate, in the stable case, a support array as large as { @code perm } (note that the
stable
* version is slightly faster).
*
* @param perm a permutation array indexing { @code a }.
* @param a the array to be sorted.

```

```

* @param from the index of the first element of { @code perm } (inclusive) to be permuted.
* @param to the index of the last element of { @code perm } (exclusive) to be permuted.
* @param stable whether the sorting algorithm should be stable.
*/
public static void radixSortIndirect(final int[] perm, final KEY_TYPE[] a, final int from, final int to, final boolean
stable) {
    if (to - from < RADIXSORT_NO_REC) {
        insertionSortIndirect(perm, a, from, to);
        return;
    }

    final int maxLevel = DIGITS_PER_ELEMENT - 1;

    final int stackSize = ((1 << DIGIT_BITS) - 1) * (DIGITS_PER_ELEMENT - 1) + 1;
    int stackPos = 0;
    final int[] offsetStack = new int[stackSize];
    final int[] lengthStack = new int[stackSize];
    final int[] levelStack = new int[stackSize];

    offsetStack[stackPos] = from;
    lengthStack[stackPos] = to - from;
    levelStack[stackPos++] = 0;

    final int[] count = new int[1 << DIGIT_BITS];
    final int[] pos = new int[1 << DIGIT_BITS];
    final int[] support = stable ? new int[perm.length] : null;

    while(stackPos > 0) {
        final int first = offsetStack[--stackPos];
        final int length = lengthStack[stackPos];
        final int level = levelStack[stackPos];
        #if KEY_CLASS_Character
            final int signMask = 0;
        #else
            final int signMask = level % DIGITS_PER_ELEMENT == 0 ? 1 << DIGIT_BITS - 1 : 0;
        #endif

        final int shift = (DIGITS_PER_ELEMENT - 1 - level % DIGITS_PER_ELEMENT) * DIGIT_BITS; // This is the
shift that extract the right byte from a key

        // Count keys.
        for(int i = first + length; i-- != first;) count[(INT(KEY2LEXINT(a[perm[i]])) >>> shift & DIGIT_MASK ^
signMask)]++;
        // Compute cumulative distribution
        int lastUsed = -1;
        for (int i = 0, p = stable ? 0 : first; i < 1 << DIGIT_BITS; i++) {
            if (count[i] != 0) lastUsed = i;
            pos[i] = (p += count[i]);
        }
    }

```



```

/** Sorts the specified range of an array using parallel indirect radix sort.
 *
 * <p>The sorting algorithm is a tuned radix sort adapted from Peter M. McIlroy, Keith Bostic and M. Douglas
 * McIlroy, &ldquo;Engineering radix sort&rdquo;, <i>Computing Systems</i>, 6(1), pages 5&minus;27 (1993).
 *
 * <p>This method implement an <em>indirect</em> sort. The elements of { @code perm } (which must
 * be exactly the numbers in the interval { @code [0..perm.length) }) will be permuted so that
 * { @code a[perm[i]] &le; a[perm[i + 1]]}.
 *
 * <p>This implementation uses a pool of { @link Runtime#availableProcessors() } threads.
 *
 * @param perm a permutation array indexing { @code a }.
 * @param a the array to be sorted.
 * @param from the index of the first element (inclusive) to be sorted.
 * @param to the index of the last element (exclusive) to be sorted.
 * @param stable whether the sorting algorithm should be stable.
 */
public static void parallelRadixSortIndirect(final int perm[], final KEY_TYPE[] a, final int from, final int to, final
boolean stable) {
    if (to - from < PARALLEL_RADIXSORT_NO_FORK) {
        radixSortIndirect(perm, a, from, to, stable);
        return;
    }
    final int maxLevel = DIGITS_PER_ELEMENT - 1;
    final LinkedBlockingQueue<Segment> queue = new LinkedBlockingQueue<>();
    queue.add(new Segment(from, to - from, 0));
    final AtomicInteger queueSize = new AtomicInteger(1);
    final int numberOfThreads = Runtime.getRuntime().availableProcessors();
    final ExecutorService executorService = Executors.newFixedThreadPool(numberOfThreads,
Executors.defaultThreadFactory());
    final ExecutorCompletionService<Void> executorCompletionService = new
ExecutorCompletionService<>(executorService);
    final int[] support = stable ? new int[perm.length] : null;

    for(int j = numberOfThreads; j-- != 0;) executorCompletionService.submit(() -> {
        final int[] count = new int[1 << DIGIT_BITS];
        final int[] pos = new int[1 << DIGIT_BITS];

        for(;;) {
            if (queueSize.get() == 0) for(int i = numberOfThreads; i-- != 0;) queue.add(POISON_PILL);
            final Segment segment = queue.take();
            if (segment == POISON_PILL) return null;

            final int first = segment.offset;
            final int length = segment.length;
            final int level = segment.level;

```

```

#if KEY_CLASS_Character
    final int signMask = 0;
#else
    final int signMask = level % DIGITS_PER_ELEMENT == 0 ? 1 << DIGIT_BITS - 1 : 0;
#endif

    final int shift = (DIGITS_PER_ELEMENT - 1 - level % DIGITS_PER_ELEMENT) * DIGIT_BITS; // This is the
    shift that extract the right byte from a key

    // Count keys.
    for(int i = first + length; i-- != first;) count[(INT(KEY2LEXINT(a[perm[i]])) >>> shift & DIGIT_MASK ^
signMask)]++;
    // Compute cumulative distribution
    int lastUsed = -1;
    for (int i = 0, p = first; i < 1 << DIGIT_BITS; i++) {
        if (count[i] != 0) lastUsed = i;
        pos[i] = (p += count[i]);
    }

    if (stable) {
        for(int i = first + length; i-- != first;) support[--pos[(INT(KEY2LEXINT(a[perm[i]])) >>> shift & DIGIT_MASK ^
signMask)]] = perm[i];
        System.arraycopy(support, first, perm, first, length);
        for(int i = 0, p = first; i <= lastUsed; i++) {
            if (level < maxLevel && count[i] > 1) {
                if (count[i] < PARALLEL_RADIXSORT_NO_FORK) radixSortIndirect(perm, a, p, p + count[i], stable);
            } else {
                queueSize.incrementAndGet();
                queue.add(new Segment(p, count[i], level + 1));
            }
        }
        p += count[i];
    }
    java.util.Arrays.fill(count, 0);
}
else {
    final int end = first + length - count[lastUsed];
    // i moves through the start of each block
    for(int i = first, c = -1, d; i <= end; i += count[c], count[c] = 0) {
        int t = perm[i];
        c = INT(KEY2LEXINT(a[t]) >>> shift & DIGIT_MASK ^ signMask);

        if (i < end) { // When all slots are OK, the last slot is necessarily OK.
            while((d = --pos[c]) > i) {
                final int z = t;
                t = perm[d];
                perm[d] = z;
                c = INT(KEY2LEXINT(a[t]) >>> shift & DIGIT_MASK ^ signMask);
            }
        }
    }
}

```

```

    }
    perm[i] = t;
    }

    if (level < maxLevel && count[c] > 1) {
        if (count[c] < PARALLEL_RADIXSORT_NO_FORK) radixSortIndirect(perm, a, i, i + count[c], stable);
        else {
            queueSize.incrementAndGet();
            queue.add(new Segment(i, count[c], level + 1));
        }
    }
    }
    }
    queueSize.decrementAndGet();
    }
});

Throwable problem = null;
for(int i = numberOfThreads; i-- != 0;)
try {
    executorCompletionService.take().get();
}
catch(Exception e) {
    problem = e.getCause(); // We keep only the last one. They will be logged anyway.
}

executorService.shutdown();
if (problem != null) throw (problem instanceof RuntimeException) ? (RuntimeException)problem : new
RuntimeException(problem);
}

/** Sorts the specified array using parallel indirect radix sort.
 *
 * <p>The sorting algorithm is a tuned radix sort adapted from Peter M. McIlroy, Keith Bostic and M. Douglas
 * McIlroy, &ldquo;Engineering radix sort&rdquo;, <i>Computing Systems</i>, 6(1), pages 5&minus;27 (1993).
 *
 * <p>This method implement an <em>indirect</em> sort. The elements of { @code perm } (which must
 * be exactly the numbers in the interval { @code [0..perm.length) }) will be permuted so that
 * { @code a[perm[i]] &le; a[perm[i + 1]] }.
 *
 * <p>This implementation uses a pool of { @link Runtime#availableProcessors() } threads.
 *
 * @param perm a permutation array indexing { @code a }.
 * @param a the array to be sorted.
 * @param stable whether the sorting algorithm should be stable.
 */
public static void parallelRadixSortIndirect(final int perm[], final KEY_TYPE[] a, final boolean stable) {
    parallelRadixSortIndirect(perm, a, 0, a.length, stable);
}

```

```

}

/** Sorts the specified pair of arrays lexicographically using radix sort.
 * <p>The sorting algorithm is a tuned radix sort adapted from Peter M. McIlroy, Keith Bostic and M. Douglas
 * McIlroy, &ldquo;Engineering radix sort&rdquo;, <i>Computing Systems</i>, 6(1), pages 5&minus;27 (1993).
 *
 * <p>This method implements a <em>lexicographical</em> sorting of the arguments. Pairs of elements
 * in the same position in the two provided arrays will be considered a single key, and permuted
 * accordingly. In the end, either { @code a[i] &lt; a[i + 1]} or { @code a[i] == a[i + 1]} and { @code b[i] &le; b[i +
 * 1]}.
 *
 * @param a the first array to be sorted.
 * @param b the second array to be sorted.
 */

public static void radixSort(final KEY_TYPE[] a, final KEY_TYPE[] b) {
    ensureSameLength(a, b);
    radixSort(a, b, 0, a.length);
}

/** Sorts the specified range of elements of two arrays using radix sort.
 *
 * <p>The sorting algorithm is a tuned radix sort adapted from Peter M. McIlroy, Keith Bostic and M. Douglas
 * McIlroy, &ldquo;Engineering radix sort&rdquo;, <i>Computing Systems</i>, 6(1), pages 5&minus;27 (1993).
 *
 * <p>This method implements a <em>lexicographical</em> sorting of the arguments. Pairs of elements
 * in the same position in the two provided arrays will be considered a single key, and permuted
 * accordingly. In the end, either { @code a[i] &lt; a[i + 1]} or { @code a[i] == a[i + 1]} and { @code b[i] &le; b[i +
 * 1]}.
 *
 * @param a the first array to be sorted.
 * @param b the second array to be sorted.
 * @param from the index of the first element (inclusive) to be sorted.
 * @param to the index of the last element (exclusive) to be sorted.
 */

public static void radixSort(final KEY_TYPE[] a, final KEY_TYPE[] b, final int from, final int to) {
    if (to - from < RADIXSORT_NO_REC) {
        selectionSort(a, b, from, to);
        return;
    }

    final int layers = 2;
    final int maxLevel = DIGITS_PER_ELEMENT * layers - 1;

    final int stackSize = ((1 << DIGIT_BITS) - 1) * (layers * DIGITS_PER_ELEMENT - 1) + 1;
    int stackPos = 0;
    final int[] offsetStack = new int[stackSize];
    final int[] lengthStack = new int[stackSize];

```

```

final int[] levelStack = new int[stackSize];

offsetStack[stackPos] = from;
lengthStack[stackPos] = to - from;
levelStack[stackPos++] = 0;

final int[] count = new int[1 << DIGIT_BITS];
final int[] pos = new int[1 << DIGIT_BITS];

while(stackPos > 0) {
    final int first = offsetStack[--stackPos];
    final int length = lengthStack[stackPos];
    final int level = levelStack[stackPos];
    #if KEY_CLASS_Character
        final int signMask = 0;
    #else
        final int signMask = level % DIGITS_PER_ELEMENT == 0 ? 1 << DIGIT_BITS - 1 : 0;
    #endif

    final KEY_TYPE[] k = level < DIGITS_PER_ELEMENT ? a : b; // This is the key array
    final int shift = (DIGITS_PER_ELEMENT - 1 - level % DIGITS_PER_ELEMENT) * DIGIT_BITS; // This is the
    shift that extract the right byte from a key

    // Count keys.
    for(int i = first + length; i-- != first;) count[INT(KEY2LEXINT(k[i]) >>> shift & DIGIT_MASK ^ signMask)]++;

    // Compute cumulative distribution
    int lastUsed = -1;
    for (int i = 0, p = first; i < 1 << DIGIT_BITS; i++) {
        if (count[i] != 0) lastUsed = i;
        pos[i] = (p += count[i]);
    }

    final int end = first + length - count[lastUsed];
    // i moves through the start of each block
    for(int i = first, c = -1, d; i <= end; i += count[c], count[c] = 0) {
        KEY_TYPE t = a[i];
        KEY_TYPE u = b[i];
        c = INT(KEY2LEXINT(k[i]) >>> shift & DIGIT_MASK ^ signMask);

        if (i < end) { // When all slots are OK, the last slot is necessarily OK.
            while((d = --pos[c]) > i) {
                c = INT(KEY2LEXINT(k[d]) >>> shift & DIGIT_MASK ^ signMask);
                KEY_TYPE z = t;
                t = a[d];
                a[d] = z;
                z = u;
            }
        }
    }
}

```

```

    u = b[d];
    b[d] = z;
}
a[i] = t;
b[i] = u;

}

if (level < maxLevel && count[c] > 1) {
    if (count[c] < RADIXSORT_NO_REC) selectionSort(a, b, i, i + count[c]);
    else {
        offsetStack[stackPos] = i;
        lengthStack[stackPos] = count[c];
        levelStack[stackPos++] = level + 1;
    }
}
}
}
}

/** Sorts the specified range of elements of two arrays using a parallel radix sort.
 *
 * <p>The sorting algorithm is a tuned radix sort adapted from Peter M. McIlroy, Keith Bostic and M. Douglas
 * McIlroy, &ldquo;Engineering radix sort&rdquo;, <i>Computing Systems</i>, 6(1), pages 5&minus;27 (1993).
 *
 * <p>This method implements a <em>lexicographical</em> sorting of the arguments. Pairs of elements
 * in the same position in the two provided arrays will be considered a single key, and permuted
 * accordingly. In the end, either { @code a[i] &lt; a[i + 1]} or { @code a[i] == a[i + 1]} and { @code b[i] &le; b[i +
 * 1]}.
 *
 * <p>This implementation uses a pool of { @link Runtime#availableProcessors()} threads.
 *
 * @param a the first array to be sorted.
 * @param b the second array to be sorted.
 * @param from the index of the first element (inclusive) to be sorted.
 * @param to the index of the last element (exclusive) to be sorted.
 */
public static void parallelRadixSort(final KEY_TYPE[] a, final KEY_TYPE[] b, final int from, final int to) {
    if (to - from < PARALLEL_RADIXSORT_NO_FORK) {
        quickSort(a, b, from, to);
        return;
    }
    final int layers = 2;
    if (a.length != b.length) throw new IllegalArgumentException("Array size mismatch.");
    final int maxLevel = DIGITS_PER_ELEMENT * layers - 1;
    final LinkedBlockingQueue<Segment> queue = new LinkedBlockingQueue<>();
    queue.add(new Segment(from, to - from, 0));
    final AtomicInteger queueSize = new AtomicInteger(1);

```

```

final int numberOfThreads = Runtime.getRuntime().availableProcessors();
final ExecutorService executorService = Executors.newFixedThreadPool(numberOfThreads,
Executors.defaultThreadFactory());
final ExecutorCompletionService<Void> executorCompletionService = new
ExecutorCompletionService<>(executorService);
for (int j = numberOfThreads; j-- != 0;) executorCompletionService.submit(() -> {
    final int[] count = new int[1 << DIGIT_BITS];
    final int[] pos = new int[1 << DIGIT_BITS];
    for (;;) {
        if (queueSize.get() == 0) for (int i = numberOfThreads; i-- != 0;)
            queue.add(POISON_PILL);
        final Segment segment = queue.take();
        if (segment == POISON_PILL) return null;
        final int first = segment.offset;
        final int length = segment.length;
        final int level = segment.level;
        final int signMask = level % DIGITS_PER_ELEMENT == 0 ? 1 << DIGIT_BITS - 1 : 0;
        final KEY_TYPE[] k = level < DIGITS_PER_ELEMENT ? a : b; // This is the key array
        final int shift = (DIGITS_PER_ELEMENT - 1 - level % DIGITS_PER_ELEMENT) * DIGIT_BITS;
        // Count keys.
        for (int i = first + length; i-- != first;)
            count[INT(KEY2LEXINT(k[i]) >>> shift & DIGIT_MASK ^ signMask)]++;
        // Compute cumulative distribution
        int lastUsed = -1;
        for (int i = 0, p = first; i < 1 << DIGIT_BITS; i++) {
            if (count[i] != 0) lastUsed = i;
            pos[i] = (p += count[i]);
        }
        final int end = first + length - count[lastUsed];
        for (int i = first, c = -1, d; i <= end; i += count[c], count[c] = 0) {
            KEY_TYPE t = a[i];
            KEY_TYPE u = b[i];
            c = INT(KEY2LEXINT(k[i]) >>> shift & DIGIT_MASK ^ signMask);
            if (i < end) { // When all slots are OK, the last slot is necessarily OK.
                while ((d = --pos[c]) > i) {
                    c = INT(KEY2LEXINT(k[d]) >>> shift & DIGIT_MASK ^ signMask);
                    final KEY_TYPE z = t;
                    final KEY_TYPE w = u;
                    t = a[d];
                    u = b[d];
                    a[d] = z;
                    b[d] = w;
                }
                a[i] = t;
                b[i] = u;
            }
            if (level < maxLevel && count[c] > 1) {
                if (count[c] < PARALLEL_RADIXSORT_NO_FORK) quickSort(a, b, i, i + count[c]);
            }
        }
    }
}

```

```

    else {
        queueSize.incrementAndGet();
        queue.add(new Segment(i, count[c], level + 1));
    }
}
queueSize.decrementAndGet();
});
Throwable problem = null;
for (int i = numberOfThreads; i-- != 0;)
try {
    executorCompletionService.take().get();
}
catch (Exception e) {
    problem = e.getCause(); // We keep only the last one. They will be logged anyway.
}
executorService.shutdown();
if (problem != null) throw (problem instanceof RuntimeException) ? (RuntimeException)problem : new
RuntimeException(problem);
}

/** Sorts two arrays using a parallel radix sort.
 *
 * <p>The sorting algorithm is a tuned radix sort adapted from Peter M. McIlroy, Keith Bostic and M. Douglas
 * McIlroy, &ldquo;Engineering radix sort&rdquo;, <i>Computing Systems</i>, 6(1), pages 5&minus;27 (1993).
 *
 * <p>This method implements a <em>lexicographical</em> sorting of the arguments. Pairs of elements
 * in the same position in the two provided arrays will be considered a single key, and permuted
 * accordingly. In the end, either {@code a[i] < a[i + 1]} or {@code a[i] == a[i + 1]} and {@code b[i] <= b[i +
 * 1]}).
 *
 * <p>This implementation uses a pool of {@link Runtime#availableProcessors()} threads.
 *
 * @param a the first array to be sorted.
 * @param b the second array to be sorted.
 */
public static void parallelRadixSort(final KEY_TYPE[] a, final KEY_TYPE[] b) {
    ensureSameLength(a, b);
    parallelRadixSort(a, b, 0, a.length);
}

private static KEY_GENERIC void insertionSortIndirect(final int[] perm, final KEY_TYPE[] a, final KEY_TYPE[]
b, final int from, final int to) {
    for (int i = from; ++i < to;) {
        int t = perm[i];

```

```

int j = i;
for (int u = perm[j - 1]; KEY_LESS(a[t], a[u]) || KEY_CMP_EQ(a[t], a[u]) && KEY_LESS(b[t], b[u]); u = perm[--j
- 1]) {
    perm[j] = u;
    if (from == j - 1) {
        --j;
        break;
    }
}
perm[j] = t;
}
}

```

/** Sorts the specified pair of arrays lexicographically using indirect radix sort.

```

*
* <p>The sorting algorithm is a tuned radix sort adapted from Peter M. McIlroy, Keith Bostic and M. Douglas
* McIlroy, &ldquo;Engineering radix sort&rdquo;, <i>Computing Systems</i>, 6(1), pages 5&minus;27 (1993).
*
* <p>This method implement an <em>indirect</em> sort. The elements of { @code perm } (which must
* be exactly the numbers in the interval { @code [0..perm.length) }) will be permuted so that
* { @code a[perm[i]] &le; a[perm[i + 1]] } or { @code a[perm[i]] == a[perm[i + 1]] } and { @code b[perm[i]] &le;
b[perm[i + 1]] }.
*
* <p>This implementation will allocate, in the stable case, a further support array as large as { @code perm } (note
that the stable
* version is slightly faster).
*
* @param perm a permutation array indexing { @code a }.
* @param a the array to be sorted.
* @param b the second array to be sorted.
* @param stable whether the sorting algorithm should be stable.
*/

```

```

public static void radixSortIndirect(final int[] perm, final KEY_TYPE[] a, final KEY_TYPE[] b, final boolean
stable) {
    ensureSameLength(a, b);
    radixSortIndirect(perm, a, b, 0, a.length, stable);
}

```

/** Sorts the specified pair of arrays lexicographically using indirect radix sort.

```

*
* <p>The sorting algorithm is a tuned radix sort adapted from Peter M. McIlroy, Keith Bostic and M. Douglas
* McIlroy, &ldquo;Engineering radix sort&rdquo;, <i>Computing Systems</i>, 6(1), pages 5&minus;27 (1993).
*
* <p>This method implement an <em>indirect</em> sort. The elements of { @code perm } (which must
* be exactly the numbers in the interval { @code [0..perm.length) }) will be permuted so that
* { @code a[perm[i]] &le; a[perm[i + 1]] } or { @code a[perm[i]] == a[perm[i + 1]] } and { @code b[perm[i]] &le;
b[perm[i + 1]] }.
*

```

```

* <p>This implementation will allocate, in the stable case, a further support array as large as { @code perm } (note
that the stable
* version is slightly faster).
*
* @param perm a permutation array indexing { @code a }.
* @param a the array to be sorted.
* @param b the second array to be sorted.
* @param from the index of the first element of { @code perm } (inclusive) to be permuted.
* @param to the index of the last element of { @code perm } (exclusive) to be permuted.
* @param stable whether the sorting algorithm should be stable.
*/
public static void radixSortIndirect(final int[] perm, final KEY_TYPE[] a, final KEY_TYPE[] b, final int from, final
int to, final boolean stable) {
    if (to - from < RADIXSORT_NO_REC) {
        insertionSortIndirect(perm, a, b, from, to);
        return;
    }

    final int layers = 2;
    final int maxLevel = DIGITS_PER_ELEMENT * layers - 1;

    final int stackSize = ((1 << DIGIT_BITS) - 1) * (layers * DIGITS_PER_ELEMENT - 1) + 1;
    int stackPos = 0;
    final int[] offsetStack = new int[stackSize];
    final int[] lengthStack = new int[stackSize];
    final int[] levelStack = new int[stackSize];

    offsetStack[stackPos] = from;
    lengthStack[stackPos] = to - from;
    levelStack[stackPos++] = 0;

    final int[] count = new int[1 << DIGIT_BITS];
    final int[] pos = new int[1 << DIGIT_BITS];
    final int[] support = stable ? new int[perm.length] : null;

    while(stackPos > 0) {
        final int first = offsetStack[--stackPos];
        final int length = lengthStack[stackPos];
        final int level = levelStack[stackPos];
        #if KEY_CLASS_Character
            final int signMask = 0;
        #else
            final int signMask = level % DIGITS_PER_ELEMENT == 0 ? 1 << DIGIT_BITS - 1 : 0;
        #endif

        final KEY_TYPE[] k = level < DIGITS_PER_ELEMENT ? a : b; // This is the key array
        final int shift = (DIGITS_PER_ELEMENT - 1 - level % DIGITS_PER_ELEMENT) * DIGIT_BITS; // This is the
        shift that extract the right byte from a key

```

```

// Count keys.
for(int i = first + length; i-- != first;) count[INT(KEY2LEXINT(k[perm[i]]) >>> shift & DIGIT_MASK ^
signMask)]++;

// Compute cumulative distribution
int lastUsed = -1;
for (int i = 0, p = stable ? 0 : first; i < 1 << DIGIT_BITS; i++) {
    if (count[i] != 0) lastUsed = i;
    pos[i] = (p += count[i]);
}

if (stable) {
    for(int i = first + length; i-- != first;) support[--pos[INT(KEY2LEXINT(k[perm[i]]) >>> shift & DIGIT_MASK ^
signMask)]] = perm[i];
    System.arraycopy(support, 0, perm, first, length);
    for(int i = 0, p = first; i < 1 << DIGIT_BITS; i++) {
        if (level < maxLevel && count[i] > 1) {
            if (count[i] < RADIXSORT_NO_REC) insertionSortIndirect(perm, a, b, p, p + count[i]);
            else {
                offsetStack[stackPos] = p;
                lengthStack[stackPos] = count[i];
                levelStack[stackPos++] = level + 1;
            }
        }
        p += count[i];
    }
    java.util.Arrays.fill(count, 0);
}
else {
    final int end = first + length - count[lastUsed];
    // i moves through the start of each block
    for(int i = first, c = -1, d; i <= end; i += count[c], count[c] = 0) {
        int t = perm[i];
        c = INT(KEY2LEXINT(k[t]) >>> shift & DIGIT_MASK ^ signMask);

        if (i < end) { // When all slots are OK, the last slot is necessarily OK.
            while((d = --pos[c]) > i) {
                final int z = t;
                t = perm[d];
                perm[d] = z;
                c = INT(KEY2LEXINT(k[t]) >>> shift & DIGIT_MASK ^ signMask);
            }
            perm[i] = t;
        }

        if (level < maxLevel && count[c] > 1) {

```



```

* in the same position will be considered a single key, and permuted
* accordingly.
*
* @param a an array containing arrays of equal length to be sorted lexicographically in parallel.
*/
public static void radixSort(final KEY_TYPE[][] a) {
    radixSort(a, 0, a[0].length);
}

/** Sorts the specified array of arrays lexicographically using radix sort.
 *
 * <p>The sorting algorithm is a tuned radix sort adapted from Peter M. McIlroy, Keith Bostic and M. Douglas
 * McIlroy, &ldquo;Engineering radix sort&rdquo;, <i>Computing Systems</i>, 6(1), pages 5&minus;27 (1993).
 *
 * <p>This method implements a <em>lexicographical</em> sorting of the provided arrays. Tuples of elements
 * in the same position will be considered a single key, and permuted
 * accordingly.
 *
 * @param a an array containing arrays of equal length to be sorted lexicographically in parallel.
 * @param from the index of the first element (inclusive) to be sorted.
 * @param to the index of the last element (exclusive) to be sorted.
 */
public static void radixSort(final KEY_TYPE[][] a, final int from, final int to) {
    if (to - from < RADIXSORT_NO_REC) {
        selectionSort(a, from, to, 0);
        return;
    }

    final int layers = a.length;
    final int maxLevel = DIGITS_PER_ELEMENT * layers - 1;
    for(int p = layers, l = a[0].length; p-- != 0;) if (a[p].length != l) throw new IllegalArgumentException("The array of
index " + p + " has not the same length of the array of index 0.");

    final int stackSize = ((1 << DIGIT_BITS) - 1) * (layers * DIGITS_PER_ELEMENT - 1) + 1;
    int stackPos = 0;
    final int[] offsetStack = new int[stackSize];
    final int[] lengthStack = new int[stackSize];
    final int[] levelStack = new int[stackSize];

    offsetStack[stackPos] = from;
    lengthStack[stackPos] = to - from;
    levelStack[stackPos++] = 0;

    final int[] count = new int[1 << DIGIT_BITS];
    final int[] pos = new int[1 << DIGIT_BITS];
    final KEY_TYPE[] t = new KEY_TYPE[layers];

    while(stackPos > 0) {

```

```

    final int first = offsetStack[--stackPos];
    final int length = lengthStack[stackPos];
    final int level = levelStack[stackPos];
#if KEY_CLASS_Character
    final int signMask = 0;
#else
    final int signMask = level % DIGITS_PER_ELEMENT == 0 ? 1 << DIGIT_BITS - 1 : 0;
#endif

    final KEY_TYPE[] k = a[level / DIGITS_PER_ELEMENT]; // This is the key array
    final int shift = (DIGITS_PER_ELEMENT - 1 - level % DIGITS_PER_ELEMENT) * DIGIT_BITS; // This is the
    shift that extract the right byte from a key

    // Count keys.
    for(int i = first + length; i-- != first;) count[INT(KEY2LEXINT(k[i]) >>> shift & DIGIT_MASK ^ signMask)]++;

    // Compute cumulative distribution
    int lastUsed = -1;
    for (int i = 0, p = first; i < 1 << DIGIT_BITS; i++) {
        if (count[i] != 0) lastUsed = i;
        pos[i] = (p += count[i]);
    }

    final int end = first + length - count[lastUsed];
    // i moves through the start of each block
    for(int i = first, c = -1, d; i <= end; i += count[c], count[c] = 0) {
        for(int p = layers; p-- != 0;) t[p] = a[p][i];
        c = INT(KEY2LEXINT(k[i]) >>> shift & DIGIT_MASK ^ signMask);

        if (i < end) { // When all slots are OK, the last slot is necessarily OK.
            while((d = --pos[c]) > i) {
                c = INT(KEY2LEXINT(k[d]) >>> shift & DIGIT_MASK ^ signMask);
                for(int p = layers; p-- != 0;) {
                    final KEY_TYPE u = t[p];
                    t[p] = a[p][d];
                    a[p][d] = u;
                }
            }
            for(int p = layers; p-- != 0;) a[p][i] = t[p];
        }

        if (level < maxLevel && count[c] > 1) {
            if (count[c] < RADIXSORT_NO_REC) selectionSort(a, i, i + count[c], level + 1);
            else {
                offsetStack[stackPos] = i;
                lengthStack[stackPos] = count[c];
                levelStack[stackPos++] = level + 1;
            }
        }
    }

```

```
}  
}  
}  
}
```

```
#endif
```

```
#endif
```

```
/** Shuffles the specified array fragment using the specified pseudorandom number generator.
```

```
*
```

```
* @param a the array to be shuffled.
```

```
* @param from the index of the first element (inclusive) to be shuffled.
```

```
* @param to the index of the last element (exclusive) to be shuffled.
```

```
* @param random a pseudorandom number generator.
```

```
* @return { @code a }.
```

```
*/
```

```
public static KEY_GENERIC KEY_GENERIC_TYPE[] shuffle(final KEY_GENERIC_TYPE[] a, final int from,  
final int to, final Random random) {  
    for(int i = to - from; i-- != 0;) {  
        final int p = random.nextInt(i + 1);  
        final KEY_GENERIC_TYPE t = a[from + i];  
        a[from + i] = a[from + p];  
        a[from + p] = t;  
    }  
    return a;  
}
```

```
/** Shuffles the specified array using the specified pseudorandom number generator.
```

```
*
```

```
* @param a the array to be shuffled.
```

```
* @param random a pseudorandom number generator.
```

```
* @return { @code a }.
```

```
*/
```

```
public static KEY_GENERIC KEY_GENERIC_TYPE[] shuffle(final KEY_GENERIC_TYPE[] a, final Random  
random) {  
    for(int i = a.length; i-- != 0;) {  
        final int p = random.nextInt(i + 1);  
        final KEY_GENERIC_TYPE t = a[i];  
        a[i] = a[p];  
        a[p] = t;  
    }  
    return a;  
}
```

```
/** Reverses the order of the elements in the specified array.
```

```
*
```

```

* @param a the array to be reversed.
* @return { @code a}.
*/
public static KEY_GENERIC KEY_GENERIC_TYPE[] reverse(final KEY_GENERIC_TYPE[] a) {
    final int length = a.length;
    for(int i = length / 2; i-- != 0;) {
        final KEY_GENERIC_TYPE t = a[length - i - 1];
        a[length - i - 1] = a[i];
        a[i] = t;
    }
    return a;
}

/** Reverses the order of the elements in the specified array fragment.
 *
 * @param a the array to be reversed.
 * @param from the index of the first element (inclusive) to be reversed.
 * @param to the index of the last element (exclusive) to be reversed.
 * @return { @code a}.
 */
public static KEY_GENERIC KEY_GENERIC_TYPE[] reverse(final KEY_GENERIC_TYPE[] a, final int from,
final int to) {
    final int length = to - from;
    for(int i = length / 2; i-- != 0;) {
        final KEY_GENERIC_TYPE t = a[from + length - i - 1];
        a[from + length - i - 1] = a[from + i];
        a[from + i] = t;
    }
    return a;
}

/** A type-specific content-based hash strategy for arrays. */

private static final class ArrayHashStrategy KEY_GENERIC implements
Hash.Strategy<KEY_GENERIC_TYPE[]>, java.io.Serializable {
    private static final long serialVersionUID = -7046029254386353129L;

    @Override
    public int hashCode(final KEY_GENERIC_TYPE[] o) { return java.util.Arrays.hashCode(o); }

    @Override
    public boolean equals(final KEY_GENERIC_TYPE[] a, final KEY_GENERIC_TYPE[] b) { return
java.util.Arrays.equals(a, b); }
}

/** A type-specific content-based hash strategy for arrays.
 *
 * <p>This hash strategy may be used in custom hash collections whenever keys are

```

```
* arrays, and they must be considered equal by content. This strategy
* will handle { @code null } correctly, and it is serializable.
*/
```

```
#if KEYS_PRIMITIVE
public static final Hash.Strategy<KEY_TYPE[]> HASH_STRATEGY = new ArrayHashStrategy();
#else
@SuppressWarnings({"rawtypes"})
public static final Hash.Strategy HASH_STRATEGY = new ArrayHashStrategy();
#endif

}
```

Found in path(s):

```
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-
a775574/drv/Arrays.drv
```

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2003-2017 Paolo Boldi and Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

```
package PACKAGE;
```

```
#if KEY_CLASS_Object
import java.util.Comparator;
import it.unimi.dsi.fastutil.IndirectPriorityQueue;
#endif
```

```
import it.unimi.dsi.fastutil.ints.IntArrays;
```

```
import java.util.NoSuchElementException;
```

```
/** A type-specific array-based semi-indirect priority queue.
*
*/
```

```

* <p>Instances of this class use as reference list a reference array,
* which must be provided to each constructor, and represent a priority queue
* using a backing array of integer indices&mdash;all operations are performed
* directly on the array. The array is enlarged as needed, but it is never
* shrunk. Use the { @link #trim() } method to reduce its size, if necessary.
*
* <p>This implementation is extremely inefficient, but it is difficult to beat
* when the size of the queue is very small. Moreover, it allows to enqueue several
* time the same index, without limitations.
*/

```

```

public class ARRAY_INDIRECT_PRIORITY_QUEUE<KEY_GENERIC> implements
INDIRECT_PRIORITY_QUEUE<KEY_GENERIC> {

    /** The reference array. */
    protected KEY_GENERIC_TYPE refArray[];

    /** The backing array. */
    protected int array[] = IntArrays.EMPTY_ARRAY;

    /** The number of elements in this queue. */
    protected int size;

    /** The type-specific comparator used in this queue. */
    protected KEY_COMPARATOR<KEY_SUPER_GENERIC> c;

    /** The first index, cached, if { @link #firstIndexValid } is true. */
    protected int firstIndex;

    /** Whether { @link #firstIndex } contains a valid value. */
    protected boolean firstIndexValid;

    /** Creates a new empty queue without elements with a given capacity and comparator.
    *
    * @param refArray the reference array.
    * @param capacity the initial capacity of this queue.
    * @param c the comparator used in this queue, or { @code null } for the natural order.
    */
    public ARRAY_INDIRECT_PRIORITY_QUEUE(KEY_GENERIC_TYPE[] refArray, int capacity,
KEY_COMPARATOR<KEY_SUPER_GENERIC> c) {
        if (capacity > 0) this.array = new int[capacity];
        this.refArray = refArray;
        this.c = c;
    }

    /** Creates a new empty queue with given capacity and using the natural order.
    *
    */

```

```

* @param refArray the reference array.
* @param capacity the initial capacity of this queue.
*/
public ARRAY_INDIRECT_PRIORITY_QUEUE(KEY_GENERIC_TYPE[] refArray, int capacity) {
    this(refArray, capacity, null);
}

/** Creates a new empty queue with capacity equal to the length of the reference array and a given comparator.
*
* @param refArray the reference array.
* @param c the comparator used in this queue, or { @code null } for the natural order.
*/
public ARRAY_INDIRECT_PRIORITY_QUEUE(KEY_GENERIC_TYPE[] refArray, KEY_COMPARATOR
KEY_SUPER_GENERIC c) {
    this(refArray, refArray.length, c);
}

/** Creates a new empty queue with capacity equal to the length of the reference array and using the natural order.
* @param refArray the reference array.
*/
public ARRAY_INDIRECT_PRIORITY_QUEUE(KEY_GENERIC_TYPE[] refArray) {
    this(refArray, refArray.length, null);
}

/** Wraps a given array in a queue using a given comparator.
*
* <p>The queue returned by this method will be backed by the given array.
*
* @param refArray the reference array.
* @param a an array of indices into { @code refArray }.
* @param size the number of elements to be included in the queue.
* @param c the comparator used in this queue, or { @code null } for the natural order.
*/
public ARRAY_INDIRECT_PRIORITY_QUEUE(final KEY_GENERIC_TYPE[] refArray, final int[] a, int size,
final KEY_COMPARATOR KEY_SUPER_GENERIC c) {
    this(refArray, 0, c);
    this.array = a;
    this.size = size;
}

/** Wraps a given array in a queue using a given comparator.
*
* <p>The queue returned by this method will be backed by the given array.
*
* @param refArray the reference array.
* @param a an array of indices into { @code refArray }.

```

```

* @param c the comparator used in this queue, or { @code null } for the natural order.
*/
public ARRAY_INDIRECT_PRIORITY_QUEUE(final KEY_GENERIC_TYPE[] refArray, final int[] a, final
KEY_COMPARATOR KEY_SUPER_GENERIC c) {
    this(refArray, a, a.length, c);
}

/** Wraps a given array in a queue using the natural order.
*
* <p>The queue returned by this method will be backed by the given array.
*
* @param refArray the reference array.
* @param a an array of indices into { @code refArray }.
* @param size the number of elements to be included in the queue.
*/
public ARRAY_INDIRECT_PRIORITY_QUEUE(final KEY_GENERIC_TYPE[] refArray, final int[] a, int size) {
    this(refArray, a, size, null);
}

/** Wraps a given array in a queue using the natural order.
*
* <p>The queue returned by this method will be backed by the given array.
*
* @param refArray the reference array.
* @param a an array of indices into { @code refArray }.
*/
public ARRAY_INDIRECT_PRIORITY_QUEUE(final KEY_GENERIC_TYPE[] refArray, final int[] a) {
    this(refArray, a, a.length);
}

/** Returns the index (in { @link #array }) of the smallest element. */

SUPPRESS_WARNINGS_KEY_UNCHECKED
private int findFirst() {
    if (firstIndexValid) return this.firstIndex;
    firstIndexValid = true;
    int i = size;
    int firstIndex = --i;
    KEY_GENERIC_TYPE first = refArray[array[firstIndex]];

    if (c == null) while(i-- != 0) { if (KEY_LESS(refArray[array[i]], first)) first = refArray[array[firstIndex = i]]; }
    else while(i-- != 0) { if (c.compare(refArray[array[i]], first) < 0) first = refArray[array[firstIndex = i]]; }

    return this.firstIndex = firstIndex;
}

```

```

/** Returns the index (in { @link #array }) of the largest element. */

SUPPRESS_WARNINGS_KEY_UNCHECKED
private int findLast() {
    int i = size;
    int lastIndex = --i;
    KEY_GENERIC_TYPE last = refArray[array[lastIndex]];

    if (c == null) { while(i-- != 0) if (KEY_LESS(last, refArray[array[i]])) last = refArray[array[lastIndex = i]]; }
    else { while(i-- != 0) if (c.compare(last, refArray[array[i]]) < 0) last = refArray[array[lastIndex = i]]; }

    return lastIndex;
}

protected final void ensureNonEmpty() {
    if (size == 0) throw new NoSuchElementException();
}

/** Ensures that the given index is a firstIndexValid reference.
 *
 * @param index an index in the reference array.
 * @throws IndexOutOfBoundsException if the given index is negative or larger than the reference array length.
 */
protected void ensureElement(final int index) {
    if (index < 0) throw new IndexOutOfBoundsException("Index (" + index + ") is negative");
    if (index >= refArray.length) throw new IndexOutOfBoundsException("Index (" + index + ") is larger than or equal
to reference array size (" + refArray.length + ")");
}

/** {@inheritDoc}
 *
 * <p>Note that for efficiency reasons this method will <em>not</em> throw an exception
 * when { @code x } is already in the queue. However, the queue state will become
 * inconsistent and the following behaviour will not be predictable.
 */
@Override
SUPPRESS_WARNINGS_KEY_UNCHECKED
public void enqueue(int x) {
    ensureElement(x);

    if (size == array.length) array = IntArrays.grow(array, size + 1);
    if (firstIndexValid) {
        if (c == null) { if (KEY_LESS(refArray[x], refArray[array[firstIndex]])) firstIndex = size; }
        else if (c.compare(refArray[x], refArray[array[firstIndex]]) < 0) firstIndex = size;
    }
    else firstIndexValid = false;
    array[size++] = x;
}

```

```

@Override
public int dequeue() {
    ensureNonEmpty();
    final int firstIndex = findFirst();
    final int result = array[firstIndex];
    if (--size != 0) System.arraycopy(array, firstIndex + 1, array, firstIndex, size - firstIndex);
    firstIndexValid = false;
    return result;
}

```

```

@Override
public int first() {
    ensureNonEmpty();
    return array[findFirst()];
}

```

```

@Override
public int last() {
    ensureNonEmpty();
    return array[findLast()];
}

```

```

@Override
public void changed() {
    ensureNonEmpty();
    firstIndexValid = false;
}

```

```

/** {@inheritDoc}
 *
 * <p>Note that for efficiency reasons this method will <em>not</em> throw an exception
 * when {@code index} is not in the queue.
 */

```

```

@Override
public void changed(int index) {
    ensureElement(index);
    if (index == firstIndex) firstIndexValid = false;
}

```

```

/** Signals the queue that all elements have changed. */
@Override
public void allChanged() { firstIndexValid = false; }

```

```

@Override
public boolean remove(int index) {
    ensureElement(index);
    final int[] a = array;
}

```

```

int i = size;
while(i-- != 0) if (a[i] == index) break;
if (i < 0) return false;
firstIndexValid = false;
if (--size != 0) System.arraycopy(a, i + 1, a, i, size - i);
return true;
}

/** Writes in the provided array the <em>front</em> of the queue, that is, the set of indices
 * whose elements have the same priority as the top.
 * @param a an array whose initial part will be filled with the front (must be sized as least as the heap size).
 * @return the number of elements of the front.
 */
@Override
public int front(int[] a) {
    final KEY_GENERIC_TYPE top = refArray[array[findFirst()]];
    int i = size, c = 0;
    while(i-- != 0) if (KEY_EQUALS_NOT_NULL(top, refArray[array[i]])) a[c++] = array[i];
    return c;
}

@Override
public int size() { return size; }

@Override
public void clear() { size = 0; firstIndexValid = false; }

/** Trims the backing array so that it has exactly { @link #size() } elements. */
public void trim() {
    array = IntArrays.trim(array, size);
}

@Override
public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() { return c; }

@Override
public String toString() {
    StringBuffer s = new StringBuffer();
    s.append("[");
    for (int i = 0; i < size; i++) {
        if (i != 0) s.append(", ");
        s.append(refArray[array [i]]);
    }
    s.append("]");
    return s.toString();
}

```

```

#ifdef TEST
private static long seed = System.currentTimeMillis();
private static java.util.Random r = new java.util.Random(seed);

private static KEY_TYPE genKey() {
#ifdef KEY_CLASS_Byte || KEY_CLASS_Short || KEY_CLASS_Character
return (KEY_TYPE)(r.nextInt());
#elif KEYS_PRIMITIVE
return r.NEXT_KEY();
#elif KEY_CLASS_Object
return Integer.toBinaryString(r.nextInt());
#else
return new java.io.Serializable() {};
#endif
}

private static java.text.NumberFormat format = new java.text.DecimalFormat("#,###.00");
private static java.text.FieldPosition p = new java.text.FieldPosition(0);

private static String format(double d) {
StringBuffer s = new StringBuffer();
return format.format(d, s, p).toString();
}

private static void speedTest(int n, boolean comp) {
int i, j, s;
ARRAY_INDIRECT_PRIORITY_QUEUE[] m = new ARRAY_INDIRECT_PRIORITY_QUEUE[100000];
HEAP_INDIRECT_PRIORITY_QUEUE[] t = new HEAP_INDIRECT_PRIORITY_QUEUE[m.length];
KEY_TYPE k[] = new KEY_TYPE[n];
KEY_TYPE nk[] = new KEY_TYPE[m.length];
long ms;

for(i = 0; i < n; i++) k[i] = genKey();
for(i = 0; i < m.length; i++) nk[i] = genKey();

double totEnq = 0, totDeq = 0, totChange = 0, d;

for(i = 0; i < m.length; i++) {
t[i] = new HEAP_INDIRECT_PRIORITY_QUEUE(k);
m[i] = new ARRAY_INDIRECT_PRIORITY_QUEUE(k);
}

if (comp) {
for(j = 0; j < 20; j++) {

for(i = 0; i < m.length; i++) t[i].clear();

ms = System.currentTimeMillis();

```

```

s = m.length;
while(s-- != 0) { i = n; while(i-- != 0) t[s].enqueue(i); }
d = System.currentTimeMillis() - ms;
if (j > 2) totEnq += d;
System.out.print("Enqueue: " + format(m.length * n/d) + " K/s ");

ms = System.currentTimeMillis();
s = m.length;
while(s-- != 0) { i = n; while(i-- != 0) { k[t[s].first()] = nk[i]; t[s].changed(); } }
d = System.currentTimeMillis() - ms;
if (j > 2) totChange += d;
System.out.print("Change: " + format(m.length * n/d) + " K/s ");

ms = System.currentTimeMillis();
s = m.length;
while(s-- != 0) { i = n; while(i-- != 0) t[s].dequeue(); }
d = System.currentTimeMillis() - ms;
if (j > 2) totDeq += d;
System.out.print("Dequeue: " + format(m.length * n/d) + " K/s ");

System.out.println();
}

System.out.println();
System.out.println("Heap: Enqueue: " + format(m.length * (j-3)*n/totEnq) + " K/s Dequeue: " + format(m.length *
(j-3)*n/totDeq) + " K/s Change: " + format(m.length * (j-3)*n/totChange) + " K/s");

System.out.println();

totEnq = totChange = totDeq = 0;

}

for(j = 0; j < 20; j++) {

for(i = 0; i < m.length; i++) m[i].clear();

ms = System.currentTimeMillis();
s = m.length;
while(s-- != 0) { i = n; while(i-- != 0) m[s].enqueue(i); }
d = System.currentTimeMillis() - ms;
if (j > 2) totEnq += d;
System.out.print("Enqueue: " + format(m.length * n/d) + " K/s ");

ms = System.currentTimeMillis();
s = m.length;
while(s-- != 0) { i = n; while(i-- != 0) { k[m[s].first()] = nk[i]; m[s].changed(); } }
d = System.currentTimeMillis() - ms;

```

```

if (j > 2) totChange += d;
System.out.print("Change: " + format(m.length * n/d) + " K/s ");

ms = System.currentTimeMillis();
s = m.length;
while(s-- != 0) { i = n; while(i-- != 0) m[s].dequeue(); }
d = System.currentTimeMillis() - ms;
if (j > 2) totDeq += d;
System.out.print("Dequeue: " + format(m.length * n/d) + " K/s ");

System.out.println();
}

System.out.println();
System.out.println("Array: Enqueue: " + format(m.length * (j-3)*n/totEng) + " K/s Dequeue: " + format(m.length *
(j-3)*n/totDeq) + " K/s Change: " + format(m.length * (j-3)*n/totChange) + " K/s");

System.out.println();
}

private static void fatal(String msg) {
System.out.println(msg);
System.exit(1);
}

private static void ensure(boolean cond, String msg) {
if (cond) return;
fatal(msg);
}

private static boolean heapEqual(int[] a, int[] b, int sizea, int sizeb) {
if (sizea != sizeb) return false;
KEY_TYPE[] aa = new KEY_TYPE[sizea];
KEY_TYPE[] bb = new KEY_TYPE[sizea];
for(int i = 0; i < sizea; i++) {
aa[i] = ref[a[i]];
bb[i] = ref[b[i]];
}
java.util.Arrays.sort(aa);
java.util.Arrays.sort(bb);
while(sizea-- != 0) if (!KEY_EQUALS(aa[sizea], bb[sizea])) return false;
return true;
}

private static KEY_TYPE[] ref;

```

```

protected static void runTest(int n) {
    long ms;
    Exception mThrowsIllegal, tThrowsIllegal, mThrowsOutOfBounds, tThrowsOutOfBounds, mThrowsNoElement,
    tThrowsNoElement;
    int rm = 0, rt = 0;

    ref = new KEY_TYPE[n];

    for(int i = 0; i < n; i++) ref[i] = genKey();

    ARRAY_INDIRECT_PRIORITY_QUEUE m = new ARRAY_INDIRECT_PRIORITY_QUEUE(ref);
    HEAP_INDIRECT_PRIORITY_QUEUE t = new HEAP_INDIRECT_PRIORITY_QUEUE(ref);

    /* We add pairs to t. */
    for(int i = 0; i < n / 2; i++) {
        t.enqueue(i);
        m.enqueue(i);
    }

    ensure(heapEqual(m.array, t.heap, m.size(), t.size()), "Error (" + seed + "): m and t differ after creation (" + m + ", "
    + t + ")");

    /* Now we add and remove random data in m and t, checking that the result is the same. */

    for(int i=0; i<2*n; i++) {
        if (r.nextDouble() < 0.01) {
            t.clear();
            m.clear();
            for(int j = 0; j < n / 2; j++) {
                t.enqueue(j);
                m.enqueue(j);
            }
        }
    }

    int T = r.nextInt(2 * n);

    mThrowsNoElement = tThrowsNoElement = mThrowsOutOfBounds = tThrowsOutOfBounds = mThrowsIllegal =
    tThrowsIllegal = null;

    try {
        t.enqueue(T);
    }
    catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }
    catch (IllegalArgumentException e) { tThrowsIllegal = e; }

    if (tThrowsIllegal == null) { // To skip duplicates
        try {
            m.enqueue(T);

```

```

    }
    catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }
    catch (IllegalArgumentException e) { mThrowsIllegal = e; }
    }

    mThrowsIllegal = tThrowsIllegal = null; // To skip duplicates

    ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + seed + "): enqueue()
    divergence in IndexOutOfBoundsException for " + T + " (" + mThrowsOutOfBounds + ", " + tThrowsOutOfBounds
    + ")");
    ensure((mThrowsIllegal == null) == (tThrowsIllegal == null), "Error (" + seed + "): enqueue() divergence in
    IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ")");

    ensure(heapEqual(m.array, t.heap, m.size(), t.size()), "Error (" + seed + "): m and t differ after enqueue (" + m + ", "
    + t + ")");

    if (m.size() != 0) {
        ensure(KEY_EQUALS(ref[m.first()], ref[t.first()]), "Error (" + seed + "): m and t differ in first element after
        enqueue (" + m.first() + "->" + ref[m.first()] + ", " + t.first() + "->" + ref[t.first()] + ")");
    }

    mThrowsNoElement = tThrowsNoElement = mThrowsOutOfBounds = tThrowsOutOfBounds = mThrowsIllegal =
    tThrowsIllegal = null;

    try {
        rm = m.dequeue();
        while(! m.isEmpty() && KEY_EQUALS(ref[m.first()], ref[rm])) m.dequeue();
    }
    catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }
    catch (IllegalArgumentException e) { mThrowsIllegal = e; }
    catch (java.util.NoSuchElementException e) { mThrowsNoElement = e; }

    try {
        rt = t.dequeue();
        while(! t.isEmpty() && KEY_EQUALS(ref[t.first()], ref[rt])) t.dequeue();
    }
    catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }
    catch (IllegalArgumentException e) { tThrowsIllegal = e; }
    catch (java.util.NoSuchElementException e) { tThrowsNoElement = e; }

    ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + seed + "): dequeue()
    divergence in IndexOutOfBoundsException (" + mThrowsOutOfBounds + ", " + tThrowsOutOfBounds + ")");
    ensure((mThrowsIllegal == null) == (tThrowsIllegal == null), "Error (" + seed + "): dequeue() divergence in
    IllegalArgumentException (" + mThrowsIllegal + ", " + tThrowsIllegal + ")");
    ensure((mThrowsNoElement == null) == (tThrowsNoElement == null), "Error (" + seed + "): dequeue() divergence
    in java.util.NoSuchElementException (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");
    if (mThrowsOutOfBounds == null) ensure(KEY_EQUALS(ref[rt], ref[rm]), "Error (" + seed + "): divergence in
    dequeue() between m and t (" + rm + "->" + ref[rm] + ", " + rt + "->" + ref[rt] + ")");

```

```
ensure(heapEqual(m.array, t.heap, m.size(), t.size()), "Error (" + seed + "): m and t differ after dequeue (" + m + ", " + t + ")");
```

```
if (m.size() != 0) {  
    ensure(KEY_EQUALS(ref[m.first()], ref[t.first()]), "Error (" + seed + "): m and t differ in first element after dequeue (" + m.first() + "->" + ref[m.first()] + ", " + t.first() + "->" + ref[t.first()] + ")");  
}
```

```
mThrowsNoElement = tThrowsNoElement = mThrowsOutOfBounds = tThrowsOutOfBounds = mThrowsIllegal =  
tThrowsIllegal = null;
```

```
int pos = r.nextInt(n * 2);
```

```
try {  
    m.remove(pos);  
}  
catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }  
catch (IllegalArgumentException e) { mThrowsIllegal = e; }  
catch (java.util.NoSuchElementException e) { mThrowsNoElement = e; }
```

```
try {  
    t.remove(pos);  
}  
catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }  
catch (IllegalArgumentException e) { tThrowsIllegal = e; }  
catch (java.util.NoSuchElementException e) { tThrowsNoElement = e; }
```

```
ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + seed + "): remove(int)  
divergence in IndexOutOfBoundsException (" + mThrowsOutOfBounds + ", " + tThrowsOutOfBounds + ")");  
ensure((mThrowsIllegal == null) == (tThrowsIllegal == null), "Error (" + seed + "): remove(int) divergence in  
IllegalArgumentException (" + mThrowsIllegal + ", " + tThrowsIllegal + ")");  
ensure((mThrowsNoElement == null) == (tThrowsNoElement == null), "Error (" + seed + "): remove(int)  
divergence in java.util.NoSuchElementException (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");
```

```
ensure(heapEqual(m.array, t.heap, m.size(), t.size()), "Error (" + seed + "): m and t differ after remove(int) (" + m +  
", " + t + ")");
```

```
if (m.size() != 0) {  
    ensure(KEY_EQUALS(ref[m.first()], ref[t.first()]), "Error (" + seed + "): m and t differ in first element after  
remove(int) (" + m.first() + "->" + ref[m.first()] + ", " + t.first() + "->" + ref[t.first()] + ")");  
}
```

```
mThrowsNoElement = tThrowsNoElement = mThrowsOutOfBounds = tThrowsOutOfBounds = mThrowsIllegal =  
tThrowsIllegal = null;
```

```

pos = r.nextInt(n);

try {
    t.changed(pos);
}
catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }
catch (IllegalArgumentException e) { tThrowsIllegal = e; }
catch (java.util.NoSuchElementException e) { tThrowsNoElement = e; }

if (tThrowsIllegal == null) {
    try {
        m.changed(pos);
    }
    catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }
    catch (IllegalArgumentException e) { mThrowsIllegal = e; }
    catch (java.util.NoSuchElementException e) { mThrowsNoElement = e; }
}

ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + seed + "): change(int)
divergence in IndexOutOfBoundsException (" + mThrowsOutOfBounds + ", " + tThrowsOutOfBounds + ")");
//ensure((mThrowsIllegal == null) == (tThrowsIllegal == null), "Error (" + seed + "): change(int) divergence in
IllegalArgumentException (" + mThrowsIllegal + ", " + tThrowsIllegal + ")");
ensure((mThrowsNoElement == null) == (tThrowsNoElement == null), "Error (" + seed + "): change(int)
divergence in java.util.NoSuchElementException (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");

ensure(heapEqual(m.array, t.heap, m.size(), t.size()), "Error (" + seed + "): m and t differ after change(int) (" + m +
", " + t + ")");

if (m.size() != 0) {
    ensure(KEY_EQUALS(ref[m.first()], ref[t.first()]), "Error (" + seed + "): m and t differ in first element after
change(int) (" + m.first() + "->" + ref[m.first()] + ", " + t.first() + "->" + ref[t.first()] + ")");
}

int[] temp = (int[])t.heap.clone();
java.util.Arrays.sort(temp, 0, t.size()); // To scramble a bit
m = new ARRAY_INDIRECT_PRIORITY_QUEUE(m.refArray, temp, t.size());

ensure(heapEqual(m.array, t.heap, m.size(), t.size()), "Error (" + seed + "): m and t differ after wrap (" + m + ", " + t
+ ")");

if (m.size() != 0) {
    ensure(KEY_EQUALS(ref[m.first()], ref[t.first()]), "Error (" + seed + "): m and t differ in first element after wrap ("
+ m.first() + "->" + ref[m.first()] + ", " + t.first() + "->" + ref[t.first()] + ")");
}

if (m.size() != 0 && ((new it.unimi.dsi.fastutil.ints.IntOpenHashSet(m.array, 0, m.size())).size() == m.size())) {

```

```

int first = m.first();
ref[first] = genKey();

//System.err.println("Pre-change m: " +m);
//System.err.println("Pre-change t: " +t);
m.changed();
t.changed(first);

//System.err.println("Post-change m: " +m);
//System.err.println("Post-change t: " +t);

ensure(heapEqual(m.array, t.heap, m.size(), t.size()), "Error (" + seed + "): m and t differ after change (" + m + ", " +
t + ")");

if (m.size() != 0) {
    ensure(KEY_EQUALS(ref[m.first()], ref[t.first()]), "Error (" + seed + "): m and t differ in first element after change
(" + m.first() + "->" + ref[m.first()] + ", " + t.first() + "->" + ref[t.first()] + ")");
}
}
}

/* Now we check that m actually holds the same data. */

m.clear();
ensure(m.isEmpty(), "Error (" + seed + "): m is not empty after clear()");

System.out.println("Test OK");
}

public static void main(String args[]) {
int n = Integer.parseInt(args[1]);
if (args.length > 2) r = new java.util.Random(seed = Long.parseLong(args[2]));

try {
if ("speedTest".equals(args[0]) || "speedComp".equals(args[0])) speedTest(n, "speedComp".equals(args[0]));
else if ("test".equals(args[0])) runTest(n);
} catch(Throwable e) {
e.printStackTrace(System.err);
System.err.println("seed: " + seed);
}
}

#endif
}

```

Found in path(s):

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/drv/ArrayIndirectPriorityQueue.drv

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2010-2017 Sebastiano Vigna

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*

* For the sorting code:

*

* Copyright (C) 1999 CERN - European Organization for Nuclear Research.

*

* Permission to use, copy, modify, distribute and sell this software and

* its documentation for any purpose is hereby granted without fee,

* provided that the above copyright notice appear in all copies and that

* both that copyright notice and this permission notice appear in

* supporting documentation. CERN makes no representations about the

* suitability of this software for any purpose. It is provided "as is"

* without expressed or implied warranty.

*/

Found in path(s):

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/src/it/unimi/dsi/fastutil/BigArrays.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2002-2017 Sebastiano Vigna

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

- * Unless required by applicable law or agreed to in writing, software
- * distributed under the License is distributed on an "AS IS" BASIS,
- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.

*/

```
package PACKAGE;
```

```
import VALUE_PACKAGE.VALUE_COLLECTION;  
import VALUE_PACKAGE.VALUE_ABSTRACT_COLLECTION;  
import VALUE_PACKAGE.VALUE_ITERATOR;
```

```
#if KEYS_PRIMITIVE && VALUES_PRIMITIVE  
import it.unimi.dsi.fastutil.objects.ObjectIterator;  
#endif  
import it.unimi.dsi.fastutil.objects.AbstractObjectSet;
```

```
import java.util.Iterator;  
import java.util.Map;
```

```
/** An abstract class providing basic methods for maps implementing a type-specific interface.  
 *  
 * <p>Optional operations just throw an { @link  
 * UnsupportedOperationException}. Generic versions of accessors delegate to  
 * the corresponding type-specific counterparts following the interface rules  
 * (they take care of returning { @code null} on a missing key).  
 *  
 * <p>As a further help, this class provides a { @link BasicEntry BasicEntry} inner class  
 * that implements a type-specific version of { @link java.util.Map.Entry}; it  
 * is particularly useful for those classes that do not implement their own  
 * entries (e.g., most immutable maps).  
 */
```

```
public abstract class ABSTRACT_MAP KEY_VALUE_GENERIC extends ABSTRACT_FUNCTION  
KEY_VALUE_GENERIC implements MAP KEY_VALUE_GENERIC, java.io.Serializable {
```

```
private static final long serialVersionUID = -4940583368468432370L;
```

```
protected ABSTRACT_MAP() {}
```

```
@Override  
public boolean containsValue(final VALUE_TYPE v) {  
    return values().contains(v);  
}
```

```
@Override
```

```

public boolean containsKey(final KEY_TYPE k) {
    final ObjectIterator<MAP.Entry KEY_VALUE_GENERIC> i = ENTRYSET().iterator();
    while(i.hasNext())
        if (i.next().ENTRY_GET_KEY() == k)
            return true;

    return false;
}

@Override
public boolean isEmpty() {
    return size() == 0;
}

/** This class provides a basic but complete type-specific entry class for all those maps implementations
 * that do not have entries on their own (e.g., most immutable maps).
 *
 * <p>This class does not implement { @link java.util.Map.Entry#setValue(Object) setValue()}, as the modification
 * would not be reflected in the base map.
 */

public static class BasicEntry KEY_VALUE_GENERIC implements MAP.Entry KEY_VALUE_GENERIC {
    protected KEY_GENERIC_TYPE key;
    protected VALUE_GENERIC_TYPE value;

    public BasicEntry() {}

    public BasicEntry(final KEY_GENERIC_CLASS key, final VALUE_GENERIC_CLASS value) {
        this.key = KEY_CLASS2TYPE(key);
        this.value = VALUE_CLASS2TYPE(value);
    }

    #if KEYS_PRIMITIVE || VALUES_PRIMITIVE

    public BasicEntry(final KEY_GENERIC_TYPE key, final VALUE_GENERIC_TYPE value) {
        this.key = key;
        this.value = value;
    }

    #endif

    @Override
    public KEY_GENERIC_TYPE ENTRY_GET_KEY() {
        return key;
    }

    @Override
    public VALUE_GENERIC_TYPE ENTRY_GET_VALUE() {

```

```

    return value;
}

@Override
public VALUE_GENERIC_TYPE setValue(final VALUE_GENERIC_TYPE value) {
    throw new UnsupportedOperationException();
}

SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
@Override
public boolean equals(final Object o) {
    if (!(o instanceof Map.Entry)) return false;
    if (o instanceof MAP.Entry) {
        final MAP.Entry KEY_VALUE_GENERIC e = (MAP.Entry KEY_VALUE_GENERIC) o;
        return KEY_EQUALS(key, e.ENTRY_GET_KEY()) && VALUE_EQUALS(value, e.ENTRY_GET_VALUE());
    }

    final Map.Entry<?,?> e = (Map.Entry<?,?>)o;
    final Object key = e.getKey();
    #if KEYS_PRIMITIVE
        if (key == null || !(key instanceof KEY_CLASS)) return false;
    #endif
    final Object value = e.getValue();
    #if VALUES_PRIMITIVE
        if (value == null || !(value instanceof VALUE_CLASS)) return false;
    #endif
    return KEY_EQUALS(this.key, KEY_OBJ2TYPE(key)) && VALUE_EQUALS(this.value,
VALUE_OBJ2TYPE(value));
}

@Override
public int hashCode() {
    return KEY2JAVAHASH(key) ^ VALUE2JAVAHASH(value);
}

@Override
public String toString() {
    return key + "->" + value;
}
}

/** This class provides a basic implementation for an Entry set which forwards some queries to the map.
*/

public abstract class BasicEntrySet KEY_VALUE_GENERIC extends AbstractObjectSet<Entry
KEY_VALUE_GENERIC> {
    protected final MAP KEY_VALUE_GENERIC map;

```

```

public BasicEntrySet(final MAP KEY_VALUE_GENERIC map) {
    this.map = map;
}

SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
@Override
public boolean contains(final Object o) {
    if (!(o instanceof Map.Entry)) return false;

    if (o instanceof MAP.Entry) {
        final MAP.Entry KEY_VALUE_GENERIC e = (MAP.Entry KEY_VALUE_GENERIC) o;
        final KEY_GENERIC_TYPE k = e.ENTRY_GET_KEY();
        return map.containsKey(k) && VALUE_EQUALS(map.GET_VALUE(k), e.ENTRY_GET_VALUE());
    }

    final Map.Entry<?, ?> e = (Map.Entry<?, ?>) o;

    #if KEYS_PRIMITIVE
        final Object key = e.getKey();
        if (key == null || !(key instanceof KEY_GENERIC_CLASS)) return false;
        final KEY_TYPE k = KEY_OBJ2TYPE(key);
    #else
        final Object k = e.getKey();
    #endif
    #if VALUES_PRIMITIVE
        final Object value = e.getValue();
        if (value == null || !(value instanceof VALUE_GENERIC_CLASS)) return false;
    #endif

    return map.containsKey(k) && VALUE_EQUALS(map.GET_VALUE(k), VALUE_OBJ2TYPE(value));
}

SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
@Override
public boolean remove(final Object o) {
    if (!(o instanceof Map.Entry)) return false;

    if (o instanceof MAP.Entry) {
        final MAP.Entry KEY_VALUE_GENERIC e = (MAP.Entry KEY_VALUE_GENERIC) o;
        return map.remove(e.ENTRY_GET_KEY(), e.ENTRY_GET_VALUE());
    }

    Map.Entry<?, ?> e = (Map.Entry<?, ?>) o;

    #if KEYS_PRIMITIVE
        final Object key = e.getKey();
        if (key == null || !(key instanceof KEY_GENERIC_CLASS)) return false;
        final KEY_TYPE k = KEY_OBJ2TYPE(key);

```

```

#else
    final Object k = e.getKey();
#endif
#if VALUES_PRIMITIVE
    final Object value = e.getValue();
    if (value == null || !(value instanceof VALUE_GENERIC_CLASS)) return false;
    final VALUE_TYPE v = VALUE_OBJ2TYPE(value);
#else
    final Object v = e.getValue();
#endif

    return map.remove(k, v);
}

@Override
public int size() {
    return map.size();
}
}

/** Returns a type-specific-set view of the keys of this map.
 *
 * <p>The view is backed by the set returned by { @link Map#entrySet()}. Note that
 * <em>no attempt is made at caching the result of this method</em>, as this would
 * require adding some attributes that lightweight implementations would
 * not need. Subclasses may easily override this policy by calling
 * this method and caching the result, but implementors are encouraged to
 * write more efficient ad-hoc implementations.
 *
 * @return a set view of the keys of this map; it may be safely cast to a type-specific interface.
 */
@Override
public SET KEY_GENERIC keySet() {
    return new ABSTRACT_SET KEY_GENERIC() {
        @Override
        public boolean contains(final KEY_TYPE k) { return containsKey(k); }
        @Override
        public int size() { return ABSTRACT_MAP.this.size(); }
        @Override
        public void clear() { ABSTRACT_MAP.this.clear(); }
        @Override
        public KEY_ITERATOR KEY_GENERIC iterator() {
            return new KEY_ITERATOR KEY_GENERIC() {
                private final ObjectIterator<MAP.Entry KEY_VALUE_GENERIC> i =
MAPS.fastIterator(ABSTRACT_MAP.this);
                @Override
                public KEY_GENERIC_TYPE NEXT_KEY() { return i.next().ENTRY_GET_KEY(); };
            };
        }
    };
}

```

```

    @Override
    public boolean hasNext() { return i.hasNext(); }
    @Override
    public void remove() { i.remove(); }
};
}
};
}

/** Returns a type-specific-set view of the values of this map.
 *
 * <p>The view is backed by the set returned by { @link Map#entrySet()}. Note that
 * <em>no attempt is made at caching the result of this method</em>, as this would
 * require adding some attributes that lightweight implementations would
 * not need. Subclasses may easily override this policy by calling
 * this method and caching the result, but implementors are encouraged to
 * write more efficient ad-hoc implementations.
 *
 * @return a set view of the values of this map; it may be safely cast to a type-specific interface.
 */
@Override
public VALUE_COLLECTION VALUE_GENERIC values() {
    return new VALUE_ABSTRACT_COLLECTION VALUE_GENERIC() {
        @Override
        public boolean contains(final VALUE_TYPE k) { return containsValue(k); }
        @Override
        public int size() { return ABSTRACT_MAP.this.size(); }
        @Override
        public void clear() { ABSTRACT_MAP.this.clear(); }

        @Override
        public VALUE_ITERATOR VALUE_GENERIC iterator() {
            return new VALUE_ITERATOR VALUE_GENERIC() {
                private final ObjectIterator<MAP.Entry KEY_VALUE_GENERIC> i =
MAPS.fastIterator(ABSTRACT_MAP.this);
                @Override
                public VALUE_GENERIC_TYPE NEXT_VALUE() { return i.next().ENTRY_GET_VALUE(); }
                @Override
                public boolean hasNext() { return i.hasNext(); }
            };
        }
    };
}

/** { @inheritDoc} */
@SuppressWarnings({"unchecked","deprecation"})
@Override
public void putAll(final Map<? extends KEY_GENERIC_CLASS,? extends VALUE_GENERIC_CLASS> m) {

```

```

if (m instanceof MAP) {
    ObjectIterator<MAP.Entry KEY_VALUE_GENERIC> i = MAPS.fastIterator((MAP KEY_VALUE_GENERIC)
m);

    while (i.hasNext()) {
        final MAP.Entry KEY_VALUE_EXTENDS_GENERIC e = i.next();
        put(e.ENTRY_GET_KEY(), e.ENTRY_GET_VALUE());
    }
} else {
    int n = m.size();
    final Iterator<? extends Map.Entry<? extends KEY_GENERIC_CLASS,? extends VALUE_GENERIC_CLASS>> i
= m.entrySet().iterator();
    Map.Entry<? extends KEY_GENERIC_CLASS,? extends VALUE_GENERIC_CLASS> e;
    while (n-- != 0) {
        e = i.next();
        put(e.getKey(), e.getValue());
    }
}
}

/** Returns a hash code for this map.
 *
 * The hash code of a map is computed by summing the hash codes of its entries.
 *
 * @return a hash code for this map.
 */
@Override
public int hashCode() {
    int h = 0, n = size();
    final ObjectIterator<MAP.Entry KEY_VALUE_GENERIC> i = MAPS.fastIterator(this);

    while(n-- != 0) h += i.next().hashCode();
    return h;
}

@Override
public boolean equals(Object o) {
    if (o == this) return true;
    if (!(o instanceof Map)) return false;

    final Map<?,?> m = (Map<?,?>)o;
    if (m.size() != size()) return false;
    return ENTRYSET().containsAll(m.entrySet());
}

@Override
public String toString() {
    final StringBuilder s = new StringBuilder();

```

```

final ObjectIterator<MAP.Entry KEY_VALUE_GENERIC> i = MAPS.fastIterator(this);
int n = size();
MAP.Entry KEY_VALUE_GENERIC e;
boolean first = true;

s.append("{}");

while(n-- != 0) {
    if (first) first = false;
    else s.append(", ");

    e = i.next();

    #if KEYS_REFERENCE
        if (this == e.getKey()) s.append("(this map)"); else
    #endif
        s.append(String.valueOf(e.ENTRY_GET_KEY()));
    s.append("=>");
    #if VALUES_REFERENCE
        if (this == e.getValue()) s.append("(this map)"); else
    #endif
        s.append(String.valueOf(e.ENTRY_GET_VALUE()));
    }

    s.append("}");
    return s.toString();
}
}

```

Found in path(s):

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/drv/AbstractMap.drv

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2002-2017 Sebastiano Vigna

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

```
*/
```

```
package PACKAGE;
```

```
import it.unimi.dsi.fastutil.Hash;
```

```
/** A type-specific {@link Hash} interface.
```

```
*
```

```
* @see Hash
```

```
*/
```

```
public interface HASH {
```

```
/** A type-specific hash strategy.
```

```
*
```

```
* @see it.unimi.dsi.fastutil.Hash.Strategy
```

```
*/
```

```
interface Strategy {
```

```
/** Returns the hash code of the specified element with respect to this hash strategy.
```

```
*
```

```
* @param e an element.
```

```
* @return the hash code of the given element with respect to this hash strategy.
```

```
*/
```

```
int hashCode(KEY_TYPE e);
```

```
/** Returns true if the given elements are equal with respect to this hash strategy.
```

```
*
```

```
* @param a an element.
```

```
* @param b another element.
```

```
* @return true if the two specified elements are equal with respect to this hash strategy.
```

```
*/
```

```
boolean equals(KEY_TYPE a, KEY_TYPE b);
```

```
}
```

```
}
```

Found in path(s):

```
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-  
a775574/drv/Hash.drv
```

No license file was found, but licenses were detected in source scan.

8.2.3

- Array-based lists were not allocating a backing array even if the

required capacity was greater than the default capacity, violating the contract. Thanks to for reporting this bug.

- FastMultiByteArrayInputStream had since 2014 the wrong slice size (1Ki instead of 1Gi). Thanks to Thibault Allançon for his detailed reports, which dug out this old one.

8.2.2

- Fixed small bug in new lazy allocation scheme for array-based lists. Resizing would throw an exception in certain circumstances.

- Now strategies must accept a supertype of the base type. Thanks to shevek@github.com for suggesting this change.

- The Maven source jar does not contain tests anymore.

8.2.1

- Added default modularization to OSGi version, too.

8.2.0

- Added default modularization. Thanks to Joshua Popoff for taking care of this issue.

- Implemented lazy allocation of arrays in array-based lists. Thanks to Zheka Kozlov for suggesting this feature.

- Fixed a long-standing issue: the allocation of big lists based on big arrays could not go beyond 2^{31} because of a cut-and-paste bug.

- In line with the JDK, the default initial capacity for lists is 10 and backing arrays (and, in generally, arrays) are automatically enlarged by a 50% factor instead of a 100% factor.

- Included dependency-finding scripts by Tobias Meggendorfer.

8.1.1

- More default methods.

- Fixed lack of proper `forEach()/fastForEach()` method for entry sets of linked maps (the order of iteration would be random). Thanks to Søren Gjesse for reporting this bug.

- Fixed lack of proper licensing in test sources.

- A new NO_SMALL_TYPES makefile variable makes it possible to compile a version of fastutil for ints, longs and doubles only. Thanks to Tobias Meggendorfer for implementing this feature.

8.1.0

- WARNING: backward-incompatible name change for a few "compute" methods.

- Implemented efficient new map methods and new iterable methods in hash-based maps.

- A number of minor glitches has been fixed.

- FastIterable now has a fastForEach() method.

- Fixed ancient bug in the remove() method of entry sets of tree-based maps.

- Array-based maps have a working key set and value collection.

8.0.0

- First release for Java 8 only, with implementation of new default methods for iterators and maps. Thanks to Tobias Meggendorfer and Salomon Sickert from the Technical University of Munich for help in the implementation. Structure-specific, more efficient code will be added in the near future. All wrapper (synchronized, etc.) have been updated.

- Abstract type-specific comparators are deprecated: their only abstract method has been pulled up as a default method of the type-specific interface, which makes it possible to use lambda expressions to define type-specific comparators.

- The default return value setter/getter are now optional for functions. In this way, functions can now be defined using lambda expressions (and they implement java.util.function.Function).

- New static methods in type-specific Maps and SortedMaps make it possible to use easily fast iterators, even within for loops. Thanks to Tobias Meggendorfer and Salomon Sickert from the Technical University of Munich for implementing this feature.

- All hash-based structure now keep track of the size of their backing array at creation time, and will never rehash below that size. Thanks to Patrick Julien for suggesting this feature.

- Every deprecated method marked to be removed, or replaced by another method, has been eliminated.
- Many abstract classes are now obsolete as their abstract methods have become default methods.
- The type hierarchy of big-list iterators has been fixed.

7.2.0

- Major code cleanup. Several unnecessary method implementations have been deleted. Several missing methods (e.g., equals()/hashCode() in wrappers) have been implemented. All methods should now have a reasonable Javadoc comment.
- Clarified the rem()/remove() conundrum: type-specific collections should use and override rem(), but type-specific sets should use and override remove().
- Fixed circular implementation of containsKey() in type-specific abstract collections.
- Pulled up deprecation at the highest possible (interface) level.
- Collection.*All() methods do not assume anymore that they can rely on the argument size().

7.1.1

- Fixed decade-old efficiency bug: implementation of RandomAccess was hidden from synchronized and unmodifiable wrappers. Thanks to Peter Burka for reporting this bug.
- The defaultReturnValue() getter in unmodifiable maps was throwing an UnsupportedOperationException instead of delegating the call to the underlying map. Thanks to Alex Fiennes for reporting this bug.

7.1.0

- Fixed decade-old efficiency bug. Due to a name clash between lists and sets, the type-specific deletion method of a type-specific collection is rem(), not remove(). The latter is reinstated in sets (but not, for example, in lists) by the type-specific abstract set classes. Nonetheless, implementors of subclasses must override rem(), not remove(), as methods such as the type-specific version of removeAll() invoke necessarily invoke rem() rather than remove(). Up to this version, all concrete set implementations were overriding remove(), instead, causing inefficiencies in the inherited methods. Thanks

to Christian Habermehl for reporting this bug.

- Fixed a bug introduced with the removal of old-style gcc assertions: all load methods in BinIO that did not specify the number of elements to read were computing the number of items in the loaded file incorrectly, causing an EOFException (except for booleans and bytes).

7.0.13

- Fixed inheritance problem that would surface as key sets of maps not implementing remove(). Thanks to Luke Nezda for reporting this bug.

7.0.12

- Collection.isEmpty() was checking for iterator().hasNext() instead of the opposite. Thanks to Olaf Krische for reporting this bug.
- Fixed lack of test for null/wrong class when testing entries.

7.0.11

- Several small glitches that were making fastutil's classes behave differently from those of java.util have been fixed. Thanks to Balázs Attila-Mihály for reporting these bug (obtained by massive testing using Guava's battery of unit tests).

7.0.10

- The infinite-loop bug was affecting trim(int) besides trim(). Thanks to Igor Kabiljo for reporting this bug.
- With the help of Erich Schubert, all methods with a type-specific, more efficient counterpart have been deprecated.

7.0.9

- A subtle infinite-loop bug in hash-based structures (happening with load factor 1 and tiny structures) has been fixed. Thanks to Tuomas Välimäki and Jarkko Mönkkönen for reporting this bug.
- Now tree-based map have an addTo() method analogous to that of hash-based maps. Thanks to Almog Gavra for implementing the method.

7.0.8

- Non-indirect priority queues are now serializable.

- Fixed implementation of structures based on a custom hash: keys strategy-equal to zero would not be managed correctly. Thanks to Shawn Cao for reporting this bug.

- Natural/opposite/abstract comparators are now serializable.

7.0.7

- Now we check whether ranges of parallel sorting algorithms are too small *before* creating the thread pool.

- Merged Erich Schubert's fix for `Object{AVL,RB}TreeSet.get()`.

7.0.6

- Faster priority queues: better variable caching, deleted a spurious check, tests for parameters turned into assertions.

- New collection-based constructors for heap-based priority queues.

- Reviewed `ObjectArrays.newArray()` so that there is a fast track for reallocation of arrays of type `Object[]`.

7.0.4

- Fixed old-standing bug: iterators in linked maps would return bogus data on `entrySet().next()/entrySet().previous()` when no element is available instead of throwing an exception.

7.0.3

- Fixed wrong generation of custom-hash classes with primitive keys. Thanks to Michael Henke for reporting this bug.

7.0.2

- Now we `shutdown()` correctly `ForkJoinPool`'s.

- Constants limiting parallelism and recursion have been tuned.

- New implementations of indirect [parallel] quicksort (in ascending order only).

- New stabilization method for post-processing of non-stable indirect sorts.

7.0.1

- Now generated sources are formatted using the Eclipse command-line facility.

7.0.0

- Now we need Java 7.

- New parallel versions of radix sort and quicksort. The sequential implementations have been further improved.

- Restored the previous constants in mixing functions.

6.6.4

- Hopefully better mixing functions created by a genetic algorithm.

- Fixed a bug in floating-point hash-based containers: -0.0 and +0.0 were both converted to +0.0. Thanks to Dawid Weiss for reporting this bug.

6.6.3

- Fixed subtle wrap-around bug in removal from iterator. Thanks to Eugene Yakavets for reporting this bug.

6.6.2

- We now reduce backing arrays of hash-based classes when they are filled below one fourth of the load factor. The reduction is not performed when deleting from an iterator, as it would make iteration impossible.

- Significant simplification of `Iterator.remove()`'s implementations for hash-based data structures.

6.6.1

- Fixed missed implementation: `setValue()` was not implemented for fast iterators in hash-based maps.

6.6.0

- Major (transparent) rewrite of all hash-based classes inspired by the Goldman-Sachs collections. We no longer allocate a byte array to store the status of each slot: a null (or zero) key denotes an empty slot. The null key is handled separately. The reduction in memory accesses makes the cost of the additional logic negligible, and brings in significant performance improvements. The code is actually simplified, as all loops become a search for a nonzero element.

- Partial (one-step) unrolling of all lookup loops, following the strategy used in Koloboke.
- Fixed an old bug: `entrySet().remove(Entry)` would remove entries checking the value of the key, only.
- Fixed a bug in the iterator over hash big sets.
- OSGI metadata, thanks to Benson Margulies.

6.5.17

- Now `TextIO` methods trim strings before parsing numbers. This avoids obnoxious exceptions when numbers are followed by whitespace.

6.5.16

- Improved speed of `FastMultiByteArrayInputStream`, and removed support for `mark()/reset()`.
- Deprecated array `fill()` methods in favour of `java.util`'s.

6.5.15

- De-deprecated quicksort methods for primitive-type arrays. It turned out that Java's `Arrays.sort()` switches to mergesort on large, semi-sorted arrays. Moreover, in Java 7 the support array is allocated of the same size of the argument array, not of the sorted fragment. This performance bug was entirely killing the performance of `Transform.transposeOffline()` and other methods. Until that bug is fixed, we will have to rely on our quicksort method (which is a pity, because Java's sort is, for the rest, so beautifully engineered).

6.5.14

- Equality in type-specific hash-based data structures with float or double keys is now checked by converting to int/long bits using the conversion method of the appropriate class. Previously, using NaNs as keys would have led to misbehaviour. Thanks to Davide Savazzi for reporting this bug.

6.5.13

- Fixed a very unlikely corner case that might have led to reduction in size of an array instead of a growth. Thanks to Ernst Reissner for reporting this bug.

- InspectableFileCachedInputStream no longer performs a call to RandomAccessFile.position() when the end of file has been reached and the file is entirely held in memory.
- All front-coded lists now implement java.util.RandomAccess.

6.5.12

- Removed some useless wrapper creation in a few methods of tree-based map classes.
- Fixed pathological maxFill computation for very small-sized big open hash sets.

6.5.11

- A very old and subtle performance bug in hash-based data structures has been fixed. Backing arrays were allocated using the number of expected elements divided by the load factor. However, since the test for rehashing was fired by equality with the table size multiplied by the load factor, if the expected number of elements multiplied by the load factor was an integer a useless rehash would happen for the very last added element. The only effect was an useless increase in object creation.

6.5.10

- Now iterators in object set constructors are of type Iterator, and not anymore ObjectIterator. The kind of allowed iterators has been rationalised and made uniform through all classes implementing Set.

6.5.9

- New methods to get a type-specific Iterable from binary or text files.

6.5.8

- Fixed stupid bug in creation of array-based FIFO queues.

6.5.7

- Fixed a very subtle bug in hash-based data structures: addAll() to a newly created structure could require a very long time due to correlation between the positions in structures with different table sizes.

6.5.6

- equals() method between arrays have been deprecated in favour of the java.util.Arrays version, which is intrinsified in recent JVMs.

- InspectableFileCachedInputStream.reopen() makes it possible to read again from the start an instance on which close() was invoked.

6.5.5

- The abstract implementation of equals() between (big) lists now uses type-specific access methods (as the compareTo() method was already doing) to avoid massive boxing/unboxing. Thanks to Adrien Grand for suggesting this improvement.

- FIFO array-based queues are now serializable.

6.5.4

- Further fixes related to NaNs in sorting.

- Fixed very old bug in FastByteArrayOutputStream.write(int). Thanks to Massimo Santini for reporting this bug.

- We now use Arrays.MAX_ARRAY_SIZE, which is equal to Integer.MAX_VALUE minus 8, to bound all array allocations. Previously, it might happen that grow() and other array-related functions could try to allocate an array of size Integer.MAX_VALUE, which is technically correct from the JLS, but will not work on most JVMs. The maximum length we use now is the same value as that used by java.util.ArrayList. Thanks to William Harvey for suggesting this change.

6.5.3

- Corrected erroneous introduction of compare() methods on integral classes (they appeared in Java 7).

6.5.2

- A few changes were necessary to make fastutil behave as Java on NaNs when sorting. Double.compareTo() and Float.compareTo() treat Double.NaN as greater than Double.POSITIVE_INFINITY, and fastutil was not doing it. As part of the change, now all comparisons between primitive types are performed using the compare() method of the wrapper class (microbenchmarks confirmed that there is no speed penalty for that, probably due to inlining or even intrinsification). Thanks to Adam Klein for reporting this bug.

- All quickSort() implementations that do not involve a comparator are now deprecated, as there are equivalent/better versions in java.util.Arrays.

6.5.0 -> 6.5.1

- Now FastBuffered{Input/Output}Stream has a constructor with an explicitly given buffer.
- Abandoned golden-ratio based expansion of arrays and lists in favour of a (more standard) doubling approach.
- Array-based FIFO queues now reduce their capacity automatically by halving when the size becomes one fourth of the length.
- The add() method for open hash maps has been deprecated and replaced by addTo(), as the name choice proved to be a recipe for disaster.
- New InspectableFileCachedInputStream for caching easily large byte streams partially on file and partially in memory.
- The front() method for semi-indirect heaps took no comparator, but was used in queues in which you could support a comparator. There is now a further version accepting a comparator.
- Serial Version UIDs are now private.

6.4.6 -> 6.5.0

- Fixed type of array hash strategies.
- Fixed use of equals() instead of compareTo() in SemiIndirectHeaps.front(). Thanks to Matthew Hatem for reporting this bug.
- Now we generate custom hash maps for primitive types, too (as we were already doing for sets).

6.4.5 -> 6.4.6

- In array-based priority queues changed() would not invalidate the cached index of the smallest element.

6.4.4 -> 6.4.5

- In some very rare circumstances, enumeration of hash sets or maps combined with massive element removal (using the iterator remove() method) could have led to inconsistent enumeration (duplicates and missing elements). Thanks to Hamish Morgan for reporting this bug.

6.4.3 -> 6.4.4

- Array-based maps were not implementing correctly `entrySet().contains()`, and as a consequence `equals()` between such maps was broken. Thanks to Benson Margulies for reporting this bug.

6.4.2 -> 6.4.3

- Now array-based priority queue cache their first element. Moreover, they implement the correct type-specific interface.

6.4.1 -> 6.4.2

- Now we have indirect lexicographical radix sort on pairs of arrays, mainly used to compute quickly Kendall's tau.
- New reverse method for arrays (useful for radix descending sorts).
- Radix sort (one or two arrays) for big arrays.
- Now radix sort uses correctly (minimally) sized support arrays when sorting subarrays.

6.4 -> 6.4.1

- Now we have a separate directory, settable in the makefile, to generate sources. This makes Maven integration easier.
- The store methods in TextIO for big arrays were broken.
- Now big-array lists implement the Stack interface.
- Fixed subtle bug in `rehash()` methods of big hash sets.

6.3 -> 6.4

- WARNING: Indirect queues must obviously have a way to determine whether an index is in the queue. It was an oversight in the interface design that a `contains()` method was not present. We took the risk of adding it now. At the same time, we modified `remove()` so that now returns a boolean specifying whether the index to be removed was actually in the queue, as this is more in line with the Java Collections Framework.
- Removed unused double-priority queue related classes.
- Now array-based sets and maps have a constructor based on `java.util.Collection` and `java.util.Map` (as for the other

kind of sets and maps).

- New doubly linked implementation for linked hash maps and sets. It uses twice the space for pointers, but mixes well with linear probing, so we have again constant-time true deletions. Moreover, iterators can be started from any key in constant time (albeit the first access to the index of the list iterator will require a linear scan, unless the iterator started from the first or the last key). Additional methods such as `getAndMoveToFirst()` make the creation of LRU caches very easy. Thanks to Brien Colwell for donating the code.
- Now object-based array FIFO queues provide `dequeue` methods. Moreover, they clean up the backing array after returning an object or when performing a `clear()`.
- New `get()` method in set implementations makes it possible to recover the actual object in the collection that is equal to the query key.
- A number of bugs were found and fixed by Christian Falz (thanks!). In all binary search code the "to" parameter was **inclusive**, but the documentation said **exclusive**, with obvious problems. Hash map iterators could return under some very subtle and almost irreproducible circumstances a previously deleted slot. Deleted hash map entries would return spurious null values.

6.2.2 -> 6.3

- We now have radix sort. It's much faster than quicksort, but it can only sort keys in their natural order. There are multiple-array and indirect (and possibly stable) versions available.
- There are now custom hash sets also for type-specific keys. This makes it possible to use hash sets to index data indirectly (e.g., using integer or long just as indices).
- Shuffling static methods for all kinds of (big) list and arrays.

6.2.1 -> 6.2.2

- A new `add()` method makes the usage of maps as counters easier and faster.

6.2.0 -> 6.2.1

- A very stupid bug was causing twice the rehashing that was necessary. Now insertions in hash-based classes are significantly faster.

6.1.0 -> 6.2.0

- A better structure of the scan loop for hash tables borrowed from HPPC (<http://labs.carrotsearch.com/hppc.html>) gives some speed improvement to hash-based classes.

6.0.0 -> 6.1.0

- Hash-based classes have been rewritten using linear probing and a good hash (MurmurHash3). The old classes can be still generated using the target oldsources.
- Bizarre queues (double- and sesqui-indirect) have been removed from the standard jar, but they can be still generated using the target oldsources.

5.1.5 -> 6.0.0

- WARNING: the jar file is now fastutil.jar (not fastutil5.jar), again.
- WARNING: now fastutil requires Java 6+.
- fastutil is now released under the Apache License 2.0.
- New framework for big arrays, represented as arrays-of-arrays. BigArrays and the type-specific counterparts provide static methods of all kinds.
- New Size64 interface for classes implementing big collections.
- New framework for big lists--lists with longs as indices. The only present implementation uses big arrays, but, for instance, Sux4J's succinct lists will be retrofitted to LongBigList (presently they implement LongBigList from dsutils, which will be deprecated).
- List.iterator() now returns a ListIterator. There is no real reason not to do this, and the API change is handled from an implementation viewpoint in AbstractList, so nobody should really notice.
- New Collections.asCollection(Iterable) method to expose iterables as collections (missing methods are computed using the iterator). This was also the occasion to streamline type-specific abstract collections, which now inherit from java.util.AbstractCollection, so we support contains, clear, etc. methods as long as there is an iterator.
- Fixed bugged array-based constructors of ArrayMap and ArraySet.
- Fixed bugged put/remove methods in abstract functions. Thanks to

Katja Filippova for reporting this bug.

- New front-coded lists use big arrays, so they can store much more (in fact, unlimited) data. Unfortunately, they are no longer serialisation-compatible with previous versions.
- New MeasurableStream interface that is implemented by MeasurableInputStream and by a new, analogous MeasurableOutputStream.
- Better FastBufferedOutputStream and FastByteArrayOutputStream that are measurable and positionable.
- Now all clone() methods override covariantly the default return type (Object).

5.1.4 -> 5.1.5

- ArraySet was implementing isEmpty() with inverted logic (thanks to Marko Srdanovic for reporting this bug).
- New constructor for FastMultiByteArrayInputStream: it takes a MeasurableInputStream and uses length() to determine the number of bytes to load into memory.

5.1.3 -> 5.1.4

- The implementation of RepositionableStream in FastByteArrayOutputStream was fraught with a horrendous bug (thanks to Claudio Corsi for reporting), in spite of extensive unit tests.

5.1.2 -> 5.1.3

- A bug existing since the first release was preventing tables larger than 2^{30} bits to work (the computation of the next bucket to look at would cause an integer overflow).
- FastByteArrayOutputStream now implements RepositionableStream.
- Type-specific versions of Iterable.
- Some methods (e.g., iterator() and values()) are now explicitly re-strengthened wherever necessary to avoid complaints about ambiguous method invocations by some compilers.
- The introduction of functions added several bugs to the empty/singleton map classes. Inheriting from the respective function counterparts left several methods underspecified (equals(), etc.). This has been (hopefully) fixed.

5.1.1 -> 5.1.2

- `FastBufferedInputStream` now supports `length()` by `FileChannel`-fetching on `FileInputStream` instances (it already used to support `position()` by the same mechanism).

5.1.0 -> 5.1.1

- Byte-array MG4J I/O classes have been moved here.

5.0.9 -> 5.1.0

- Fixed documentation for custom/noncustom maps (it was exchanged).
- New type-specific `entrySet()` methods that avoid complicated casting to get a type-specific `entryset`. Moreover, now `entrySet()` can return an object implementing `Fast(Sorted)EntrySet` to indicate that a `fastIterator()` method is available. Fast iterators can return always the same `Entry` object, suitably mutated. We thank Daniel Ramage for suggesting this feature.
- Several hundreds of new classes generated by the new `Function` interface, which represent mappings for which the entry set is not enumerable (e.g., hashes). Functions have their usual share of satellite objects (wrappers, etc.). There are no implementations--the main purpose of the new interfaces is to make `Sux4J` (<http://sux.dsi.unimi.it/>) more object-oriented.

5.0.8 -> 5.0.9

- Slightly reduced overhead for bound checks in heap-based queues.
- `BinIO` was loading byte arrays one byte at a time. Now some conditionally compiled code uses bulk-read methods instead. Moreover, horrible kluges to work around Java bug #6478546 have been included.

5.0.7 -> 5.0.8

- Faster array maps and sets: `System.arraycopy()` is very slow on small arrays (due to inherent costs of calling native code) and reflection-based array creation is a disaster. Now we use object arrays and loops.
- New `clone()` methods for array-based structures and custom serialisation.
- `FastBuffered*Stream` has been simplified and streamlined. No more block alignment.

5.0.6 -> 5.0.7

- Better algorithm for `front()` in heaps.
- New comprehensive collection of array-based maps and sets. The motivation behind such structures is the need for quick, low-footprint data structures for *very* small sets (say, less than 10 elements). For instance, in MG4J we were using sparse reference-based hash tables, but it turned out that `System.identityHashCode()` is *deadly* slow and scanning linearly an array searching for the desired element is significantly faster.

5.0.5 -> 5.0.6

- Due to erratic and unpredictable behaviour of `InputStream.skip()`, which does not correspond to its specification and Sun refuses to fix (see bug 6222822; don't be fooled by the "closed, fixed" label), `FastBufferedInputStream` now peeks at the underlying stream and if it is `System.in` it uses repeated reads. Moreover, it will use alternatively reads and skips to guarantee that the number of skipped bytes will be smaller than requested only if end of file has been reached.
- The insertion and key retrieval methods of hash-based structures are now protected and final.
- New `front()` method for indirect queues. It retrieves quickly the indices associated to elements equal to the top.
- First JUnit tests.

5.0.4 -> 5.0.5

- Fixed possible overflow in `FastBufferedInputStream.available()`.
- Indirect heaps have faster checks for elements belonging or not to the queue. In particular, we just rely on array access for detecting indices out of bounds. Profiling with LaMa4J showed that in some circumstances checking explicitly the indices were within bounds was taking more time than the actual heap inner workings.
- Fixed obnoxious bug dating to the first `fastutil` implementation. The macro `KEY_EQUALS_HASH(x,h,y)`, which checks for equality between `x` and `y` given that the hash of `x` is `h`, was evaluating `hashCode()` on `y` without guarantee that `y` was non-null. As a result, adding a null to a mapped followed by the insertion of an element with hash code 0 would have thrown a `NullPointerException`. The bug went unobserved for years because no one use nulls as keys, and was actually detected by a bug in BUbiNG's code (which was in turn mistakenly inserting nulls in a set).

5.0.3 -> 5.0.4

- Fixed missing declaration of generic type for HASH_STRATEGY.
- A new abstract class, MeasurableInputStream, is used for streams whose length and current position are always known. This actually was needed for BUbiNG development.
- New readLine() family of method for reading "lines" directly from a FastBufferedInputStream.
- In FastBufferedInputStream, reset() has been deprecated in favour of flush().
- Array-based lists of objects now reallocate the backing array using reflection **only** if they were created by wrapping. This won't change the previous behaviour, but at the price of a boolean per list we have unbelievably faster array reallocation.
- New explicit fast load factors in Hash.

5.0.2 -> 5.0.3

- Bizarrely, java.util.List re-specifies iterator(), even if it extends Collection. As a result, we need to re-strengthen it in type-specific lists.
- Fixed new horrible bug introduced by adding Booleans to BinIO and TextIO. Problem is, I didn't know #assert is cumulative.

5.0.1 -> 5.0.2

- Fixed bug in sorted maps key sets and values that would cause a stack overflow when calling size() and a few other methods.
- Fixed lack of booleans in BinIO and TextIO.
- BinIO now checks for too large files.

5.0 -> 5.0.1

- In BinIO, it was assumed that .SIZE would give the size of primitive types in **bytes**. Bad mistake.

4.4.3 -> 5.0

- Java 5 only!
- Support for generics. This led to a number of backward-incompatible changes:

- * toArray(Object[]) does not accept any longer null as an argument;
 - * singletons for empty collections (sets, lists, ecc.) are type-specific;
 - * iterators on sorted collections are bidirectional *by specification*;
 - * the new, covariantly stronger methods defined in all interfaces (e.g., iterator() returning a type-specific iterator) are now the default, and in the abstract classes the old methods (e.g., objectIterator()) now just delegate to the standard method, which is the contrary of what was happening before: you'll have to turn all methods such as objectIterator() in iterator(), etc.
 - * all deprecated methods have been dropped.
- Array growth functions now will return the correct empty array for object arrays (it used to return ObjectArrays.EMPTY_ARRAY).
 - Strategies are generic and no longer required to accept REMOVED.
 - Stale references could hang around in the nodePath array for Red-Black trees and map; this has been fixed.
 - The difference in semantics with the standard toArray(Object[]) specification, which has always been in place, is now exhaustively explained.
 - Major code cleanup (mostly code deletion) due to passing fastutil into Eclipse to check unused code, etc.

4.4.2 -> 4.4.3

- Important bug fix in FastBufferedInputStream.

4.4.1 -> 4.4.2

- New reset() method to invalidate the buffer of a FastBufferedInputStream, making it possible to read safely files written by other processes (given, of course, that you are synchronising the accesses).

4.4.0 -> 4.4.1

- New parallel-array constructor for all maps. Very useful for static final map initialisation.
- Following considerations in Jakarta Commons I/O, the standard buffer size has be lowered to 8Ki.
- Some arguments were declared as DataInputStream instead of DataInput.
- New methods for reading/writing objects from/to streams.

4.3.2 -> 4.4

- New static containers for reading and writing easily text and binary data streams. They load/save arrays, iterators etc. to buffered readers or streams.
- Moved here fast input/output buffered classes from MG4J. This makes fastutil self-contained.
- The trivial implementation of the type-specific iterator was missing from AbstractList.drv (surprisingly, not from the subclass implementation!).
- The sublist implementation in AbstractList.drv is now protected and static. The attributes are protected, too.
- Now we compare booleans (false<true). As a result, also lists of booleans do get lexicographical comparability.
- add(k) in AbstractList.drv now calls add(size(), k).
- Fixed error messages for out-of-bound indices in lists.

4.3.1 -> 4.3.2

- Fixed small innocuous bug: a code fragment related to non-linked hash table was generated for linked hash tables, too, do to a case type in a preprocessor directive. The code fragment, however, had no effect.
- Fixed memory leak in OpenHashMap: the remove() method was not clearing the key (whereas OpenHashSet was).

4.3 -> 4.3.1

- New fully indirect heap-based double priority queues.
- Fixed docs for queues: in 4.3, we were claiming that greater elements are dequeued first, while the opposite happens.

4.2 -> 4.3

- New full-fledged set of unmodifiable structures *and* iterators.
- Removed about a dozen spurious final method modifiers.
- Made rehash() protected, so that everybody can play with different rehashing strategies.

- trim() in array lists wasn't doing the right thing, because trim(int) wasn't doing it in the first place. Now if n is smaller than the size of the list, we trim at the list size (previously we were doing nothing).

- Analogously, trim() in hash-table-based structures was fixed so that trimming a table below its size will result in rehashing to the minimum possible size.

4.1 -> 4.2

- Improved array methods: now all methods on objects (e.g., grow()) return an array of the same type of the array that was passed to them, similarly to toArray() in collections.

- Fixed missing macro substitution for empty iterator methods. In any case, they were already deprecated.

4.0 -> 4.1

- New classes for custom hashing methods (mainly thought for arrays). Correspondingly, methods for arrays have been implemented in the static containers.

- BasicEntry now throws an UnsupportedOperationException on calls to setValue(). If you ever used that method, you got wierd results, as it does not update the underlying map. The method is now implemented correctly in open hash maps, in which previously did not correctly update the underlying map.

- Reimplemented copy of an entire array using clone().

- Fixed a bug in clear() for indirect heaps (the inversion array was not being cleared).

- Indirect priority queue interfaces now feature an optional allChanged() method that can be used to force a complete heap rebuild. It is implemented by all current array-based and head-based concrete classes.

3.1 -> 4.0

- IMPORTANT: The optimized methods that a type-specific must provide now include an addElements() method that quickly adds an array of elements. As usual, the method is fully implemented by the type-specific abstract lists.

- IMPORTANT: The abstract generic version of get(), put() and remove() for maps with non-object keys or values now always return null to denote a missing key. They used to return an object-wrapped default return value.

- Completely new and comprehensive implementation of priority queues, both direct and indirect. Implementations are by heaps and by flat arrays. There are also static containers with all relevant heap methods, for people wanting to do their own thing.
- New static containers for comparators.
- All singletons, empty sets and synchronized wrappers are public so you can inherit from them.
- Abstract maps now provide keySet() and values() based on entrySet().
- New abstract classes for sorted sets and maps with delegators to type-specific methods.
- New public methods in Arrays and in type-specific Arrays classes for checking ranges.
- New static methods for type-specific arrays that allow to grow, enlarge and trim them with ease.
- Clarified abstract implementation of default return values, and implemented clarified specification. Just a couple of method in hash maps were not already compliant.
- The pour() method now returns a list. The previous version was returning a linked hash set, which was rather nonsensical anyway, since an iterator build on the returned set could have been different from the original iterator. You can always pour an iterator into a set by providing the set explicitly.
- An exception-throwing implementation of some methods in AbstractSet was missing. Same for AbstractCollection, AbstractMap and AbstractList.
- New basic inner entry class for abstract maps, which makes it easier to write entrySet() methods for classes that do not have their own entries.
- Added missing get(Object) method in AbstractMap (just delegates to the type-specific version).
- For lazy people, now containsKey() and containsValue() in AbstractMap are defined by looking into keySet() and values().
- Fixed a few methods of EMPTY_LIST which were throwing exception semantically (see the introduction).
- The interval iterators are now list iterators, except for longs.

- Fixed a bug in size() for array lists (reducing the size of an array would lead to an exception).
- Fixed double bug in hash tables: first of all, on very small sizes adding growthFactor would have left the size unchanged, giving rise to infinite loops. (Thanks to Heikki Uusitalo for reporting this bug.) Second, growthFactor was not being used *at all* by hash maps.
- Fixed entries emitted by singleton maps. Now they are type-specific.
- Fixed a number of minor glitches in gencsource.sh, and added some comments.
- HashCommon.removed has been renamed HashCommon.REMOVED.
- Boolean objects are now generated using valueOf() instead of the constructor.
- New type-specific wrappers for list iterators.

3.0 -> 3.1

- IMPORTANT: it.unimi.dsi.fastutil.Iterators methods have been spread in type-specific static containers.
- New Stack interface, implemented by type-specific lists.
- New static container classes Collections, Sets, and Lists. Presently they just provide empty containers.
- New type-specific static contains (e.g., IntSets) providing singletons and synchronized wrappers.
- Entry sets now have entries that are equal() to entries coming from corresponding maps in java.util.
- Spelling everywhere changed to Pure American. "synchronized" in code and "synchronise" in text side-by-side were looking really wierd...

3.0 -> 3.0.1

- New unwrap() methods for type-specific collections.
- Fixed old-as-world-bug, apparently wide but that evidently no one ever noticed: AbstractMap was not serialisable, and, as a result, the default return value was not serialised (I find sincerely counterintuitive that making a class serialisable doesn't do the same for its supertypes). It wasn't ever even *documented* as preserved, so probably everyone thought this was my idea, too. Too bad this breaks once more serialisation compatibility. Since I had to break some serialisation anyway, I decided

to eliminate the residual serialisation of p in hash table classes, too (which breaks serialisation for all hash-based classes).

2.60 -> 3.0

- IMPORTANT: All classes have been repackaged following the type of elements/keys. Sources will have to be retouched (just to change the import clause) and recompiled.
- IMPORTANT: Because of an unavoidable name clash in the new type-specific list interface, the method `remove(int)` of `IntCollection` has been renamed `rem(int)`. The only really unpleasant effect is that you must use `rem(int)` on variables of type `IntCollection` that are not of type `IntSet` (as `IntSet` reinstates `remove(int)` in its right place)--for instance, `IntList`.
- Brand-new implementation of type-specific lists, with all the features you'd expect and more.
- Insertions for `readObject()` in hash tables are now handled in a special way (20% faster).
- Implemented linear-time tree reconstruction for `readObject()` (in practice, more than twice faster).
- Fixed a problem with serialisation of hash tables: the table would have been reloaded with the same p, even if it was preposterous. We still save p, however, to avoid breaking serialisation compatibility.
- Fixed missing implementation of type-specific sets, which should have extended type-specific collections, but they weren't.
- The default return value is now protected.
- New family of `pour()` methods that pour an iterator into a set.
- New programmable growth factor for hash-table-based classes.
- Eliminated a few useless method calls in tree map.
- Wide range of complex assertions, which are compiled in or out using the "private static final boolean" idiom.
- For references we now use `System.identityHashCode()`; this shouldn't change much, but it seems definitely more sensible.
- Fixed major bug in `subSet()/subMap()`: creating a `subMap` of a `tailMap` (or `headMap`) a right extreme (left, resp.) equal to 0 would have caused the creation of a `tailMap` (or `headMap`, resp.), discarding the extreme. Very,

very unlikely, but it happened in a test.

- Fixed small bug in standard `remove()` method of submaps, which would have returned a default return value wrapped in a suitable object instead of null on non-existing keys.

2.52 -> 2.60

- IMPORTANT: Major overhaul of iterators. Now iterators must be skippable, so previous implementation of type-specific iterator interfaces will not work. However, new abstract classes allow to build iterator with ease by providing for free the skipping logic, and many useful static methods in Iterators allow to generate type-specific iterators wrapping standard iterators, arrays, etc.
- Better strategy for `clear()` on hash tables: we don't do anything only if all entries are free (which means that an empty table with deleted entry will be cleared).

2.51 -> 2.52

- IMPORTANT: The package name has changed to `it.unimi.dsi.fastutil` to be uniform with JPackage conventions. However, this means that you must manually erase the old one and update your sources.
- `clear()` doesn't do anything on empty hash tables.

2.50 -> 2.51

- New `trim(int)` method to reduce a hash table size avoiding to make it too small.
- `serialVersionUID` is now fixed, to avoid future incompatibilities.

2.11 -> 2.50

- IMPORTANT: The Collection interface now prescribes an iterator method with a type-specific name (e.g., `intIterator()`) that returns directly a type-specific iterator.
- New Reference maps and sets that allow to store more quickly canonised objects.
- New linked maps mimicking `java.util`'s, but with a boatload of additional features.
- Small bug fix: the `get(Object)` method would return null instead of the default return value for maps with object keys.

- Major bug fix: iterating backwards on submaps was leading to unpredictable results.
- Major bug fix: cloning maps would have caused inconsistent behaviour.
- Major code redistribution: now whenever possible wrappers belong to abstract superclasses.

2.1 -> 2.11

- Now we cache the hash of an object before entering the hash table loop.

2.0 -> 2.1

- A simple optimisation in hash-table inner loops has given quite a performance boost under certain conditions (we do not compute the secondary hashing if it is not necessary). Inspired by Gnu Trove.
- The trim() method would have in fact trimmed nothing, just rehashed the table.
- The computed maxFill value was slightly too small.
- Also tree sets now have constructors from arrays.
- More internal methods have been made final.

1.3 -> 2.0

- ALL MAPS AND SETS HAVE NEW NAMES DEPENDING ON THE IMPLEMENTATION.
- Introducing new high-performance, low memory-footprint implementation of SortedMap and SortedSet.
- Two tree implementations are available: RB trees and AVL trees. Both implementations are threaded. See the README.
- Fixed a bug in hashCode() and contains() for HashMap.drv (it was considering keys only!).
- Fixed a bug in contains() for entrySet() in all maps (it was using VALUE_EQUAL to test equality for values given as objects).
- I realised that a default return value can be useful also for maps and sets returning objects, so now you have it. It is even independent for submaps and subsets.
- Classes are no longer final. The performance gain is around 1%, and the decrease in usefulness is orders of magnitudes greater.
- We now check equality using first hashCode() and then equals().
- The tests for speed now warm up the trees by doing repeated insertions and deletions, so that the benefits of a better balancing criterion are more evident.
- The regression tests are much more stringent.
- Fixed hashCode() for hash maps (wasn't conforming to the Map interface

specification).

- Implemented linear cloning for tree classes.

Found in path(s):

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/CHANGES

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2002-2017 Sebastiano Vigna

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

package PACKAGE;

import java.util.AbstractCollection;

/** An abstract class providing basic methods for collections implementing a type-specific interface.

*

* <p>In particular, this class provide { @link #iterator()}, { @code add()}, { @link #remove(Object)} and

* { @link #contains(Object)} methods that just call the type-specific counterpart.

*

* <p>Warning: Because of a name clash between the list and collection interfaces

* the type-specific deletion method of a type-specific abstract

* collection is { @code rem()}, rather than { @code remove()}. A

* subclass must thus override { @code rem()}, rather than

* { @code remove()}, to make all inherited methods work properly.

*/

public abstract class ABSTRACT_COLLECTION KEY_GENERIC extends

AbstractCollection<KEY_GENERIC_CLASS> implements COLLECTION KEY_GENERIC {

protected ABSTRACT_COLLECTION() {}

@Override

public abstract KEY_ITERATOR KEY_GENERIC iterator();

```

#if KEYS_PRIMITIVE

/** {@inheritDoc}
 *
 * <p>This implementation always throws an {@link UnsupportedOperationException}.
 */
@Override
public boolean add(final KEY_TYPE k) {
    throw new UnsupportedOperationException();
}

/** {@inheritDoc}
 *
 * <p>This implementation iterates over the elements in the collection,
 * looking for the specified element.
 */
@Override
public boolean contains(final KEY_TYPE k) {
    final KEY_ITERATOR iterator = iterator();
    while (iterator.hasNext()) if (k == iterator.NEXT_KEY()) return true;
    return false;
}

/** {@inheritDoc}
 *
 * <p>This implementation iterates over the elements in the collection,
 * looking for the specified element and tries to remove it.
 */
@Override
public boolean rem(final KEY_TYPE k) {
    final KEY_ITERATOR KEY_GENERIC iterator = iterator();
    while (iterator.hasNext())
        if (k == iterator.NEXT_KEY()) {
            iterator.remove();
            return true;
        }
    return false;
}

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead.
 */
@SuppressWarnings("deprecation")
@Deprecated
@Override
public boolean add(final KEY_GENERIC_CLASS key) {
    return COLLECTION.super.add(key);
}

```

```

}

/** { @inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead.
 */
@Deprecated
@Override
public boolean contains(final Object key) {
    return COLLECTION.super.contains(key);
}

/** { @inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead.
 */
@Deprecated
@Override
public boolean remove(final Object key) {
    return COLLECTION.super.remove(key);
}

@Override
public KEY_TYPE[] toArray(KEY_TYPE a[]) {
    if (a == null || a.length < size()) a = new KEY_TYPE[size()];
    ITERATORS.unwrap(iterator(), a);
    return a;
}

@Override
public KEY_TYPE[] TO_KEY_ARRAY() {
    return toArray((KEY_TYPE[]) null);
}

/** { @inheritDoc}
 * @deprecated Please use { @code toArray()} instead&mdash;this method is redundant and will be removed in the
 future.
 */
@Deprecated
@Override
public KEY_TYPE[] TO_KEY_ARRAY(final KEY_TYPE a[]) {
    return toArray(a);
}

@Override
public boolean addAll(final COLLECTION c) {
    boolean retVal = false;

    for(final KEY_ITERATOR i = c.iterator(); i.hasNext();)
        if (add(i.NEXT_KEY())) retVal = true;
}

```

```

return retVal;
}

@Override
public boolean containsAll(final COLLECTION c) {
    for(final KEY_ITERATOR i = c.iterator(); i.hasNext();)
        if (! contains(i.NEXT_KEY())) return false;
    return true;
}

```

```

@Override
public boolean removeAll(final COLLECTION c) {
    boolean retVal = false;
    for(final KEY_ITERATOR i = c.iterator(); i.hasNext();)
        if (rem(i.NEXT_KEY())) retVal = true;
    return retVal;
}

```

```

@Override
public boolean retainAll(final COLLECTION c) {
    boolean retVal = false;
    for(final KEY_ITERATOR i = iterator(); i.hasNext();)
        if (! c.contains(i.NEXT_KEY())) {
            i.remove();
            retVal = true;
        }
    return retVal;
}
#endif

```

```

@Override
public String toString() {
    final StringBuilder s = new StringBuilder();
    final KEY_ITERATOR KEY_GENERIC i = iterator();
    int n = size();
    KEY_TYPE k;
    boolean first = true;

    s.append("{}");

    while(n-- != 0) {
        if (first) first = false;
        else s.append(", ");
        k = i.NEXT_KEY();
        #if KEYS_REFERENCE
            if (this == k) s.append("(this collection)"); else
        #endif
            s.append(String.valueOf(k));
    }
}

```

```
}  
  
s.append("{}");  
return s.toString();  
}  
}
```

Found in path(s):

```
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-  
a775574/drv/AbstractCollection.drv
```

No license file was found, but licenses were detected in source scan.

```
/*  
* Copyright (C) 2002-2017 Sebastiano Vigna  
*  
* Licensed under the Apache License, Version 2.0 (the "License");  
* you may not use this file except in compliance with the License.  
* You may obtain a copy of the License at  
*  
* http://www.apache.org/licenses/LICENSE-2.0  
*  
* Unless required by applicable law or agreed to in writing, software  
* distributed under the License is distributed on an "AS IS" BASIS,  
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
* See the License for the specific language governing permissions and  
* limitations under the License.  
*/
```

```
package PACKAGE;
```

```
import it.unimi.dsi.fastutil.objects.ObjectBidirectionalIterable;  
import it.unimi.dsi.fastutil.objects.ObjectBidirectionalIterator;  
import it.unimi.dsi.fastutil.objects.ObjectSortedSet;  
import it.unimi.dsi.fastutil.objects.ObjectSortedSets;  
import PACKAGE.SORTED_MAP.FastSortedEntrySet;
```

```
import java.util.Comparator;  
import java.util.Map;  
import java.util.SortedMap;  
import java.util.NoSuchElementException;
```

```
/** A class providing static methods and objects that do useful things with type-specific sorted maps.  
*  
* @see java.util.Collections  
*/
```

```
public final class SORTED_MAPS {
```

```

private SORTED_MAPS() {}

/** Returns a comparator for entries based on a given comparator on keys.
 *
 * @param comparator a comparator on keys.
 * @return the associated comparator on entries.
 */
public static KEY_GENERIC Comparator<? super Map.Entry<KEY_GENERIC_CLASS, ?>>
entryComparator(final KEY_COMPARATOR KEY_SUPER_GENERIC comparator) {
    return (Comparator<Map.Entry<KEY_GENERIC_CLASS, ?>>) (x, y) ->
comparator.compare(KEY_CLASS2TYPE(x.getKey()), KEY_CLASS2TYPE(y.getKey()));
}

/** Returns a bidirectional iterator that will be { @linkplain FastSortedEntrySet fast}, if possible, on the { @linkplain
Map#entrySet() entry set} of the provided { @code map}.
 * @param map a map from which we will try to extract a (fast) bidirectional iterator on the entry set.
 * @return a bidirectional iterator on the entry set of the given map that will be fast, if possible.
 * @since 8.0.0
 */
SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
public static KEY_VALUE_GENERIC ObjectBidirectionalIterator<MAP.Entry KEY_VALUE_GENERIC>
fastIterator(SORTED_MAP KEY_VALUE_GENERIC map) {
    final ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> entries = map.ENTRYSET();
    return entries instanceof SORTED_MAP.FastSortedEntrySet ? ((SORTED_MAP.FastSortedEntrySet
KEY_VALUE_GENERIC) entries).fastIterator() : entries.iterator();
}

/** Returns an iterable yielding a bidirectional iterator that will be { @linkplain FastSortedEntrySet fast}, if possible,
on the { @linkplain Map#entrySet() entry set} of the provided { @code map}.
 * @param map a map from which we will try to extract an iterable yielding a (fast) bidirectional iterator on the
entry set.
 * @return an iterable yielding a bidirectional iterator on the entry set of the given map that will be fast, if possible.
 * @since 8.0.0
 */
SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
public static KEY_VALUE_GENERIC ObjectBidirectionalIterable<MAP.Entry KEY_VALUE_GENERIC>
fastIterable(SORTED_MAP KEY_VALUE_GENERIC map) {
    final ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> entries = map.ENTRYSET();
    return entries instanceof SORTED_MAP.FastSortedEntrySet ? ((SORTED_MAP.FastSortedEntrySet
KEY_VALUE_GENERIC) entries)::fastIterator : entries;
}

/** An immutable class representing an empty type-specific sorted map.
 *
 * <p>This class may be useful to implement your own in case you subclass
 * a type-specific sorted map.

```

```

*/

public static class EmptySortedMap KEY_VALUE_GENERIC extends MAPS.EmptyMap
KEY_VALUE_GENERIC implements SORTED_MAP KEY_VALUE_GENERIC, java.io.Serializable, Cloneable
{

private static final long serialVersionUID = -7046029254386353129L;

protected EmptySortedMap() {}

@Override
public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() { return null; }

@Override
public ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> ENTRYSET() { return
ObjectSortedSets.EMPTY_SET; }

#if KEYS_PRIMITIVE || VALUES_PRIMITIVE
/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
#else
/** {@inheritDoc} */
#endif
@Override
@Override
@Override
public ObjectSortedSet<Map.Entry<KEY_GENERIC_CLASS, VALUE_GENERIC_CLASS>> entrySet() { return
ObjectSortedSets.EMPTY_SET; }

SUPPRESS_WARNINGS_KEY_UNCHECKED
@Override
public SORTED_SET KEY_GENERIC keySet() { return SORTED_SETS.EMPTY_SET; }

SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
@Override
public SORTED_MAP KEY_VALUE_GENERIC subMap(final KEY_GENERIC_TYPE from, final
KEY_GENERIC_TYPE to) { return EMPTY_MAP; }

SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
@Override
public SORTED_MAP KEY_VALUE_GENERIC headMap(final KEY_GENERIC_TYPE to) { return
EMPTY_MAP; }

SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
@Override
public SORTED_MAP KEY_VALUE_GENERIC tailMap(final KEY_GENERIC_TYPE from) { return
EMPTY_MAP; }

```

```

@Override
public KEY_GENERIC_TYPE FIRST_KEY() { throw new NoSuchElementException(); }

@Override
public KEY_GENERIC_TYPE LAST_KEY() { throw new NoSuchElementException(); }

#if KEYS_PRIMITIVE
/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public SORTED_MAP KEY_VALUE_GENERIC headMap(KEY_GENERIC_CLASS oto) { return
headMap(KEY_CLASS2TYPE(oto)); }

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public SORTED_MAP KEY_VALUE_GENERIC tailMap(KEY_GENERIC_CLASS ofrom) { return
tailMap(KEY_CLASS2TYPE(ofrom)); }

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public SORTED_MAP KEY_VALUE_GENERIC subMap(KEY_GENERIC_CLASS ofrom,
KEY_GENERIC_CLASS oto) { return subMap(KEY_CLASS2TYPE(ofrom), KEY_CLASS2TYPE(oto)); }

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public KEY_GENERIC_CLASS firstKey() { return KEY2OBJ(FIRST_KEY()); }

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public KEY_GENERIC_CLASS lastKey() { return KEY2OBJ(LAST_KEY()); }
#endif
}

/** An empty sorted map (immutable). It is serializable and cloneable.
 */
SUPPRESS_WARNINGS_KEY_VALUE_RAWTYPES

```

```

public static final EmptySortedMap EMPTY_MAP = new EmptySortedMap();

#if KEYS_REFERENCE || VALUES_REFERENCE
/** Returns an empty sorted map (immutable). It is serializable and cloneable.
 *
 * <p>This method provides a typesafe access to {@link #EMPTY_MAP}.
 * @return an empty sorted map (immutable).
 */
@SuppressWarnings("unchecked")
public static KEY_VALUE_GENERIC SORTED_MAP KEY_VALUE_GENERIC emptyMap() {
    return EMPTY_MAP;
}
#endif

/** An immutable class representing a type-specific singleton sorted map.
 *
 * <p>This class may be useful to implement your own in case you subclass
 * a type-specific sorted map.
 */

public static class Singleton KEY_VALUE_GENERIC extends MAPS.Singleton KEY_VALUE_GENERIC
implements SORTED_MAP KEY_VALUE_GENERIC, java.io.Serializable, Cloneable {

    private static final long serialVersionUID = -7046029254386353129L;

    protected final KEY_COMPARATOR KEY_SUPER_GENERIC comparator;

    protected Singleton(final KEY_GENERIC_TYPE key, final VALUE_GENERIC_TYPE value,
KEY_COMPARATOR KEY_SUPER_GENERIC comparator) {
        super(key, value);
        this.comparator = comparator;
    }

    protected Singleton(final KEY_GENERIC_TYPE key, final VALUE_GENERIC_TYPE value) {
        this(key, value, null);
    }

    SUPPRESS_WARNINGS_KEY_UNCHECKED
    final int compare(final KEY_GENERIC_TYPE k1, final KEY_GENERIC_TYPE k2) {
        return comparator == null ? KEY_CMP(k1, k2) : comparator.compare(k1, k2);
    }

    @Override
    public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() { return comparator; }

    SUPPRESS_WARNINGS_KEY_UNCHECKED
    @Override
    public ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> ENTRYSET() { if (entries == null) entries =

```

```

ObjectSortedSets.singleton(new ABSTRACT_MAP.BasicEntry KEY_VALUE_GENERIC_DIAMOND(key,
value), entryComparator(comparator)); return (ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC>)entries; }

#if KEYS_PRIMITIVE || VALUES_PRIMITIVE
/** { @inheritDoc }
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
#else
/** { @inheritDoc } */
#endif
@Override
@SuppressWarnings({ "rawtypes", "unchecked" })
public ObjectSortedSet<Map.Entry<KEY_GENERIC_CLASS, VALUE_GENERIC_CLASS>> entrySet() { return
(ObjectSortedSet)ENTRYSET(); }

@Override
public SORTED_SET KEY_GENERIC keySet() { if (keys == null) keys = SORTED_SETS.singleton(key,
comparator); return (SORTED_SET KEY_GENERIC)keys; }

SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
@Override
public SORTED_MAP KEY_VALUE_GENERIC subMap(final KEY_GENERIC_TYPE from, final
KEY_GENERIC_TYPE to) { if (compare(from, key) <= 0 && compare(key, to) < 0) return this; return
EMPTY_MAP; }

SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
@Override
public SORTED_MAP KEY_VALUE_GENERIC headMap(final KEY_GENERIC_TYPE to) { if (compare(key,
to) < 0) return this; return EMPTY_MAP; }

SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
@Override
public SORTED_MAP KEY_VALUE_GENERIC tailMap(final KEY_GENERIC_TYPE from) { if (compare(from,
key) <= 0) return this; return EMPTY_MAP; }

@Override
public KEY_GENERIC_TYPE FIRST_KEY() { return key; }

@Override
public KEY_GENERIC_TYPE LAST_KEY() { return key; }

#if KEYS_PRIMITIVE
/** { @inheritDoc }
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public SORTED_MAP KEY_VALUE_GENERIC headMap(KEY_GENERIC_CLASS oto) { return
headMap(KEY_CLASS2TYPE(oto)); }

```

```

/** { @inheritDoc }
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public SORTED_MAP KEY_VALUE_GENERIC tailMap(KEY_GENERIC_CLASS ofrom) { return
tailMap(KEY_CLASS2TYPE(ofrom)); }

```

```

/** { @inheritDoc }
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public SORTED_MAP KEY_VALUE_GENERIC subMap(KEY_GENERIC_CLASS ofrom,
KEY_GENERIC_CLASS oto) { return subMap(KEY_CLASS2TYPE(ofrom), KEY_CLASS2TYPE(oto)); }

```

```

/** { @inheritDoc }
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public KEY_GENERIC_CLASS firstKey() { return KEY2OBJ(FIRST_KEY()); }

```

```

/** { @inheritDoc }
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public KEY_GENERIC_CLASS lastKey() { return KEY2OBJ(LAST_KEY()); }
#endif
}

```

/** Returns a type-specific immutable sorted map containing only the specified pair. The returned sorted map is serializable and cloneable.

*

* <p>Note that albeit the returned map is immutable, its default return value may be changed.

*

* @param key the only key of the returned sorted map.

* @param value the only value of the returned sorted map.

* @return a type-specific immutable sorted map containing just the pair { @code <key,value> }.

*/

```

public static KEY_VALUE_GENERIC SORTED_MAP KEY_VALUE_GENERIC singleton(final
KEY_GENERIC_CLASS key, VALUE_GENERIC_CLASS value) { return new Singleton
KEY_VALUE_GENERIC_DIAMOND(KEY_CLASS2TYPE(key), VALUE_CLASS2TYPE(value)); }

```

/** RETURNS a type-specific immutable sorted map containing only the specified pair. The returned sorted map is serializable and cloneable.

*

* <p>Note that albeit the returned map is immutable, its default return value may be changed.

*

```

* @param key the only key of the returned sorted map.
* @param value the only value of the returned sorted map.
* @param comparator the comparator to use in the returned sorted map.
* @return a type-specific immutable sorted map containing just the pair { @code &lt;key,value&gt; }.
*/

public static KEY_VALUE_GENERIC SORTED_MAP KEY_VALUE_GENERIC singleton(final
KEY_GENERIC_CLASS key, VALUE_GENERIC_CLASS value, KEY_COMPARATOR
KEY_SUPER_GENERIC comparator) { return new Singleton
KEY_VALUE_GENERIC_DIAMOND(KEY_CLASS2TYPE(key), VALUE_CLASS2TYPE(value), comparator);
}

#if KEYS_PRIMITIVE || VALUES_PRIMITIVE

/** Returns a type-specific immutable sorted map containing only the specified pair. The returned sorted map is
serializable and cloneable.
*
* <p>Note that albeit the returned map is immutable, its default return value may be changed.
*
* @param key the only key of the returned sorted map.
* @param value the only value of the returned sorted map.
* @return a type-specific immutable sorted map containing just the pair { @code &lt;key,value&gt; }.
*/

public static KEY_VALUE_GENERIC SORTED_MAP KEY_VALUE_GENERIC singleton(final
KEY_GENERIC_TYPE key, final VALUE_GENERIC_TYPE value) {
return new Singleton KEY_VALUE_GENERIC_DIAMOND(key, value);
}

/** Returns a type-specific immutable sorted map containing only the specified pair. The returned sorted map is
serializable and cloneable.
*
* <p>Note that albeit the returned map is immutable, its default return value may be changed.
*
* @param key the only key of the returned sorted map.
* @param value the only value of the returned sorted map.
* @param comparator the comparator to use in the returned sorted map.
* @return a type-specific immutable sorted map containing just the pair { @code &lt;key,value&gt; }.
*/

public static KEY_VALUE_GENERIC SORTED_MAP KEY_VALUE_GENERIC singleton(final
KEY_GENERIC_TYPE key, final VALUE_GENERIC_TYPE value, KEY_COMPARATOR
KEY_SUPER_GENERIC comparator) {
return new Singleton KEY_VALUE_GENERIC_DIAMOND(key, value, comparator);
}

#endif

```

```

/** A synchronized wrapper class for sorted maps. */

public static class SynchronizedSortedMap KEY_VALUE_GENERIC extends MAPS.SynchronizedMap
KEY_VALUE_GENERIC implements SORTED_MAP KEY_VALUE_GENERIC, java.io.Serializable {

    private static final long serialVersionUID = -7046029254386353129L;

    protected final SORTED_MAP KEY_VALUE_GENERIC sortedMap;

    protected SynchronizedSortedMap(final SORTED_MAP KEY_VALUE_GENERIC m, final Object sync) {
        super(m, sync);
        sortedMap = m;
    }

    protected SynchronizedSortedMap(final SORTED_MAP KEY_VALUE_GENERIC m) {
        super(m);
        sortedMap = m;
    }

    @Override
    public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() { synchronized(sync) { return
sortedMap.comparator(); } }

    @Override
    public ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> ENTRYSET() { if (entries == null) entries =
ObjectSortedSets.synchronize(sortedMap.ENTRYSET(), sync); return (ObjectSortedSet<MAP.Entry
KEY_VALUE_GENERIC>)entries; }

#if KEYS_PRIMITIVE || VALUES_PRIMITIVE
    /** {@inheritDoc}
     * @deprecated Please use the corresponding type-specific method instead. */
    @Deprecated
#else
    /** {@inheritDoc} */
#endif

    @Override
    @SuppressWarnings({ "rawtypes", "unchecked" })
    public ObjectSortedSet<Map.Entry<KEY_GENERIC_CLASS, VALUE_GENERIC_CLASS>> entrySet() { return
(ObjectSortedSet)ENTRYSET(); }

    @Override
    public SORTED_SET KEY_GENERIC keySet() { if (keys == null) keys =
SORTED_SETS.synchronize(sortedMap.keySet(), sync); return (SORTED_SET KEY_GENERIC)keys; }

    @Override
    public SORTED_MAP KEY_VALUE_GENERIC subMap(final KEY_GENERIC_TYPE from, final
KEY_GENERIC_TYPE to) { return new SynchronizedSortedMap

```

```
KEY_VALUE_GENERIC_DIAMOND(sortedMap.subMap(from, to), sync); }
```

```
@Override
```

```
public SORTED_MAP KEY_VALUE_GENERIC headMap(final KEY_GENERIC_TYPE to) { return new  
SynchronizedSortedMap KEY_VALUE_GENERIC_DIAMOND(sortedMap.headMap(to), sync); }
```

```
@Override
```

```
public SORTED_MAP KEY_VALUE_GENERIC tailMap(final KEY_GENERIC_TYPE from) { return new  
SynchronizedSortedMap KEY_VALUE_GENERIC_DIAMOND(sortedMap.tailMap(from), sync); }
```

```
@Override
```

```
public KEY_GENERIC_TYPE FIRST_KEY() { synchronized(sync) { return sortedMap.FIRST_KEY(); } }
```

```
@Override
```

```
public KEY_GENERIC_TYPE LAST_KEY() { synchronized(sync) { return sortedMap.LAST_KEY(); } }
```

```
#if KEYS_PRIMITIVE
```

```
/** { @inheritDoc }
```

```
 * @deprecated Please use the corresponding type-specific method instead. */
```

```
@Deprecated
```

```
@Override
```

```
public KEY_GENERIC_CLASS firstKey() { synchronized(sync) { return sortedMap.firstKey(); } }
```

```
/** { @inheritDoc }
```

```
 * @deprecated Please use the corresponding type-specific method instead. */
```

```
@Deprecated
```

```
@Override
```

```
public KEY_GENERIC_CLASS lastKey() { synchronized(sync) { return sortedMap.lastKey(); } }
```

```
/** { @inheritDoc }
```

```
 * @deprecated Please use the corresponding type-specific method instead. */
```

```
@Deprecated
```

```
@Override
```

```
public SORTED_MAP KEY_VALUE_GENERIC subMap(final KEY_GENERIC_CLASS from, final  
KEY_GENERIC_CLASS to) { return new SynchronizedSortedMap  
KEY_VALUE_GENERIC_DIAMOND(sortedMap.subMap(from, to), sync); }
```

```
/** { @inheritDoc }
```

```
 * @deprecated Please use the corresponding type-specific method instead. */
```

```
@Deprecated
```

```
@Override
```

```
public SORTED_MAP KEY_VALUE_GENERIC headMap(final KEY_GENERIC_CLASS to) { return new  
SynchronizedSortedMap KEY_VALUE_GENERIC_DIAMOND(sortedMap.headMap(to), sync); }
```

```
/** { @inheritDoc }
```

```
 * @deprecated Please use the corresponding type-specific method instead. */
```

```
@Deprecated
```

```
@Override
```

```

    public SORTED_MAP KEY_VALUE_GENERIC tailMap(final KEY_GENERIC_CLASS from) { return new
    SynchronizedSortedMap KEY_VALUE_GENERIC_DIAMOND(sortedMap.tailMap(from), sync); }
#endif

}

```

```

/** Returns a synchronized type-specific sorted map backed by the given type-specific sorted map.
 *
 * @param m the sorted map to be wrapped in a synchronized sorted map.
 * @return a synchronized view of the specified sorted map.
 * @see java.util.Collections#synchronizedSortedMap(SortedMap)
 */

```

```

public static KEY_VALUE_GENERIC SORTED_MAP KEY_VALUE_GENERIC synchronize(final
SORTED_MAP KEY_VALUE_GENERIC m) { return new SynchronizedSortedMap
KEY_VALUE_GENERIC_DIAMOND(m); }

```

```

/** Returns a synchronized type-specific sorted map backed by the given type-specific sorted map, using an
assigned object to synchronize.
 *
 * @param m the sorted map to be wrapped in a synchronized sorted map.
 * @param sync an object that will be used to synchronize the access to the sorted sorted map.
 * @return a synchronized view of the specified sorted map.
 * @see java.util.Collections#synchronizedSortedMap(SortedMap)
 */

```

```

public static KEY_VALUE_GENERIC SORTED_MAP KEY_VALUE_GENERIC synchronize(final
SORTED_MAP KEY_VALUE_GENERIC m, final Object sync) { return new SynchronizedSortedMap
KEY_VALUE_GENERIC_DIAMOND(m, sync); }

```

```

/** An unmodifiable wrapper class for sorted maps. */

```

```

public static class UnmodifiableSortedMap KEY_VALUE_GENERIC extends MAPS.UnmodifiableMap
KEY_VALUE_GENERIC implements SORTED_MAP KEY_VALUE_GENERIC, java.io.Serializable {

```

```

    private static final long serialVersionUID = -7046029254386353129L;

```

```

    protected final SORTED_MAP KEY_VALUE_GENERIC sortedMap;

```

```

    protected UnmodifiableSortedMap(final SORTED_MAP KEY_VALUE_GENERIC m) {
        super(m);
        sortedMap = m;
    }

```

```

    @Override

```

```

public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() { return sortedMap.comparator(); }

@Override
public ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> ENTRYSET() { if (entries == null) entries =
ObjectSortedSets.unmodifiable(sortedMap.ENTRYSET()); return (ObjectSortedSet<MAP.Entry
KEY_VALUE_GENERIC>)entries; }

#if KEYS_PRIMITIVE || VALUES_PRIMITIVE
/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
#else
/** {@inheritDoc} */
#endif
@Override
@SuppressWarnings({ "rawtypes", "unchecked" })
public ObjectSortedSet<Map.Entry<KEY_GENERIC_CLASS, VALUE_GENERIC_CLASS>> entrySet() { return
(ObjectSortedSet)ENTRYSET(); }

@Override
public SORTED_SET KEY_GENERIC keySet() { if (keys == null) keys =
SORTED_SETS.unmodifiable(sortedMap.keySet()); return (SORTED_SET KEY_GENERIC)keys; }

@Override
public SORTED_MAP KEY_VALUE_GENERIC subMap(final KEY_GENERIC_TYPE from, final
KEY_GENERIC_TYPE to) { return new UnmodifiableSortedMap
KEY_VALUE_GENERIC_DIAMOND(sortedMap.subMap(from, to)); }

@Override
public SORTED_MAP KEY_VALUE_GENERIC headMap(final KEY_GENERIC_TYPE to) { return new
UnmodifiableSortedMap KEY_VALUE_GENERIC_DIAMOND(sortedMap.headMap(to)); }

@Override
public SORTED_MAP KEY_VALUE_GENERIC tailMap(final KEY_GENERIC_TYPE from) { return new
UnmodifiableSortedMap KEY_VALUE_GENERIC_DIAMOND(sortedMap.tailMap(from)); }

@Override
public KEY_GENERIC_TYPE FIRST_KEY() { return sortedMap.FIRST_KEY(); }

@Override
public KEY_GENERIC_TYPE LAST_KEY() { return sortedMap.LAST_KEY(); }

#if KEYS_PRIMITIVE
/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public KEY_GENERIC_CLASS firstKey() { return sortedMap.firstKey(); }

```

```

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public KEY_GENERIC_CLASS lastKey() { return sortedMap.lastKey(); }

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public SORTED_MAP KEY_VALUE_GENERIC subMap(final KEY_GENERIC_CLASS from, final
KEY_GENERIC_CLASS to) { return new UnmodifiableSortedMap
KEY_VALUE_GENERIC_DIAMOND(sortedMap.subMap(from, to)); }

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public SORTED_MAP KEY_VALUE_GENERIC headMap(final KEY_GENERIC_CLASS to) { return new
UnmodifiableSortedMap KEY_VALUE_GENERIC_DIAMOND(sortedMap.headMap(to)); }

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public SORTED_MAP KEY_VALUE_GENERIC tailMap(final KEY_GENERIC_CLASS from) { return new
UnmodifiableSortedMap KEY_VALUE_GENERIC_DIAMOND(sortedMap.tailMap(from)); }
#endif

}

/** Returns an unmodifiable type-specific sorted map backed by the given type-specific sorted map.
 *
 * @param m the sorted map to be wrapped in an unmodifiable sorted map.
 * @return an unmodifiable view of the specified sorted map.
 * @see java.util.Collections#unmodifiableSortedMap(SortedMap)
 */
public static KEY_VALUE_GENERIC SORTED_MAP KEY_VALUE_GENERIC unmodifiable(final
SORTED_MAP KEY_VALUE_GENERIC m) { return new UnmodifiableSortedMap
KEY_VALUE_GENERIC_DIAMOND(m); }

#if defined(TEST) && ! KEY_CLASS_Reference

private static long seed = System.currentTimeMillis();

```

```

private static java.util.Random r = new java.util.Random(seed);

private static KEY_TYPE genKey() {
#if KEY_CLASS_Byte || KEY_CLASS_Short || KEY_CLASS_Character
    return (KEY_TYPE)(r.nextInt());
#elif KEYS_PRIMITIVE
    return r.NEXT_KEY();
#else
    return Integer.toBinaryString(r.nextInt());
#endif
}

private static VALUE_TYPE genValue() {
#if VALUE_CLASS_Byte || VALUE_CLASS_Short || VALUE_CLASS_Character
    return (VALUE_TYPE)(r.nextInt());
#elif VALUES_PRIMITIVE
    return r.NEXT_VALUE();
#elif !VALUE_CLASS_Reference || KEY_CLASS_Reference
    return Integer.toBinaryString(r.nextInt());
#else
    return new java.io.Serializable() {};
#endif
}

private static java.text.NumberFormat format = new java.text.DecimalFormat("#,###.00");
private static java.text.FieldPosition p = new java.text.FieldPosition(0);

private static String format(double d) {
    StringBuffer s = new StringBuffer();
    return format.format(d, s, p).toString();
}

private static void speedTest(int n, boolean comp) {
    System.out.println("There are presently no speed tests for this class.");
}

private static boolean valEquals(Object o1, Object o2) {
    return o1 == null ? o2 == null : o1.equals(o2);
}

private static void fatal(String msg) {
    System.out.println(msg);
    System.exit(1);
}

private static void ensure(boolean cond, String msg) {

```

```

if (cond) return;
fatal(msg);
}

private static Object[] k, v, nk;
private static KEY_TYPE kt[];
private static KEY_TYPE nkt[];
private static VALUE_TYPE vt[];
private static SORTED_MAP topMap;

protected static void testMaps(SORTED_MAP m, SortedMap t, int n, int level) {
    long ms;
    boolean mThrowsIllegal, tThrowsIllegal, mThrowsNoElement, tThrowsNoElement, mThrowsUnsupp,
    tThrowsUnsupp;
    Object rt = null, rm = null;

    if (level > 1) return;

    /* Now we check that both maps agree on first/last keys. */

    mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

    try {
        m.firstKey();
    }
    catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
    try {
        t.firstKey();
    }
    catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }

    ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): firstKey() divergence at
    start in java.util.NoSuchElementException (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");
    if (!mThrowsNoElement) ensure(t.firstKey().equals(m.firstKey()), "Error (" + level + ", " + seed + "): m and t differ
    at start on their first key (" + m.firstKey() + ", " + t.firstKey() + ")");

    mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

    try {
        m.lastKey();
    }
    catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
    try {
        t.lastKey();
    }
    catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }

```

```

ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): lastKey() divergence at start
in java.util.NoSuchElementException (" + mThrowsNoElement + ", " + tThrowsNoElement + "));

if (! mThrowsNoElement) ensure(t.lastKey().equals(m.lastKey()), "Error (" + level + ", " + seed + "): m and t differ
at start on their last key (" + m.lastKey() + ", " + t.lastKey() + "));

/* Now we check that m and t are equal. */
if (!m.equals(t) || ! t.equals(m)) System.err.println("m: " + m + " t: " + t);

ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) at start");
ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) at start");

/* Now we check that m actually holds that data. */
for(java.util.Iterator i=t.entrySet().iterator(); i.hasNext();) {
    java.util.Map.Entry e = (java.util.Map.Entry)i.next();
    ensure(valEquals(e.getValue(), m.get(e.getKey())), "Error (" + level + ", " + seed + "): m and t differ on an entry
("+e+") after insertion (iterating on t)");
}

/* Now we check that m actually holds that data, but iterating on m. */
for(java.util.Iterator i=m.entrySet().iterator(); i.hasNext();) {
    java.util.Map.Entry e = (java.util.Map.Entry)i.next();
    ensure(valEquals(e.getValue(), t.get(e.getKey())), "Error (" + level + ", " + seed + "): m and t differ on an entry
("+e+") after insertion (iterating on m)");
}

/* Now we check that m actually holds the same keys. */
for(java.util.Iterator i=t.keySet().iterator(); i.hasNext();) {
    Object o = i.next();
    ensure(m.containsKey(o), "Error (" + level + ", " + seed + "): m and t differ on a key (" +o+) after insertion
(iterating on t)");
    ensure(m.keySet().contains(o), "Error (" + level + ", " + seed + "): m and t differ on a key (" +o+", in keySet()) after
insertion (iterating on t)");
}

/* Now we check that m actually holds the same keys, but iterating on m. */
for(java.util.Iterator i=m.keySet().iterator(); i.hasNext();) {
    Object o = i.next();
    ensure(t.containsKey(o), "Error (" + level + ", " + seed + "): m and t differ on a key after insertion (iterating on m)");
    ensure(t.keySet().contains(o), "Error (" + level + ", " + seed + "): m and t differ on a key (in keySet()) after insertion
(iterating on m)");
}

```

```

/* Now we check that m actually hold the same values. */
for(java.util.Iterator i=t.values().iterator(); i.hasNext();) {
    Object o = i.next();
    ensure(m.containsValue(o), "Error (" + level + ", " + seed + "): m and t differ on a value after insertion (iterating on
t)");
    ensure(m.values().contains(o), "Error (" + level + ", " + seed + "): m and t differ on a value (in values()) after
insertion (iterating on t)");
}

/* Now we check that m actually hold the same values, but iterating on m. */
for(java.util.Iterator i=m.values().iterator(); i.hasNext();) {
    Object o = i.next();
    ensure(t.containsValue(o), "Error (" + level + ", " + seed + "): m and t differ on a value after insertion (iterating on
m)");
    ensure(t.values().contains(o), "Error (" + level + ", " + seed + "): m and t differ on a value (in values()) after
insertion (iterating on m)");
}

/* Now we check that inquiries about random data give the same answer in m and t. For
m we use the polymorphic method. */

for(int i=0; i<n; i++) {
    KEY_TYPE T = genKey();

    mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

    try {
        m.containsKey(KEY2OBJ(T));
    }
    catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
    catch (IllegalArgumentException e) { mThrowsIllegal = true; }

    try {
        t.containsKey(KEY2OBJ(T));
    }
    catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
    catch (IllegalArgumentException e) { tThrowsIllegal = true; }

    ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): containsKey() divergence in
java.util.NoSuchElementException (" + mThrowsNoElement + ", " + tThrowsNoElement + "));
    ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): containsKey() divergence in
IllegalArgumentException (" + mThrowsIllegal + ", " + tThrowsIllegal + "));
    if (!mThrowsNoElement && !mThrowsIllegal) {
        ensure(m.containsKey(KEY2OBJ(T)) == t.containsKey(KEY2OBJ(T)), "Error (" + level + ", " + seed + "):
divergence in keys between t and m (polymorphic method)");
    }

    #if KEY_CLASS_Object && ! (VALUES_REFERENCE)
        if ((m.GET_VALUE(T) != VALUE_NULL) != ((t.get(KEY2OBJ(T)) == null ? VALUE_NULL :

```

```

VALUE_OBJ2TYPE(t.get(KEY2OBJ(T)))) != VALUE_NULL) ||
    t.get(KEY2OBJ(T)) != null &&
    ! VALUE2OBJ(m.GET_VALUE(T)).equals(t.get(KEY2OBJ(T))))
#else
    if ((m.get(T) != VALUE_NULL) != ((t.get(KEY2OBJ(T)) == null ? VALUE_NULL :
VALUE_OBJ2TYPE(t.get(KEY2OBJ(T)))) != VALUE_NULL) ||
        t.get(KEY2OBJ(T)) != null &&
        ! m.get(KEY2OBJ(T)).equals(t.get(KEY2OBJ(T))))
#endif
    {
        System.out.println("Error (" + level + ", " + seed + "): divergence between t and m (polymorphic method)");
        System.exit(1);
    }
}
}

/* Again, we check that inquiries about random data give the same answer in m and t, but
for m we use the standard method. */

for(int i=0; i<n; i++) {
    KEY_TYPE T = genKey();

    mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

    try {
        m.get(KEY2OBJ(T));
    }
    catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
    catch (IllegalArgumentException e) { mThrowsIllegal = true; }

    try {
        t.get(KEY2OBJ(T));
    }
    catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
    catch (IllegalArgumentException e) { tThrowsIllegal = true; }

    ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): get() divergence in
java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");
    ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): get() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ")");
    if (!mThrowsNoElement && !mThrowsIllegal) ensure(valEquals(m.get(KEY2OBJ(T)), t.get(KEY2OBJ(T))),
"Error (" + level + ", " + seed + "): divergence between t and m (standard method)");
}

/* Now we put and remove random data in m and t, checking that the result is the same. */

for(int i=0; i<20*n; i++) {
    KEY_TYPE T = genKey();

```

```

VALUE_TYPE U = genValue();

mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = mThrowsUnsupp =
tThrowsUnsupp = false;

try {
    rm = m.put(KEY2OBJ(T), VALUE2OBJ(U));
}
catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
catch (IllegalArgumentException e) { mThrowsIllegal = true; }
catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

try {
    rt = t.put(KEY2OBJ(T), VALUE2OBJ(U));
}
catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
catch (IllegalArgumentException e) { tThrowsIllegal = true; }
catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }

ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): put() divergence in
java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");
ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): put() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ")");
//ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + level + ", " + seed + "): put() divergence in
UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);
if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsUnsupp) ensure(valEquals(rm, rt), "Error (" + level +
", " + seed + "): divergence in put() between t and m (" + rt + ", " + rm + ")");

T = genKey();

mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = mThrowsUnsupp =
tThrowsUnsupp = false;

try {
    rm = m.remove(KEY2OBJ(T));
}
catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
catch (IllegalArgumentException e) { mThrowsIllegal = true; }
catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

try {
    rt = t.remove(KEY2OBJ(T));
}
catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
catch (IllegalArgumentException e) { tThrowsIllegal = true; }
catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }

```

```

    ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): remove() divergence in
java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");
    ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): remove() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ")");
    //ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + level + ", " + seed + "): remove() divergence in
UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);
    if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsUnsupp) ensure(valEquals(rm, rt), "Error (" + level +
", " + seed + "): divergence in remove() between t and m (" + rt + ", " + rm + ")");
}

ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after removal");
ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after removal");

/* Now we check that m actually holds the same data. */

for(java.util.Iterator i=t.entrySet().iterator(); i.hasNext();) {
    java.util.Map.Entry e = (java.util.Map.Entry)i.next();
    ensure(valEquals(e.getValue(), m.get(e.getKey())), "Error (" + level + ", " + seed + "): m and t differ on an entry
("+e+") after removal (iterating on t)");
}

/* Now we check that m actually holds that data, but iterating on m. */

for(java.util.Iterator i=m.entrySet().iterator(); i.hasNext();) {
    java.util.Map.Entry e = (java.util.Map.Entry)i.next();
    ensure(valEquals(e.getValue(), t.get(e.getKey())), "Error (" + level + ", " + seed + "): m and t differ on an entry
("+e+") after removal (iterating on m)");
}

/* Now we check that m actually holds the same keys. */

for(java.util.Iterator i=t.keySet().iterator(); i.hasNext();) {
    Object o = i.next();
    ensure(m.containsKey(o), "Error (" + level + ", " + seed + "): m and t differ on a key (" + o + ") after removal (iterating
on t)");
    ensure(m.keySet().contains(o), "Error (" + level + ", " + seed + "): m and t differ on a key (" + o + ", in keySet()) after
removal (iterating on t)");
}

/* Now we check that m actually holds the same keys, but iterating on m. */

for(java.util.Iterator i=m.keySet().iterator(); i.hasNext();) {
    Object o = i.next();
    ensure(t.containsKey(o), "Error (" + level + ", " + seed + "): m and t differ on a key after removal (iterating on m)");
    ensure(t.keySet().contains(o), "Error (" + level + ", " + seed + "): m and t differ on a key (in keySet()) after removal
(iterating on m)");
}

```

```

/* Now we check that m actually hold the same values. */

for(java.util.Iterator i=t.values().iterator(); i.hasNext();) {
    Object o = i.next();
    ensure(m.containsValue(o), "Error (" + level + ", " + seed + "): m and t differ on a value after removal (iterating on
t)");
    ensure(m.values().contains(o), "Error (" + level + ", " + seed + "): m and t differ on a value (in values()) after
removal (iterating on t)");
}

/* Now we check that m actually hold the same values, but iterating on m. */

for(java.util.Iterator i=m.values().iterator(); i.hasNext();) {
    Object o = i.next();
    ensure(t.containsValue(o), "Error (" + level + ", " + seed + "): m and t differ on a value after removal (iterating on
m)");
    ensure(t.values().contains(o), "Error (" + level + ", " + seed + "): m and t differ on a value (in values()) after removal
(iterating on m)");
}

/* Now we check that both maps agree on first/last keys. */

mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

try {
    m.firstKey();
}
catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
try {
    t.firstKey();
}
catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }

ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): firstKey() divergence in
java.util.NoSuchElementException (" + mThrowsNoElement + ", " + tThrowsNoElement + "));
if (! mThrowsNoElement) ensure(t.firstKey().equals(m.firstKey()), "Error (" + level + ", " + seed + "): m and t differ
on their first key (" + m.firstKey() + ", " + t.firstKey() + "));

mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

try {
    m.lastKey();
}
catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
try {
    t.lastKey();
}

```

```

catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }

ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): lastKey() divergence in
java.util.NoSuchElementException (" + mThrowsNoElement + ", " + tThrowsNoElement + "));

if (! mThrowsNoElement) ensure(t.lastKey().equals(m.lastKey()), "Error (" + level + ", " + seed + "): m and t differ
on their last key (" + m.lastKey() + ", " + t.lastKey() + "));

/* Now we check cloning. */

if (level == 0) {
ensure(m.equals(((Singleton)m).clone()), "Error (" + level + ", " + seed + "): m does not equal m.clone()");
ensure(((Singleton)m).clone().equals(m), "Error (" + level + ", " + seed + "): m.clone() does not equal m");
m = (SORTED_MAP)((Singleton)m).clone();
}

int h = m.hashCode();

/* Now we save and read m. */

SORTED_MAP m2 = null;

try {
java.io.File ff = new java.io.File("it.unimi.dsi.fastutil.test");
java.io.OutputStream os = new java.io.FileOutputStream(ff);
java.io.ObjectOutputStream oos = new java.io.ObjectOutputStream(os);

oos.writeObject(m);
oos.close();

java.io.InputStream is = new java.io.FileInputStream(ff);
java.io.ObjectInputStream ois = new java.io.ObjectInputStream(is);

m2 = (SORTED_MAP)ois.readObject();
ois.close();
ff.delete();
}
catch(Exception e) {
e.printStackTrace();
System.exit(1);
}

#if !VALUE_CLASS_Reference
ensure(m2.hashCode() == h, "Error (" + level + ", " + seed + "): hashCode() changed after save/read");

/* Now we check that m2 actually holds that data. */

```

```

ensure(m2.equals(t), "Error (" + level + ", " + seed + "): ! m2.equals(t) after save/read");
ensure(t.equals(m2), "Error (" + level + ", " + seed + "): ! t.equals(m2) after save/read");
/* Now we take out of m everything, and check that it is empty. */
#endif

/* Now we play with iterators. */

{
java.util.ListIterator i, j;
Object J;
i = (java.util.ListIterator)m.entrySet().iterator();
j = new java.util.LinkedList(t.entrySet()).listIterator();

for(int k = 0; k < 2*n; k++) {
ensure(i.hasNext() == j.hasNext(), "Error (" + level + ", " + seed + "): divergence in hasNext()");
ensure(i.hasPrevious() == j.hasPrevious(), "Error (" + level + ", " + seed + "): divergence in hasPrevious()");

if (r.nextFloat() < .8 && i.hasNext()) {
ensure(((java.util.Map.Entry)i.next()).getKey().equals(J = ((Map.Entry)j.next()).getKey()), "Error (" + level + ", " +
seed + "): divergence in next()");
}
else if (r.nextFloat() < .2 && i.hasPrevious()) {
ensure(((java.util.Map.Entry)i.previous()).getKey().equals(J = ((Map.Entry)j.previous()).getKey()), "Error (" + level
+ ", " + seed + "): divergence in previous()");
}

ensure(i.nextIndex() == j.nextIndex(), "Error (" + level + ", " + seed + "): divergence in nextIndex()");
ensure(i.previousIndex() == j.previousIndex(), "Error (" + level + ", " + seed + "): divergence in previousIndex()");

}
}

{
boolean badPrevious = false;
Object previous = null;
it.unimi.dsi.fastutil.BidirectionalIterator i;
java.util.ListIterator j;
Object I, J;
KEY_TYPE from = genKey();
j = new java.util.LinkedList(t.keySet()).listIterator();
while(j.hasNext()) {
Object k = j.next();
if (((Comparable)k).compareTo(KEY2OBJ(from)) > 0) {
badPrevious = true;
j.previous();
break;
}
}
}

```

```

    }
    previous = k;
}

i = (it.unimi.dsi.fastutil.BidirectionalIterator)((SORTED_SET)m.keySet()).iterator(from);

for(int k = 0; k < 2*n; k++) {
    ensure(i.hasNext() == j.hasNext(), "Error (" + level + ", " + seed + "): divergence in hasNext() (iterator with starting
point " + from + ")");
    ensure(i.hasPrevious() == j.hasPrevious() || badPrevious && (i.hasPrevious() == (previous != null)), "Error (" +
level + ", " + seed + "): divergence in hasPrevious() (iterator with starting point " + from + ")" + badPrevious);

    if (r.nextFloat() < .8 && i.hasNext()) {
        ensure((I = i.next()).equals(J = j.next()), "Error (" + level + ", " + seed + "): divergence in next() (" + I + ", " + J + ",
iterator with starting point " + from + ")");
        //System.err.println("Done next " + I + " " + J + " " + badPrevious);

        badPrevious = false;

        if (r.nextFloat() < 0.5) {
            }
        }
        else if (!badPrevious && r.nextFloat() < .2 && i.hasPrevious()) {
            ensure((I = i.previous()).equals(J = j.previous()), "Error (" + level + ", " + seed + "): divergence in previous() (" + I +
", " + J + ", iterator with starting point " + from + ")");

            if (r.nextFloat() < 0.5) {
                }
            }
        }

}

/* Now we check that m actually holds that data. */

ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after iteration");
ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after iteration");

/* Now we select a pair of keys and create a submap. */

if (! m.isEmpty()) {
    java.util.ListIterator i;
    Object start = m.firstKey(), end = m.firstKey();
    for(i = (java.util.ListIterator)m.keySet().iterator(); i.hasNext() && r.nextFloat() < .3; start = end = i.next());
    for(; i.hasNext() && r.nextFloat() < .95; end = i.next());

    //System.err.println("Checking subMap from " + start + " to " + end + " (level=" + (level+1) + ")...");
    testMaps((SORTED_MAP)m.subMap((KEY_CLASS)start, (KEY_CLASS)end), t.subMap(start, end), n, level + 1);
}

```

```

ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after subMap");
ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after subMap");

//System.err.println("Checking headMap to " + end + " (level=" + (level+1) + ")...");
testMaps((SORTED_MAP)m.headMap((KEY_CLASS)end), t.headMap(end), n, level + 1);

ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after headMap");
ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after headMap");

//System.err.println("Checking tailMap from " + start + " (level=" + (level+1) + ")...");
testMaps((SORTED_MAP)m.tailMap((KEY_CLASS)start), t.tailMap(start), n, level + 1);

ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after tailMap");
ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after tailMap");
}
}

```

```

private static void test() {
    int n = 1;
    k = new Object[n];
    v = new Object[n];
    nk = new Object[n];
    kt = new KEY_TYPE[n];
    nkt = new KEY_TYPE[n];
    vt = new VALUE_TYPE[n];

    for(int i = 0; i < n; i++) {
#if KEY_CLASS_Object
        k[i] = kt[i] = genKey();
        nk[i] = nkt[i] = genKey();
#else
        k[i] = new KEY_CLASS(kt[i] = genKey());
        nk[i] = new KEY_CLASS(nkt[i] = genKey());
#endif
#if VALUES_REFERENCE
        v[i] = vt[i] = genValue();
#else
        v[i] = new VALUE_CLASS(vt[i] = genValue());
#endif
    }

    SORTED_MAP m = new Singleton(kt[0], vt[0]);
    topMap = m;
}

```

```

SortedMap t1 = new java.util.TreeMap();
t1.put(k[0], v[0]);
SortedMap t = java.util.Collections.unmodifiableSortedMap(t1);

testMaps(m, t, n, 0);

System.out.println("Test OK");
return;
}

public static void main(String args[]) {
if (args.length > 1) r = new java.util.Random(seed = Long.parseLong(args[1]));

try {
test();
} catch(Throwable e) {
e.printStackTrace(System.err);
System.err.println("seed: " + seed);
}
}

#endif

}

```

Found in path(s):

```
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-
a775574/drv/SortedMaps.drv
```

No license file was found, but licenses were detected in source scan.

```

/*
* Copyright (C) 2002-2017 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

```

```

package PACKAGE;

import java.util.Comparator;

/** A type-specific {@link Comparator}; provides methods to compare two primitive types both as objects
 * and as primitive types.
 *
 * <p>Note that {@code fastutil} provides a corresponding abstract class that
 * can be used to implement this interface just by specifying the type-specific
 * comparator.
 *
 * @see Comparator
 */

@FunctionalInterface
public interface KEY_COMPARATOR KEY_GENERIC extends Comparator<KEY_GENERIC_CLASS> {

    /** Compares its two primitive-type arguments for order. Returns a negative integer,
     * zero, or a positive integer as the first argument is less than, equal
     * to, or greater than the second.
     *
     * @see java.util.Comparator
     * @return a negative integer, zero, or a positive integer as the first
     * argument is less than, equal to, or greater than the second.
     */

    int compare(KEY_TYPE k1, KEY_TYPE k2);

    #if KEYS_PRIMITIVE
    /** {@inheritDoc}
     * <p>This implementation delegates to the corresponding type-specific method.
     * @deprecated Please use the corresponding type-specific method instead. */
    @Deprecated
    @Override
    default int compare(KEY_GENERIC_CLASS ok1, KEY_GENERIC_CLASS ok2) {
        return compare(ok1.KEY_VALUE(), ok2.KEY_VALUE());
    }
    #endif

}

```

Found in path(s):

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/drv/Comparator.drv

No license file was found, but licenses were detected in source scan.

/*

```
* Copyright (C) 2002-2017 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

```
package PACKAGE;
```

```
/** An abstract class facilitating the creation of type-specific { @link plain.it.unimi.dsi.fastutil.BidirectionalIterator
bidirectional iterators}.
```

```
*
* @deprecated As of fastutil 8 this class is no longer necessary, as its previous abstract
* methods are now default methods of the type-specific interface.
*/
```

```
@Deprecated
```

```
public abstract class KEY_ABSTRACT_BIDI_ITERATOR KEY_GENERIC extends
KEY_ABSTRACT_ITERATOR KEY_GENERIC implements KEY_BIDI_ITERATOR KEY_GENERIC {
    protected KEY_ABSTRACT_BIDI_ITERATOR() {}
}
```

```
Found in path(s):
```

```
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-
a775574/drv/AbstractBidirectionalIterator.drv
```

```
No license file was found, but licenses were detected in source scan.
```

```
/*
* Copyright (C) 2002-2017 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
```

* See the License for the specific language governing permissions and
* limitations under the License.
*/

```
package PACKAGE;
```

```
import java.util.Collection;  
import java.util.Comparator;  
import java.util.Iterator;  
import java.util.SortedSet;  
import java.util.NoSuchElementException;
```

```
/** A type-specific AVL tree set with a fast, small-footprint implementation.  
 *  
 * <p>The iterators provided by this class are type-specific { @link  
 * it.unimi.dsi.fastutil.BidirectionalIterator bidirectional iterators }.  
 * Moreover, the iterator returned by { @code iterator() } can be safely cast  
 * to a type-specific { @linkplain java.util.ListIterator list iterator }.  
 */
```

```
public class AVL_TREE_SET KEY_GENERIC extends ABSTRACT_SORTED_SET KEY_GENERIC  
implements java.io.Serializable, Cloneable, SORTED_SET KEY_GENERIC {
```

```
    /** A reference to the root entry. */  
    protected transient Entry KEY_GENERIC tree;
```

```
    /** Number of elements in this set. */  
    protected int count;
```

```
    /** The entry of the first element of this set. */  
    protected transient Entry KEY_GENERIC firstEntry;
```

```
    /** The entry of the last element of this set. */  
    protected transient Entry KEY_GENERIC lastEntry;
```

```
    /** This set's comparator, as provided in the constructor. */  
    protected Comparator<? super KEY_GENERIC_CLASS> storedComparator;
```

```
    /** This set's actual comparator; it may differ from { @link #storedComparator } because it is  
    always a type-specific comparator, so it could be derived from the former by wrapping. */  
    protected transient KEY_COMPARATOR KEY_SUPER_GENERIC actualComparator;
```

```
    private static final long serialVersionUID = -7046029254386353130L;
```

```
    {  
        allocatePaths();  
    }
```

```

/** Creates a new empty tree set.
 */

public AVL_TREE_SET() {
    tree = null;
    count = 0;
}

/** Generates the comparator that will be actually used.
 *
 * <p>When a given { @link Comparator} is specified and stored in { @link
 * #storedComparator}, we must check whether it is type-specific. If it is
 * so, we can use it directly, and we store it in { @link #actualComparator}. Otherwise,
 * we adapt it using a helper static method.
 */
private void setActualComparator() {
#if KEY_CLASS_Object
    actualComparator = storedComparator;
#else
    actualComparator = COMPARATORS.AS_KEY_COMPARATOR(storedComparator);
#endif
}

/** Creates a new empty tree set with the given comparator.
 *
 * @param c a { @link Comparator} (even better, a type-specific comparator).
 */

public AVL_TREE_SET(final Comparator<? super KEY_GENERIC_CLASS> c) {
    this();
    storedComparator = c;
    setActualComparator();
}

/** Creates a new tree set copying a given set.
 *
 * @param c a collection to be copied into the new tree set.
 */

public AVL_TREE_SET(final Collection<? extends KEY_GENERIC_CLASS> c) {
    this();
    addAll(c);
}

/** Creates a new tree set copying a given sorted set (and its { @link Comparator}).

```

```

*
* @param s a {@link SortedSet} to be copied into the new tree set.
*/

public AVL_TREE_SET(final SortedSet <KEY_GENERIC_CLASS> s) {
    this(s.comparator());
    addAll(s);
}

/** Creates a new tree set copying a given type-specific collection.
*
* @param c a type-specific collection to be copied into the new tree set.
*/

public AVL_TREE_SET(final COLLECTION KEY_EXTENDS_GENERIC c) {
    this();
    addAll(c);
}

/** Creates a new tree set copying a given type-specific sorted set (and its {@link Comparator}).
*
* @param s a type-specific sorted set to be copied into the new tree set.
*/

public AVL_TREE_SET(final SORTED_SET KEY_GENERIC s) {
    this(s.comparator());
    addAll(s);
}

/** Creates a new tree set using elements provided by a type-specific iterator.
*
* @param i a type-specific iterator whose elements will fill the set.
*/

public AVL_TREE_SET(final STD_KEY_ITERATOR KEY_EXTENDS_GENERIC i) {
    while(i.hasNext()) add(i.NEXT_KEY());
}

#if KEYS_PRIMITIVE

/** Creates a new tree set using elements provided by an iterator.
*
* @param i an iterator whose elements will fill the set.
*/

SUPPRESS_WARNINGS_KEY_UNCHECKED
public AVL_TREE_SET(final Iterator<?> i) {

```

```
    this(ITERATORS.AS_KEY_ITERATOR(i));
}
```

```
#endif
```

```
/** Creates a new tree set and fills it with the elements of a given array using a given {@link Comparator}.
 *
 * @param a an array whose elements will be used to fill the set.
 * @param offset the first element to use.
 * @param length the number of elements to use.
 * @param c a {@link Comparator} (even better, a type-specific comparator).
 */
```

```
public AVL_TREE_SET(final KEY_GENERIC_TYPE[] a, final int offset, final int length, final Comparator<?
super KEY_GENERIC_CLASS> c) {
    this(c);
    ARRAYS.ensureOffsetLength(a, offset, length);
    for(int i = 0; i < length; i++) add(a[offset + i]);
}
```

```
/** Creates a new tree set and fills it with the elements of a given array.
 *
 * @param a an array whose elements will be used to fill the set.
 * @param offset the first element to use.
 * @param length the number of elements to use.
 */
```

```
public AVL_TREE_SET(final KEY_GENERIC_TYPE[] a, final int offset, final int length) {
    this(a, offset, length, null);
}
```

```
/** Creates a new tree set copying the elements of an array.
 *
 * @param a an array to be copied into the new tree set.
 */
```

```
public AVL_TREE_SET(final KEY_GENERIC_TYPE[] a) {
    this();
    int i = a.length;
    while(i-- != 0) add(a[i]);
}
```

```
/** Creates a new tree set copying the elements of an array using a given {@link Comparator}.
 *
 * @param a an array to be copied into the new tree set.
```

```

* @param c a { @link Comparator} (even better, a type-specific comparator).
*/

public AVL_TREE_SET(final KEY_GENERIC_TYPE[] a, final Comparator<? super KEY_GENERIC_CLASS>
c) {
    this(c);
    int i = a.length;
    while(i-- != 0) add(a[i]);
}

/*
* The following methods implements some basic building blocks used by
* all accessors. They are (and should be maintained) identical to those used in AVLTreeMap.drv.
*
* The add()/remove() code is derived from Ben Pfaff's GNU libavl
* (http://www.msu.edu/~pfaffben/avl/). If you want to understand what's
* going on, you should have a look at the literate code contained therein
* first.
*/

/** Compares two keys in the right way.
*
* <p>This method uses the { @link #actualComparator} if it is non-{@code null}.
* Otherwise, it resorts to primitive type comparisons or to { @link Comparable#compareTo(Object) compareTo()}.
*
* @param k1 the first key.
* @param k2 the second key.
* @return a number smaller than, equal to or greater than 0, as usual
* (i.e., when  $k1 < k2$ ,  $k1 = k2$  or  $k1 > k2$ , respectively).
*/

SUPPRESS_WARNINGS_KEY_UNCHECKED
final int compare(final KEY_GENERIC_TYPE k1, final KEY_GENERIC_TYPE k2) {
    return actualComparator == null ? KEY_CMP(k1, k2) : actualComparator.compare(k1, k2);
}

/** Returns the entry corresponding to the given key, if it is in the tree; {@code null}, otherwise.
*
* @param k the key to search for.
* @return the corresponding entry, or { @code null} if no entry with the given key exists.
*/

```

```

private Entry KEY_GENERIC findKey(final KEY_GENERIC_TYPE k) {
    Entry KEY_GENERIC e = tree;
    int cmp;

    while (e != null && (cmp = compare(k, e.key)) != 0)
        e = cmp < 0 ? e.left() : e.right();

    return e;
}

/** Locates a key.
 *
 * @param k a key.
 * @return the last entry on a search for the given key; this will be
 * the given key, if it present; otherwise, it will be either the smallest greater key or the greatest smaller key.
 */

final Entry KEY_GENERIC locateKey(final KEY_GENERIC_TYPE k) {
    Entry KEY_GENERIC e = tree, last = tree;
    int cmp = 0;

    while (e != null && (cmp = compare(k, e.key)) != 0) {
        last = e;
        e = cmp < 0 ? e.left() : e.right();
    }

    return cmp == 0 ? e : last;
}

/** This vector remembers the path followed during the current insertion. It suffices for
    about 2<sup>32</sup> entries. */
private transient boolean dirPath[];

private void allocatePaths() {
    dirPath = new boolean[48];
}

@Override
public boolean add(final KEY_GENERIC_TYPE k) {

    if (tree == null) { // The case of the empty tree is treated separately.
        count++;
        tree = lastEntry = firstEntry = new Entry KEY_GENERIC_DIAMOND(k);
    }
    else {
        Entry KEY_GENERIC p = tree, q = null, y = tree, z = null, e = null, w = null;

```

```

int cmp, i = 0;

while(true) {
    if ((cmp = compare(k, p.key)) == 0) return false;

    if (p.balance() != 0) {
        i = 0;
        z = q;
        y = p;
    }

    if (dirPath[i++] = cmp > 0) {
        if (p.succ()) {
            count++;
            e = new Entry KEY_GENERIC_DIAMOND(k);

            if (p.right == null) lastEntry = e;

            e.left = p;
            e.right = p.right;

            p.right(e);

            break;
        }

        q = p;
        p = p.right;
    }
    else {
        if (p.pred()) {
            count++;
            e = new Entry KEY_GENERIC_DIAMOND(k);

            if (p.left == null) firstEntry = e;

            e.right = p;
            e.left = p.left;

            p.left(e);

            break;
        }

        q = p;
        p = p.left;
    }
}

```

```

p = y;
i = 0;

while(p != e) {
    if (dirPath[i]) p.incBalance();
    else p.decBalance();

    p = dirPath[i++] ? p.right : p.left;
}

if (y.balance() == -2) {
    Entry KEY_GENERIC x = y.left;

    if (x.balance() == -1) {
        w = x;
        if (x.succ()) {
            x.succ(false);
            y.pred(x);
        }
        else y.left = x.right;

        x.right = y;
        x.balance(0);
        y.balance(0);
    }
    else {
        assert x.balance() == 1;

        w = x.right;
        x.right = w.left;
        w.left = x;
        y.left = w.right;
        w.right = y;
        if (w.balance() == -1) {
            x.balance(0);
            y.balance(1);
        }
        else if (w.balance() == 0) {
            x.balance(0);
            y.balance(0);
        }
        else {
            x.balance(-1);
            y.balance(0);
        }
        w.balance(0);
    }
}

```

```

if (w.pred()) {
    x.succ(w);
    w.pred(false);
}
if (w.succ()) {
    y.pred(w);
    w.succ(false);
}

}
}
else if (y.balance() == +2) {
    Entry KEY_GENERIC x = y.right;

    if (x.balance() == 1) {
        w = x;
        if (x.pred()) {
            x.pred(false);
            y.succ(x);
        }
        else y.right = x.left;

        x.left = y;
        x.balance(0);
        y.balance(0);
    }
    else {
        assert x.balance() == -1;

        w = x.left;
        x.left = w.right;
        w.right = x;
        y.right = w.left;
        w.left = y;
        if (w.balance() == 1) {
            x.balance(0);
            y.balance(-1);
        }
        else if (w.balance() == 0) {
            x.balance(0);
            y.balance(0);
        }
        else {
            x.balance(1);
            y.balance(0);
        }
        w.balance(0);
    }
}

```

```

    if (w.pred()) {
        y.succ(w);
        w.pred(false);
    }
    if (w.succ()) {
        x.pred(w);
        w.succ(false);
    }

}
}
else return true;

if (z == null) tree = w;
else {
    if (z.left == y) z.left = w;
    else z.right = w;
}
}

return true;
}

/** Finds the parent of an entry.
 *
 * @param e a node of the tree.
 * @return the parent of the given node, or { @code null } for the root.
 */

private Entry KEY_GENERIC parent(final Entry KEY_GENERIC e) {
    if (e == tree) return null;

    Entry KEY_GENERIC x, y, p;
    x = y = e;

    while(true) {
        if (y.succ()) {
            p = y.right;
            if (p == null || p.left != e) {
                while(! x.pred()) x = x.left;
                p = x.left;
            }
        }
        return p;
    }
}

```

```

    }
    else if (x.pred()) {
        p = x.left;
        if (p == null || p.right != e) {
            while(! y.succ()) y = y.right;
            p = y.right;
        }
        return p;
    }
}

x = x.left;
y = y.right;
}
}

```

SUPPRESS_WARNINGS_KEY_UNCHECKED

@Override

```

public boolean remove(final KEY_TYPE k) {
    if (tree == null) return false;

```

```

    int cmp;
    Entry KEY_GENERIC p = tree, q = null;
    boolean dir = false;
    final KEY_GENERIC_TYPE kk = KEY_GENERIC_CAST k;

```

```

    while(true) {
        if ((cmp = compare(kk, p.key)) == 0) break;
        else if (dir = cmp > 0) {
            q = p;
            if ((p = p.right()) == null) return false;
        }
        else {
            q = p;
            if ((p = p.left()) == null) return false;
        }
    }
}

```

```

if (p.left == null) firstEntry = p.next();
if (p.right == null) lastEntry = p.prev();

```

```

if (p.succ()) {
    if (p.pred()) {
        if (q != null) {
            if (dir) q.succ(p.right);
            else q.pred(p.left);
        }
    }
    else tree = dir ? p.right : p.left;
}

```

```

}
else {
    p.prev().right = p.right;

    if (q != null) {
        if (dir) q.right = p.left;
        else q.left = p.left;
    }
    else tree = p.left;
}
}
else {
    Entry KEY_GENERIC r = p.right;

    if (r.pred()) {
        r.left = p.left;
        r.pred(p.pred());
        if (! r.pred()) r.prev().right = r;
        if (q != null) {
            if (dir) q.right = r;
            else q.left = r;
        }
        else tree = r;

        r.balance(p.balance());
        q = r;
        dir = true;
    }
    else {
        Entry KEY_GENERIC s;

        while(true) {
            s = r.left;
            if (s.pred()) break;
            r = s;
        }

        if (s.succ()) r.pred(s);
        else r.left = s.right;

        s.left = p.left;

        if (! p.pred()) {
            p.prev().right = s;
            s.pred(false);
        }
    }
}

```

```

s.right = p.right;
s.succ(false);

if (q != null) {
    if (dir) q.right = s;
    else q.left = s;
}
else tree = s;

s.balance(p.balance());
q = r;
dir = false;
}
}

Entry KEY_GENERIC y;

while(q != null) {
    y = q;
    q = parent(y);

    if (! dir) {
        dir = q != null && q.left != y;
        y.incBalance();

        if (y.balance() == 1) break;
        else if (y.balance() == 2) {

            Entry KEY_GENERIC x = y.right;
            assert x != null;

            if (x.balance() == -1) {
                Entry KEY_GENERIC w;

                assert x.balance() == -1;

                w = x.left;
                x.left = w.right;
                w.right = x;
                y.right = w.left;
                w.left = y;

                if (w.balance() == 1) {
                    x.balance(0);
                    y.balance(-1);
                }
                else if (w.balance() == 0) {
                    x.balance(0);

```

```

y.balance(0);
}
else {
    assert w.balance() == -1;

    x.balance(1);
    y.balance(0);
}

w.balance(0);

if (w.pred()) {
    y.succ(w);
    w.pred(false);
}
if (w.succ()) {
    x.pred(w);
    w.succ(false);
}

if (q != null) {
    if (dir) q.right = w;
    else q.left = w;
}
else tree = w;
}
else {
    if (q != null) {
        if (dir) q.right = x;
        else q.left = x;
    }
    else tree = x;

    if (x.balance() == 0) {
        y.right = x.left;
        x.left = y;
        x.balance(-1);
        y.balance(+1);
        break;
    }

    assert x.balance() == 1;

    if (x.pred()) {
        y.succ(true);
        x.pred(false);
    }
    else y.right = x.left;
}

```

```

x.left = y;
y.balance(0);
x.balance(0);
}
}
}
else {
dir = q != null && q.left != y;
y.decBalance();

if (y.balance() == -1) break;
else if (y.balance() == -2) {

Entry KEY_GENERIC x = y.left;
assert x != null;

if (x.balance() == 1) {
Entry KEY_GENERIC w;

assert x.balance() == 1;

w = x.right;
x.right = w.left;
w.left = x;
y.left = w.right;
w.right = y;

if (w.balance() == -1) {
x.balance(0);
y.balance(1);
}
else if (w.balance() == 0) {
x.balance(0);
y.balance(0);
}
else {
assert w.balance() == 1;

x.balance(-1);
y.balance(0);
}

w.balance(0);

if (w.pred()) {
x.succ(w);
w.pred(false);
}
}

```

```

    }
    if (w.succ()) {
        y.pred(w);
        w.succ(false);
    }

    if (q != null) {
        if (dir) q.right = w;
        else q.left = w;
    }
    else tree = w;
}
else {
    if (q != null) {
        if (dir) q.right = x;
        else q.left = x;
    }
    else tree = x;

    if (x.balance() == 0) {
        y.left = x.right;
        x.right = y;
        x.balance(+1);
        y.balance(-1);
        break;
    }

    assert x.balance() == -1;

    if (x.succ()) {
        y.pred(true);
        x.succ(false);
    }
    else y.left = x.right;

    x.right = y;
    y.balance(0);
    x.balance(0);
}
}
}

count--;
return true;
}

```

SUPPRESS_WARNINGS_KEY_UNCHECKED

```

@Override
public boolean contains(final KEY_TYPE k) {
    return findKey(KEY_GENERIC_CAST k) != null;
}

#if KEY_CLASS_Object
SUPPRESS_WARNINGS_KEY_UNCHECKED
public K get(final KEY_TYPE k) {
    final Entry KEY_GENERIC entry = findKey(KEY_GENERIC_CAST k);
    return entry == null ? null : entry.key;
}
#endif

@Override
public void clear() {
    count = 0;
    tree = null;
    firstEntry = lastEntry = null;
}

/** This class represent an entry in a tree set.
 *
 * <p>We use the only "metadata", i.e., {@link Entry#info}, to store
 * information about balance, predecessor status and successor status.
 *
 * <p>Note that since the class is recursive, it can be
 * considered equivalently a tree.
 */

private static final class Entry KEY_GENERIC implements Cloneable {
    /** If the bit in this mask is true, {@link #right} points to a successor. */
    private static final int SUCC_MASK = 1 << 31;
    /** If the bit in this mask is true, {@link #left} points to a predecessor. */
    private static final int PRED_MASK = 1 << 30;
    /** The bits in this mask hold the node balance info. You can get it just by casting to byte. */
    private static final int BALANCE_MASK = 0xFF;
    /** The key of this entry. */
    KEY_GENERIC_TYPE key;
    /** The pointers to the left and right subtrees. */
    Entry KEY_GENERIC left, right;
    /** This integers holds different information in different bits (see {@link #SUCC_MASK}, {@link
    #PRED_MASK} and {@link #BALANCE_MASK}). */
    int info;

    Entry() {}

    /** Creates a new entry with the given key.

```

```

*
* @param k a key.
*/
Entry(final KEY_GENERIC_TYPE k) {
    this.key = k;
    info = SUCC_MASK | PRED_MASK;
}

/** Returns the left subtree.
*
* @return the left subtree ({ @code null} if the left
* subtree is empty).
*/
Entry KEY_GENERIC left() {
    return (info & PRED_MASK) != 0 ? null : left;
}

/** Returns the right subtree.
*
* @return the right subtree ({ @code null} if the right
* subtree is empty).
*/
Entry KEY_GENERIC right() {
    return (info & SUCC_MASK) != 0 ? null : right;
}

/** Checks whether the left pointer is really a predecessor.
* @return true if the left pointer is a predecessor.
*/
boolean pred() {
    return (info & PRED_MASK) != 0;
}

/** Checks whether the right pointer is really a successor.
* @return true if the right pointer is a successor.
*/
boolean succ() {
    return (info & SUCC_MASK) != 0;
}

/** Sets whether the left pointer is really a predecessor.
* @param pred if true then the left pointer will be considered a predecessor.
*/
void pred(final boolean pred) {
    if (pred) info |= PRED_MASK;
    else info &= ~PRED_MASK;
}

```

```

/** Sets whether the right pointer is really a successor.
 * @param succ if true then the right pointer will be considered a successor.
 */
void succ(final boolean succ) {
    if (succ) info |= SUCC_MASK;
    else info &= ~SUCC_MASK;
}

/** Sets the left pointer to a predecessor.
 * @param pred the predecessor.
 */
void pred(final Entry KEY_GENERIC pred) {
    info |= PRED_MASK;
    left = pred;
}

/** Sets the right pointer to a successor.
 * @param succ the successor.
 */
void succ(final Entry KEY_GENERIC succ) {
    info |= SUCC_MASK;
    right = succ;
}

/** Sets the left pointer to the given subtree.
 * @param left the new left subtree.
 */
void left(final Entry KEY_GENERIC left) {
    info &= ~PRED_MASK;
    this.left = left;
}

/** Sets the right pointer to the given subtree.
 * @param right the new right subtree.
 */
void right(final Entry KEY_GENERIC right) {
    info &= ~SUCC_MASK;
    this.right = right;
}

/** Returns the current level of the node.
 * @return the current level of this node.
 */
int balance() {
    return (byte)info;
}

/** Sets the level of this node.

```

```

* @param level the new level of this node.
*/
void balance(int level) {
    info &= ~BALANCE_MASK;
    info |= (level & BALANCE_MASK);
}

/** Increments the level of this node. */
void incBalance() {
    info = info & ~BALANCE_MASK | ((byte)info + 1) & 0xFF;
}

/** Decrements the level of this node. */
protected void decBalance() {
    info = info & ~BALANCE_MASK | ((byte)info - 1) & 0xFF;
}

/** Computes the next entry in the set order.
 *
 * @return the next entry ({@code null}) if this is the last entry).
 */
Entry KEY_GENERIC next() {
    Entry KEY_GENERIC next = this.right;
    if ((info & SUCC_MASK) == 0) while ((next.info & PRED_MASK) == 0) next = next.left;
    return next;
}

/** Computes the previous entry in the set order.
 *
 * @return the previous entry ({@code null}) if this is the first entry).
 */
Entry KEY_GENERIC prev() {
    Entry KEY_GENERIC prev = this.left;
    if ((info & PRED_MASK) == 0) while ((prev.info & SUCC_MASK) == 0) prev = prev.right;
    return prev;
}

@Override
SUPPRESS_WARNINGS_KEY_UNCHECKED
public Entry KEY_GENERIC clone() {
    Entry KEY_GENERIC c;
    try {
        c = (Entry KEY_GENERIC)super.clone();
    }
    catch(CloneNotSupportedException cantHappen) {
        throw new InternalError();
    }
}

```

```

    }

    c.key = key;
    c.info = info;

    return c;
}

public boolean equals(final Object o) {
    if (!(o instanceof Entry)) return false;
    Entry KEY_GENERIC_WILDCARD e = (Entry KEY_GENERIC_WILDCARD)o;

    return KEY_EQUALS(key, e.key);
}

public int hashCode() {
    return KEY2JAVAHASH_NOT_NULL(key);
}

public String toString() {
    return String.valueOf(key);
}

/*
    public void prettyPrint() {
        prettyPrint(0);
    }

    public void prettyPrint(int level) {
        if (pred()) {
            for (int i = 0; i < level; i++)
                System.err.print(" ");
            System.err.println("pred: " + left);
        }
        else if (left != null)
            left.prettyPrint(level + 1);
        for (int i = 0; i < level; i++)
            System.err.print(" ");
        System.err.println(key + " (" + level() + ")");
        if (succ()) {
            for (int i = 0; i < level; i++)
                System.err.print(" ");
            System.err.println("succ: " + right);
        }
        else if (right != null)
            right.prettyPrint(level + 1);
    }
}

```

```

*/
}

/*
    public void prettyPrint() {
        System.err.println("size: " + count);
        if (tree != null) tree.prettyPrint();
    }
*/

@Override
public int size() {
    return count;
}

@Override
public boolean isEmpty() {
    return count == 0;
}

@Override
public KEY_GENERIC_TYPE FIRST() {
    if (tree == null) throw new NoSuchElementException();
    return firstEntry.key;
}

@Override
public KEY_GENERIC_TYPE LAST() {
    if (tree == null) throw new NoSuchElementException();
    return lastEntry.key;
}

/** An iterator on the whole range.
 *
 * <p>This class can iterate in both directions on a threaded tree.
 */

private class SetIterator implements KEY_LIST_ITERATOR KEY_GENERIC {
    /** The entry that will be returned by the next call to {@link java.util.ListIterator#previous()} (or {@code null} if
    no previous entry exists). */
    Entry KEY_GENERIC prev;
    /** The entry that will be returned by the next call to {@link java.util.ListIterator#next()} (or {@code null} if no
    next entry exists). */
    Entry KEY_GENERIC next;
    /** The last entry that was returned (or {@code null} if we did not iterate or used {@link #remove()}). */
    Entry KEY_GENERIC curr;
    /** The current index (in the sense of a {@link java.util.ListIterator}). Note that this value is not meaningful when

```

```

this { @link SetIterator } has been created using the nonempty constructor.*/
int index = 0;

SetIterator() {
    next = firstEntry;
}

SetIterator(final KEY_GENERIC_TYPE k) {
    if ((next = locateKey(k)) != null) {
        if (compare(next.key, k) <= 0) {
            prev = next;
            next = next.next();
        }
        else prev = next.prev();
    }
}

@Override
public boolean hasNext() { return next != null; }
@Override
public boolean hasPrevious() { return prev != null; }

void updateNext() { next = next.next(); }

Entry KEY_GENERIC nextEntry() {
    if (! hasNext()) throw new NoSuchElementException();
    curr = prev = next;
    index++;
    updateNext();
    return curr;
}

@Override
public KEY_GENERIC_TYPE NEXT_KEY() { return nextEntry().key; }
@Override
public KEY_GENERIC_TYPE PREV_KEY() { return previousEntry().key; }

void updatePrevious() { prev = prev.prev(); }

Entry KEY_GENERIC previousEntry() {
    if (! hasPrevious()) throw new NoSuchElementException();
    curr = next = prev;
    index--;
    updatePrevious();
    return curr;
}

@Override

```

```

public int nextIndex() { return index; }

@Override
public int previousIndex() { return index - 1; }

@Override
public void remove() {
    if (curr == null) throw new IllegalStateException();
    /* If the last operation was a next(), we are removing an entry that precedes
       the current index, and thus we must decrement it. */
    if (curr == prev) index--;
    next = prev = curr;
    updatePrevious();
    updateNext();
    AVL_TREE_SET.this.remove(curr.key);
    curr = null;
}
}

@Override
public KEY_BIDI_ITERATOR KEY_GENERIC iterator() { return new SetIterator(); }

@Override
public KEY_BIDI_ITERATOR KEY_GENERIC iterator(final KEY_GENERIC_TYPE from) { return new
SetIterator(from); }

@Override
public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() { return actualComparator; }

@Override
public SORTED_SET KEY_GENERIC headSet(final KEY_GENERIC_TYPE to) { return new
Subset(KEY_NULL, true, to, false); }

@Override
public SORTED_SET KEY_GENERIC tailSet(final KEY_GENERIC_TYPE from) { return new Subset(from,
false, KEY_NULL, true); }

@Override
public SORTED_SET KEY_GENERIC subSet(final KEY_GENERIC_TYPE from, final KEY_GENERIC_TYPE
to) { return new Subset(from, false, to, false); }

/** A subset with given range.
 *
 * <p>This class represents a subset. One has to specify the left/right
 * limits (which can be set to  $-\infty$  or  $\infty$ ). Since the subset is a
 * view on the set, at a given moment it could happen that the limits of
 * the range are not any longer in the main set. Thus, things such as
 * {@link java.util.SortedSet#first\(\)} or {@link java.util.SortedSet#size\(\)} must be always computed

```

```

* on-the-fly.
*/
private final class Subset extends ABSTRACT_SORTED_SET KEY_GENERIC implements java.io.Serializable,
SORTED_SET KEY_GENERIC {
    private static final long serialVersionUID = -7046029254386353129L;

    /** The start of the subset range, unless { @link #bottom } is true. */
    KEY_GENERIC_TYPE from;
    /** The end of the subset range, unless { @link #top } is true. */
    KEY_GENERIC_TYPE to;
    /** If true, the subset range starts from  $-\infty$ . */
    boolean bottom;
    /** If true, the subset range goes to  $\infty$ . */
    boolean top;

    /** Creates a new subset with given key range.
     *
     * @param from the start of the subset range.
     * @param bottom if true, the first parameter is ignored and the range starts from  $-\infty$ .
     * @param to the end of the subset range.
     * @param top if true, the third parameter is ignored and the range goes to  $\infty$ .
     */
    public Subset(final KEY_GENERIC_TYPE from, final boolean bottom, final KEY_GENERIC_TYPE to, final
boolean top) {
        if (! bottom && ! top && AVL_TREE_SET.this.compare(from, to) > 0) throw new
IllegalArgumentException("Start element (" + from + ") is larger than end element (" + to + ")");

        this.from = from;
        this.bottom = bottom;
        this.to = to;
        this.top = top;
    }

    @Override
    public void clear() {
        final SubsetIterator i = new SubsetIterator();
        while(i.hasNext()) {
            i.NEXT_KEY();
            i.remove();
        }
    }

    /** Checks whether a key is in the subset range.
     * @param k a key.
     * @return true if is the key is in the subset range.
     */
    final boolean in(final KEY_GENERIC_TYPE k) {
        return (bottom || AVL_TREE_SET.this.compare(k, from) >= 0) &&

```

```
(top || AVL_TREE_SET.this.compare(k, to) < 0);  
}
```

```
SUPPRESS_WARNINGS_KEY_UNCHECKED
```

```
@Override
```

```
public boolean contains(final KEY_TYPE k) {  
    return in(KEY_GENERIC_CAST k) && AVL_TREE_SET.this.contains(k);  
}
```

```
@Override
```

```
public boolean add(final KEY_GENERIC_TYPE k) {  
    if (! in(k)) throw new IllegalArgumentException("Element (" + k + ") out of range [" + (bottom ? "-" :  
String.valueOf(from)) + ", " + (top ? "-" : String.valueOf(to)) + ")");  
    return AVL_TREE_SET.this.add(k);  
}
```

```
SUPPRESS_WARNINGS_KEY_UNCHECKED
```

```
@Override
```

```
public boolean remove(final KEY_TYPE k) {  
    if (! in(KEY_GENERIC_CAST k)) return false;  
    return AVL_TREE_SET.this.remove(k);  
}
```

```
@Override
```

```
public int size() {  
    final SubsetIterator i = new SubsetIterator();  
    int n = 0;
```

```
    while(i.hasNext()) {  
        n++;  
        i.NEXT_KEY();  
    }
```

```
    return n;  
}
```

```
@Override
```

```
public boolean isEmpty() { return ! new SubsetIterator().hasNext(); }
```

```
@Override
```

```
public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() { return actualComparator; }
```

```
@Override
```

```
public KEY_BIDI_ITERATOR KEY_GENERIC iterator() { return new SubsetIterator(); }
```

```
@Override
```

```
public KEY_BIDI_ITERATOR KEY_GENERIC iterator(final KEY_GENERIC_TYPE from) { return new  
SubsetIterator(from); }
```

```

@Override
public SORTED_SET KEY_GENERIC headSet(final KEY_GENERIC_TYPE to) {
    if (top) return new Subset(from, bottom, to, false);
    return compare(to, this.to) < 0 ? new Subset(from, bottom, to, false) : this;
}

@Override
public SORTED_SET KEY_GENERIC tailSet(final KEY_GENERIC_TYPE from) {
    if (bottom) return new Subset(from, false, to, top);
    return compare(from, this.from) > 0 ? new Subset(from, false, to, top) : this;
}

@Override
public SORTED_SET KEY_GENERIC subSet(KEY_GENERIC_TYPE from, KEY_GENERIC_TYPE to) {
    if (top && bottom) return new Subset(from, false, to, false);
    if (! top) to = compare(to, this.to) < 0 ? to : this.to;
    if (! bottom) from = compare(from, this.from) > 0 ? from : this.from;
    if (! top && ! bottom && from == this.from && to == this.to) return this;
    return new Subset(from, false, to, false);
}

/** Locates the first entry.
 *
 * @return the first entry of this subset, or { @code null } if the subset is empty.
 */
public AVL_TREE_SET.Entry KEY_GENERIC firstEntry() {
    if (tree == null) return null;
    // If this subset goes to -infinity, we return the main set first entry; otherwise, we locate the start of the set.
    AVL_TREE_SET.Entry KEY_GENERIC e;
    if (bottom) e = firstEntry();
    else {
        e = locateKey(from);
        // If we find either the start or something greater we're OK.
        if (compare(e.key, from) < 0) e = e.next();
    }
    // Finally, if this subset doesn't go to infinity, we check that the resulting key isn't greater than the end.
    if (e == null || ! top && compare(e.key, to) >= 0) return null;
    return e;
}

/** Locates the last entry.
 *
 * @return the last entry of this subset, or { @code null } if the subset is empty.
 */
public AVL_TREE_SET.Entry KEY_GENERIC lastEntry() {
    if (tree == null) return null;
    // If this subset goes to infinity, we return the main set last entry; otherwise, we locate the end of the set.

```

```

AVL_TREE_SET.Entry KEY_GENERIC e;
if (top) e = lastEntry;
else {
    e = locateKey(to);
    // If we find something smaller than the end we're OK.
    if (compare(e.key, to) >= 0) e = e.prev();
}
// Finally, if this subset doesn't go to -infinity, we check that the resulting key isn't smaller than the start.
if (e == null || !bottom && compare(e.key, from) < 0) return null;
return e;
}

@Override
public KEY_GENERIC_TYPE FIRST() {
    AVL_TREE_SET.Entry KEY_GENERIC e = firstEntry();
    if (e == null) throw new NoSuchElementException();
    return e.key;
}

@Override
public KEY_GENERIC_TYPE LAST() {
    AVL_TREE_SET.Entry KEY_GENERIC e = lastEntry();
    if (e == null) throw new NoSuchElementException();
    return e.key;
}

/** An iterator for subranges.
 *
 * <p>This class inherits from { @link SetIterator}, but overrides the methods that
 * update the pointer after a { @link java.util.ListIterator#next()} or { @link java.util.ListIterator#previous()}. If we
 would
 * move out of the range of the subset we just overwrite the next or previous
 * entry with { @code null}.
 */
private final class SubsetIterator extends SetIterator {
    SubsetIterator() {
        next = firstEntry();
    }

    SubsetIterator(final KEY_GENERIC_TYPE k) {
        this();

        if (next != null) {
            if (!bottom && compare(k, next.key) < 0) prev = null;
            else if (!top && compare(k, (prev = lastEntry()).key) >= 0) next = null;
            else {
                next = locateKey(k);
            }
        }
    }
}

```

```

    if (compare(next.key, k) <= 0) {
        prev = next;
        next = next.next();
    }
    else prev = next.prev();
}
}
}

@Override
void updatePrevious() {
    prev = prev.prev();
    if (! bottom && prev != null && AVL_TREE_SET.this.compare(prev.key, from) < 0) prev = null;
}

@Override
void updateNext() {
    next = next.next();
    if (! top && next != null && AVL_TREE_SET.this.compare(next.key, to) >= 0) next = null;
}
}
}

/** Returns a deep copy of this tree set.
 *
 * <p>This method performs a deep copy of this tree set; the data stored in the
 * set, however, is not cloned. Note that this makes a difference only for object keys.
 *
 * @return a deep copy of this tree set.
 */

@Override
SUPPRESS_WARNINGS_KEY_UNCHECKED
public Object clone() {
    AVL_TREE_SET KEY_GENERIC c;
    try {
        c = (AVL_TREE_SET KEY_GENERIC)super.clone();
    }
    catch(CloneNotSupportedException cantHappen) {
        throw new InternalError();
    }

    c.allocatePaths();

    if (count != 0) {
        // Also this apparently unfathomable code is derived from GNU libavl.
        Entry KEY_GENERIC e, p, q, rp = new Entry KEY_GENERIC_DIAMOND(), rq = new Entry
        KEY_GENERIC_DIAMOND();

```

```

p = rp;
rp.left(tree);

q = rq;
rq.pred(null);

while(true) {
if (! p.pred()) {
e = p.left.clone();
e.pred(q.left);
e.succ(q);
q.left(e);

p = p.left;
q = q.left;
}
else {
while(p.succ()) {
p = p.right;

if (p == null) {
q.right = null;
c.tree = rq.left;

c.firstEntry = c.tree;
while(c.firstEntry.left != null) c.firstEntry = c.firstEntry.left;
c.lastEntry = c.tree;
while(c.lastEntry.right != null) c.lastEntry = c.lastEntry.right;

return c;
}
q = q.right;
}

p = p.right;
q = q.right;
}

if (! p.succ()) {
e = p.right.clone();
e.succ(q.right);
e.pred(q);
q.right(e);
}
}
}

```

```

return c;
}

private void writeObject(java.io.ObjectOutputStream s) throws java.io.IOException {
    int n = count;
    SetIterator i = new SetIterator();

    s.defaultWriteObject();
    while(n-- != 0) s.WRITE_KEY(i.NEXT_KEY());
}

/** Reads the given number of entries from the input stream, returning the corresponding tree.
 *
 * @param s the input stream.
 * @param n the (positive) number of entries to read.
 * @param pred the entry containing the key that precedes the first key in the tree.
 * @param succ the entry containing the key that follows the last key in the tree.
 */
SUPPRESS_WARNINGS_KEY_UNCHECKED
private Entry KEY_GENERIC readTree(final java.io.ObjectInputStream s, final int n, final Entry KEY_GENERIC
pred, final Entry KEY_GENERIC succ) throws java.io.IOException, ClassNotFoundException {
    if (n == 1) {
        final Entry KEY_GENERIC top = new Entry KEY_GENERIC_DIAMOND(KEY_GENERIC_CAST
s.READ_KEY());
        top.pred(pred);
        top.succ(succ);

        return top;
    }

    if (n == 2) {
        /* We handle separately this case so that recursion will
        *always* be on nonempty subtrees. */
        final Entry KEY_GENERIC top = new Entry KEY_GENERIC_DIAMOND(KEY_GENERIC_CAST
s.READ_KEY());
        top.right(new Entry KEY_GENERIC_DIAMOND(KEY_GENERIC_CAST s.READ_KEY()));
        top.right.pred(top);
        top.balance(1);
        top.pred(pred);
        top.right.succ(succ);

        return top;
    }

    // The right subtree is the largest one.
    final int rightN = n / 2, leftN = n - rightN - 1;

```

```

final Entry KEY_GENERIC top = new Entry KEY_GENERIC_DIAMOND();

top.left(readTree(s, leftN, pred, top));

top.key = KEY_GENERIC_CAST s.READ_KEY();

top.right(readTree(s, rightN, top, succ));

if (n == (n & -n)) top.balance(1); // Quick test for determining whether n is a power of 2.

return top;
}

private void readObject(java.io.ObjectInputStream s) throws java.io.IOException, ClassNotFoundException {
    s.defaultReadObject();
    /* The storedComparator is now correctly set, but we must restore
       on-the-fly the actualComparator. */
    setActualComparator();
    allocatePaths();

    if (count != 0) {
        tree = readTree(s, count, null, null);
        Entry KEY_GENERIC e;

        e = tree;
        while(e.left() != null) e = e.left();
        firstEntry = e;

        e = tree;
        while(e.right() != null) e = e.right();
        lastEntry = e;
    }
}

#ifdef ASSERTS_CODE
private static KEY_GENERIC int checkTree(Entry KEY_GENERIC e) {
    if (e == null) return 0;

    final int leftN = checkTree(e.left()), rightN = checkTree(e.right());
    if (leftN + e.balance() != rightN)
        throw new AssertionError("Mismatch between left tree size (" + leftN + "), right tree size (" + rightN + ") and
balance (" + e.balance() + ")");

    return Math.max(leftN, rightN) + 1;
}
#endif

```

```

#ifdef TEST

private static long seed = System.currentTimeMillis();
private static java.util.Random r = new java.util.Random(seed);

private static KEY_TYPE genKey() {
#ifdef KEY_CLASS_Byte || KEY_CLASS_Short || KEY_CLASS_Character
return (KEY_TYPE)(r.nextInt());
#elif KEYS_PRIMITIVE
return r.NEXT_KEY();
#else
return Integer.toBinaryString(r.nextInt());
#endif
}

private static java.text.NumberFormat format = new java.text.DecimalFormat("#,###.00");
private static java.text.FieldPosition p = new java.text.FieldPosition(0);

private static String format(double d) {
StringBuffer s = new StringBuffer();
return format.format(d, s, p).toString();
}

private static void speedTest(int n, boolean comp) {
int i, j;
AVL_TREE_SET m;
java.util.TreeSet t;
KEY_TYPE k[] = new KEY_TYPE[n];
KEY_TYPE nk[] = new KEY_TYPE[n];
long ms;

for(i = 0; i < n; i++) {
k[i] = genKey();
nk[i] = genKey();
}

double totAdd = 0, totYes = 0, totNo = 0, totIterFor = 0, totIterBack = 0, totRemYes = 0, d, dd;

if (comp) {
for(j = 0; j < 20; j++) {

t = new java.util.TreeSet();

/* We first add all pairs to t. */
for(i = 0; i < n; i++) t.add(KEY2OBJ(k[i]));

/* Then we remove the first half and put it back. */

```

```

for(i = 0; i < n/2; i++) t.remove(KEY2OBJ(k[i]));

ms = System.currentTimeMillis();
for(i = 0; i < n/2; i++) t.add(KEY2OBJ(k[i]));
d = System.currentTimeMillis() - ms;

/* Then we remove the other half and put it back again. */
ms = System.currentTimeMillis();
for(i = n/2; i < n; i++) t.remove(KEY2OBJ(k[i]));
dd = System.currentTimeMillis() - ms ;

ms = System.currentTimeMillis();
for(i = n/2; i < n; i++) t.add(KEY2OBJ(k[i]));
d += System.currentTimeMillis() - ms;
if (j > 2) totAdd += n/d;
System.out.print("Add: " + format(n/d) + " K/s ");

/* Then we remove again the first half. */
ms = System.currentTimeMillis();
for(i = 0; i < n/2; i++) t.remove(KEY2OBJ(k[i]));
dd += System.currentTimeMillis() - ms ;
if (j > 2) totRemYes += n/dd;
System.out.print("RemYes: " + format(n/dd) + " K/s ");

/* And then we put it back. */
for(i = 0; i < n/2; i++) t.add(KEY2OBJ(k[i]));

/* We check for pairs in t. */
ms = System.currentTimeMillis();
for(i = 0; i < n; i++) t.contains(KEY2OBJ(k[i]));
d = 1.0 * n / (System.currentTimeMillis() - ms);
if (j > 2) totYes += d;
System.out.print("Yes: " + format(d) + " K/s ");

/* We check for pairs not in t. */
ms = System.currentTimeMillis();
for(i = 0; i < n; i++) t.contains(KEY2OBJ(nk[i]));
d = 1.0 * n / (System.currentTimeMillis() - ms);
if (j > 2) totNo += d;
System.out.print("No: " + format(d) + " K/s ");

/* We iterate on t. */
ms = System.currentTimeMillis();
for(Iterator it = t.iterator(); it.hasNext(); it.next());
d = 1.0 * n / (System.currentTimeMillis() - ms);
if (j > 2) totIterFor += d;
System.out.print("IterFor: " + format(d) + " K/s ");

```

```

    System.out.println();
}

System.out.println();
System.out.println("java.util Add: " + format(totAdd/(j-3)) + " K/s RemYes: " + format(totRemYes/(j-3)) + " K/s
Yes: " + format(totYes/(j-3)) + " K/s No: " + format(totNo/(j-3)) + " K/s IterFor: " + format(totIterFor/(j-3)) + "
K/s");

System.out.println();

totAdd = totYes = totNo = totIterFor = totIterBack = totRemYes = 0;

}

for(j = 0; j < 20; j++) {

    m = new AVL_TREE_SET();

    /* We first add all pairs to m. */
    for(i = 0; i < n; i++) m.add(k[i]);

    /* Then we remove the first half and put it back. */
    for(i = 0; i < n/2; i++) m.remove(k[i]);

    ms = System.currentTimeMillis();
    for(i = 0; i < n/2; i++) m.add(k[i]);
    d = System.currentTimeMillis() - ms;

    /* Then we remove the other half and put it back again. */
    ms = System.currentTimeMillis();
    for(i = n/2; i < n; i++) m.remove(k[i]);
    dd = System.currentTimeMillis() - ms ;

    ms = System.currentTimeMillis();
    for(i = n/2; i < n; i++) m.add(k[i]);
    d += System.currentTimeMillis() - ms;
    if (j > 2) totAdd += n/d;
    System.out.print("Add: " + format(n/d) + " K/s ");

    /* Then we remove again the first half. */
    ms = System.currentTimeMillis();
    for(i = 0; i < n/2; i++) m.remove(k[i]);
    dd += System.currentTimeMillis() - ms ;
    if (j > 2) totRemYes += n/dd;
    System.out.print("RemYes: " + format(n/dd) + " K/s ");

    /* And then we put it back. */
    for(i = 0; i < n/2; i++) m.add(k[i]);

```

```

/* We check for pairs in m. */
ms = System.currentTimeMillis();
for(i = 0; i < n; i++) m.contains(k[i]);
d = 1.0 * n / (System.currentTimeMillis() - ms);
if (j > 2) totYes += d;
System.out.print("Yes: " + format(d) + " K/s ");

/* We check for pairs not in m. */
ms = System.currentTimeMillis();
for(i = 0; i < n; i++) m.contains(nk[i]);
d = 1.0 * n / (System.currentTimeMillis() - ms);
if (j > 2) totNo += d;
System.out.print("No: " + format(d) + " K/s ");

/* We iterate on m. */
KEY_LIST_ITERATOR it = (KEY_LIST_ITERATOR)m.iterator();
ms = System.currentTimeMillis();
for(; it.hasNext(); it.NEXT_KEY());
d = 1.0 * n / (System.currentTimeMillis() - ms);
if (j > 2) totIterFor += d;
System.out.print("IterFor: " + format(d) + " K/s ");

/* We iterate back on m. */
ms = System.currentTimeMillis();
for(; it.hasPrevious(); it.PREV_KEY());
d = 1.0 * n / (System.currentTimeMillis() - ms);
if (j > 2) totIterBack += d;
System.out.print("IterBack: " + format(d) + " K/s ");

System.out.println();
}

System.out.println();
System.out.println("fastutil Add: " + format(totAdd/(j-3)) + " K/s RemYes: " + format(totRemYes/(j-3)) + " K/s
Yes: " + format(totYes/(j-3)) + " K/s No: " + format(totNo/(j-3)) + " K/s IterFor: " + format(totIterFor/(j-3)) + " K/s
IterBack: " + format(totIterBack/(j-3)) + " K/s");

System.out.println();
}

private static boolean valEquals(Object o1, Object o2) {
return o1 == null ? o2 == null : o1.equals(o2);
}

private static void fatal(String msg) {

```

```

System.out.println(msg);
System.exit(1);
}

private static void ensure(boolean cond, String msg) {
    if (cond) return;
    fatal(msg);
}

private static Object[] k, v, nk;
private static KEY_TYPE kt[];
private static KEY_TYPE nkt[];
private static AVL_TREE_SET topSet;

protected static void testSets(SORTED_SET m, SortedSet t, int n, int level) {
    long ms;
    boolean mThrowsIllegal, tThrowsIllegal, mThrowsNoElement, tThrowsNoElement;
    boolean rt = false, rm = false;

    if (level > 4) return;

    /* Now we check that both sets agree on first/last keys. */

    mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

    try {
        m.first();
    }
    catch (NoSuchElementException e) { mThrowsNoElement = true; }
    try {
        t.first();
    }
    catch (NoSuchElementException e) { tThrowsNoElement = true; }

    ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): first() divergence at start in
NoSuchElementException (" + mThrowsNoElement + ", " + tThrowsNoElement + "));
    if (! mThrowsNoElement) ensure(t.first().equals(m.first()), "Error (" + level + ", " + seed + "): m and t differ at start
on their first key (" + m.first() + ", " + t.first() + "));

    mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

    try {
        m.last();
    }
    catch (NoSuchElementException e) { mThrowsNoElement = true; }
    try {
        t.last();

```

```

}
catch (NoSuchElementException e) { tThrowsNoElement = true; }

ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): last() divergence at start in
NoSuchElementException (" + mThrowsNoElement + ", " + tThrowsNoElement + "));

if (! mThrowsNoElement) ensure(t.last().equals(m.last()), "Error (" + level + ", " + seed + "): m and t differ at start
on their last key (" + m.last() + ", " + t.last() + "));

/* Now we check that m and t are equal. */
if (!m.equals(t) || ! t.equals(m)) System.err.println("m: " + m + " t: " + t);

ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) at start");
ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) at start");

/* Now we check that m actually holds that data. */
for(Iterator i=t.iterator(); i.hasNext(); ) {
    ensure(m.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after insertion (iterating on
t)");
}

/* Now we check that m actually holds that data, but iterating on m. */
for(Iterator i=m.iterator(); i.hasNext(); ) {
    ensure(t.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after insertion (iterating on
m)");
}

/* Now we check that inquiries about random data give the same answer in m and t. For
m we use the polymorphic method. */

for(int i=0; i<n; i++) {
    KEY_TYPE T = genKey();

    mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

    try {
        m.contains(T);
    }
    catch (NoSuchElementException e) { mThrowsNoElement = true; }
    catch (IllegalArgumentException e) { mThrowsIllegal = true; }

    try {
        t.contains(KEY2OBJ(T));
    }
}

```

```

catch (NoSuchElementException e) { tThrowsNoElement = true; }
catch (IllegalArgumentException e) { tThrowsIllegal = true; }

ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): contains() divergence in
NoSuchElementException (" + mThrowsNoElement + ", " + tThrowsNoElement + "));
ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): contains() divergence in
IllegalArgumentException (" + mThrowsIllegal + ", " + tThrowsIllegal + "));
if (!mThrowsNoElement && !mThrowsIllegal) ensure(m.contains(KEY2OBJ(T)) == t.contains(KEY2OBJ(T)),
"Error (" + level + ", " + seed + "): divergence in keys between t and m (polymorphic method)");
}

/* Again, we check that inquiries about random data give the same answer in m and t, but
for m we use the standard method. */

for(int i=0; i<n; i++) {
KEY_TYPE T = genKey();

mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

try {
m.contains(KEY2OBJ(T));
}
catch (NoSuchElementException e) { mThrowsNoElement = true; }
catch (IllegalArgumentException e) { mThrowsIllegal = true; }

try {
t.contains(KEY2OBJ(T));
}
catch (NoSuchElementException e) { tThrowsNoElement = true; }
catch (IllegalArgumentException e) { tThrowsIllegal = true; }

ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): contains() divergence in
NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + "));
ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): contains() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + "));
if (!mThrowsNoElement && !mThrowsIllegal) ensure(m.contains(KEY2OBJ(T)) == t.contains(KEY2OBJ(T)),
"Error (" + level + ", " + seed + "): divergence between t and m (standard method)");
}

/* Now we add and remove random data in m and t, checking that the result is the same. */

for(int i=0; i<20*n; i++) {
KEY_TYPE T = genKey();

mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

try {
rm = m.add(KEY2OBJ(T));

```

```

    }
    catch (NoSuchElementException e) { mThrowsNoElement = true; }
    catch (IllegalArgumentException e) { mThrowsIllegal = true; }

    try {
        rt = t.add(KEY2OBJ(T));
    }
    catch (NoSuchElementException e) { tThrowsNoElement = true; }
    catch (IllegalArgumentException e) { tThrowsIllegal = true; }

    ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): add() divergence in
    NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");
    ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): add() divergence in
    IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ")");
    if (!mThrowsNoElement && !mThrowsIllegal) ensure(rm == rt, "Error (" + level + ", " + seed + "): divergence in
    add() between t and m");

    T = genKey();

    mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

    try {
        rm = m.remove(KEY2OBJ(T));
    }
    catch (NoSuchElementException e) { mThrowsNoElement = true; }
    catch (IllegalArgumentException e) { mThrowsIllegal = true; }

    try {
        rt = t.remove(KEY2OBJ(T));
    }
    catch (NoSuchElementException e) { tThrowsNoElement = true; }
    catch (IllegalArgumentException e) { tThrowsIllegal = true; }

    ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): remove() divergence in
    NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");
    ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): remove() divergence in
    IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ")");
    if (!mThrowsNoElement && !mThrowsIllegal) ensure(rm == rt, "Error (" + level + ", " + seed + "): divergence in
    remove() between t and m");
    }

    ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after removal");
    ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after removal");

    /* Now we check that m actually holds the same data. */

```

```

for(Iterator i=t.iterator(); i.hasNext();) {
    ensure(m.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after removal (iterating on
t)");
}

/* Now we check that m actually holds that data, but iterating on m. */

for(Iterator i=m.iterator(); i.hasNext();) {
    ensure(t.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after removal (iterating on
m)");
}

/* Now we check that both sets agree on first/last keys. */

mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

try {
    m.first();
}
catch (NoSuchElementException e) { mThrowsNoElement = true; }
try {
    t.first();
}
catch (NoSuchElementException e) { tThrowsNoElement = true; }

ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): first() divergence in
NoSuchElementException (" + mThrowsNoElement + ", " + tThrowsNoElement + "));
if (! mThrowsNoElement) ensure(t.first().equals(m.first()), "Error (" + level + ", " + seed + "): m and t differ on their
first key (" + m.first() + ", " + t.first() + "));

mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

try {
    m.last();
}
catch (NoSuchElementException e) { mThrowsNoElement = true; }
try {
    t.last();
}
catch (NoSuchElementException e) { tThrowsNoElement = true; }

ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): last() divergence in
NoSuchElementException (" + mThrowsNoElement + ", " + tThrowsNoElement + "));

if (! mThrowsNoElement) ensure(t.last().equals(m.last()), "Error (" + level + ", " + seed + "): m and t differ on their
last key (" + m.last() + ", " + t.last() + "));

/* Now we check cloning. */

```

```

if (level == 0) {
    ensure(m.equals(((AVL_TREE_SET)m).clone()), "Error (" + level + ", " + seed + "): m does not equal m.clone()");
    ensure(((AVL_TREE_SET)m).clone().equals(m), "Error (" + level + ", " + seed + "): m.clone() does not equal m");
}

int h = m.hashCode();

/* Now we save and read m. */

SORTED_SET m2 = null;

try {
    java.io.File ff = new java.io.File("it.unimi.dsi.fastutil.test");
    java.io.OutputStream os = new java.io.FileOutputStream(ff);
    java.io.ObjectOutputStream oos = new java.io.ObjectOutputStream(os);

    oos.writeObject(m);
    oos.close();

    java.io.InputStream is = new java.io.FileInputStream(ff);
    java.io.ObjectInputStream ois = new java.io.ObjectInputStream(is);

    m2 = (SORTED_SET)ois.readObject();
    ois.close();
    ff.delete();
}
catch(Exception e) {
    e.printStackTrace();
    System.exit(1);
}

ensure(m2.hashCode() == h, "Error (" + level + ", " + seed + "): hashCode() changed after save/read");

/* Now we check that m2 actually holds that data. */

ensure(m2.equals(t), "Error (" + level + ", " + seed + "): ! m2.equals(t) after save/read");
ensure(t.equals(m2), "Error (" + level + ", " + seed + "): ! t.equals(m2) after save/read");
/* Now we take out of m everything, and check that it is empty. */

for(Iterator i=t.iterator(); i.hasNext();) m2.remove(i.next());

ensure(m2.isEmpty(), "Error (" + level + ", " + seed + "): m2 is not empty (as it should be)");

/* Now we play with iterators. */

{

```

```

java.util.ListIterator i, j;
Object J;
i = (java.util.ListIterator)m.iterator();
j = new java.util.LinkedList(t).listIterator();

for(int k = 0; k < 2*n; k++) {
    ensure(i.hasNext() == j.hasNext(), "Error (" + level + ", " + seed + "): divergence in hasNext()");
    ensure(i.hasPrevious() == j.hasPrevious(), "Error (" + level + ", " + seed + "): divergence in hasPrevious()");

    if (r.nextFloat() < .8 && i.hasNext()) {
        ensure(i.next().equals(J = j.next()), "Error (" + level + ", " + seed + "): divergence in next()");

        if (r.nextFloat() < 0.5) {
            i.remove();
            j.remove();
            t.remove(J);
        }
    }
    else if (r.nextFloat() < .2 && i.hasPrevious()) {
        ensure(i.previous().equals(J = j.previous()), "Error (" + level + ", " + seed + "): divergence in previous()");

        if (r.nextFloat() < 0.5) {
            i.remove();
            j.remove();
            t.remove(J);
        }
    }

    ensure(i.nextIndex() == j.nextIndex(), "Error (" + level + ", " + seed + "): divergence in nextIndex()");
    ensure(i.previousIndex() == j.previousIndex(), "Error (" + level + ", " + seed + "): divergence in previousIndex()");

}

}

{
    boolean badPrevious = false;
    Object previous = null;
    it.unimi.dsi.fastutil.BidirectionalIterator i;
    java.util.ListIterator j;
    Object I, J;
    KEY_TYPE from = genKey();
    j = new java.util.LinkedList(t).listIterator();
    while(j.hasNext()) {
        Object k = j.next();
        if (((Comparable)k).compareTo(KEY2OBJ(from)) > 0) {
            badPrevious = true;
            j.previous();
        }
    }
}

```

```

    break;
  }
  previous = k;
}

i = (it.unimi.dsi.fastutil.BidirectionalIterator)m.iterator(from);

for(int k = 0; k < 2*n; k++) {
  ensure(i.hasNext() == j.hasNext(), "Error (" + level + ", " + seed + "): divergence in hasNext() (iterator with starting
point " + from + ")");
  ensure(i.hasPrevious() == j.hasPrevious() || badPrevious && (i.hasPrevious() == (previous != null)), "Error (" +
level + ", " + seed + "): divergence in hasPrevious() (iterator with starting point " + from + ")");

  if (r.nextFloat() < .8 && i.hasNext()) {
    ensure((I = i.next()).equals(J = j.next()), "Error (" + level + ", " + seed + "): divergence in next() (" + I + ", " + J + ",
iterator with starting point " + from + ")");
    //System.err.println("Done next " + I + " " + J + " " + badPrevious);

    badPrevious = false;

    if (r.nextFloat() < 0.5) {
      //System.err.println("Removing in next");
      i.remove();
      j.remove();
      t.remove(J);
    }
  }
  else if (!badPrevious && r.nextFloat() < .2 && i.hasPrevious()) {
    ensure((I = i.previous()).equals(J = j.previous()), "Error (" + level + ", " + seed + "): divergence in previous() (" + I +
", " + J + ", iterator with starting point " + from + ")");

    if (r.nextFloat() < 0.5) {
      //System.err.println("Removing in prev");
      i.remove();
      j.remove();
      t.remove(J);
    }
  }
}

/* Now we check that m actually holds that data. */

ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after iteration");
ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after iteration");

/* Now we select a pair of keys and create a subset. */

```

```

if (! m.isEmpty()) {
    java.util.ListIterator i;
    Object start = m.first(), end = m.first();
    for(i = (java.util.ListIterator)m.iterator(); i.hasNext() && r.nextFloat() < .3; start = end = i.next());
    for(; i.hasNext() && r.nextFloat() < .95; end = i.next());

    //System.err.println("Checking subSet from " + start + " to " + end + " (level=" + (level+1) + ")...");
    testSets((SORTED_SET)m.subSet((KEY_CLASS)start, (KEY_CLASS)end), t.subSet(start, end), n, level + 1);

    ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after subSet");
    ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after subSet");

    //System.err.println("Checking headSet to " + end + " (level=" + (level+1) + ")...");
    testSets((SORTED_SET)m.headSet((KEY_CLASS)end), t.headSet(end), n, level + 1);

    ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after headSet");
    ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after headSet");

    //System.err.println("Checking tailSet from " + start + " (level=" + (level+1) + ")...");
    testSets((SORTED_SET)m.tailSet((KEY_CLASS)start), t.tailSet(start), n, level + 1);

    ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after tailSet");
    ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after tailSet");
}
}
}

```

```

private static void runTest(int n) {
    AVL_TREE_SET m = new AVL_TREE_SET();
    SortedSet t = new java.util.TreeSet();
    topSet = m;
    k = new Object[n];
    nk = new Object[n];
    kt = new KEY_TYPE[n];
    nkt = new KEY_TYPE[n];

    for(int i = 0; i < n; i++) {
    #if KEY_CLASS_Object
        k[i] = kt[i] = genKey();
        nk[i] = nkt[i] = genKey();
    #else
        k[i] = new KEY_CLASS(kt[i] = genKey());
        nk[i] = new KEY_CLASS(nkt[i] = genKey());
    #endif
    }
}

```

```

/* We add pairs to t. */
for(int i = 0; i < n; i++) t.add(k[i]);

/* We add to m the same data */
m.addAll(t);

testSets(m, t, n, 0);

System.out.println("Test OK");
return;
}

public static void main(String args[]) {
int n = Integer.parseInt(args[1]);
if (args.length > 2) r = new java.util.Random(seed = Long.parseLong(args[2]));

try {
if ("speedTest".equals(args[0]) || "speedComp".equals(args[0])) speedTest(n, "speedComp".equals(args[0]));
else if ("test".equals(args[0])) runTest(n);
} catch(Throwable e) {
e.printStackTrace(System.err);
System.err.println("seed: " + seed);
}
}

#endif

}

```

Found in path(s):

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/drv/AVLTreeSet.drv

No license file was found, but licenses were detected in source scan.

```

/*
* Copyright (C) 2002-2017 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software

```

```
* distributed under the License is distributed on an "AS IS" BASIS,  
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
* See the License for the specific language governing permissions and  
* limitations under the License.  
*/
```

```
package PACKAGE;
```

```
import it.unimi.dsi.fastutil.Function;
```

```
/** A class providing static methods and objects that do useful things with type-specific functions.  
*  
* @see it.unimi.dsi.fastutil.Function  
* @see java.util.Collections  
*/
```

```
public final class FUNCTIONS {
```

```
    private FUNCTIONS() {}
```

```
    /** An immutable class representing an empty type-specific function.  
    *  
    * <p>This class may be useful to implement your own in case you subclass  
    * a type-specific function.  
    */
```

```
    public static class EmptyFunction KEY_VALUE_GENERIC extends ABSTRACT_FUNCTION  
    KEY_VALUE_GENERIC implements java.io.Serializable, Cloneable {
```

```
        private static final long serialVersionUID = -7046029254386353129L;
```

```
        protected EmptyFunction() {}
```

```
        @Override
```

```
        public VALUE_GENERIC_TYPE GET_VALUE(final KEY_TYPE k) { return VALUE_NULL; }
```

```
        @Override
```

```
        public boolean containsKey(final KEY_TYPE k) { return false; }
```

```
        @Override
```

```
        public VALUE_GENERIC_TYPE defaultReturnValue() { return VALUE_NULL; }
```

```
        @Override
```

```
        public void defaultReturnValue(final VALUE_GENERIC_TYPE defRetVal) { throw new  
        UnsupportedOperationException(); }
```

```
        @Override
```

```

public int size() { return 0; }

@Override
public void clear() {}

@Override
public Object clone() { return EMPTY_FUNCTION; }

@Override
public int hashCode() { return 0; }

@Override
public boolean equals(final Object o) {
    if (! (o instanceof Function)) return false;
    return ((Function<?,?>)o).size() == 0;
}

@Override
public String toString() { return "{}"; }

private Object readResolve() { return EMPTY_FUNCTION; }
}

/** An empty type-specific function (immutable). It is serializable and cloneable. */

SUPPRESS_WARNINGS_KEY_VALUE_RAWTYPES
public static final EmptyFunction EMPTY_FUNCTION = new EmptyFunction();

/** An immutable class representing a type-specific singleton function. Note
 * that the default return value is still settable.
 *
 * <p>Note that albeit the function is immutable, its default return value may be changed.
 *
 * <p>This class may be useful to implement your own in case you subclass
 * a type-specific function.
 */

public static class Singleton KEY_VALUE_GENERIC extends ABSTRACT_FUNCTION
KEY_VALUE_GENERIC implements java.io.Serializable, Cloneable {

    private static final long serialVersionUID = -7046029254386353129L;

    protected final KEY_GENERIC_TYPE key;
    protected final VALUE_GENERIC_TYPE value;

    protected Singleton(final KEY_GENERIC_TYPE key, final VALUE_GENERIC_TYPE value) {

```

```

    this.key = key;
    this.value = value;
}

@Override
public boolean containsKey(final KEY_TYPE k) { return KEY_EQUALS(key, k); }

@Override
public VALUE_GENERIC_TYPE GET_VALUE(final KEY_TYPE k) { return KEY_EQUALS(key, k) ? value :
defRetVal; }

@Override
public int size() { return 1; }

@Override
public Object clone() { return this; }
}

/** Returns a type-specific immutable function containing only the specified pair.
 * The returned function is serializable and cloneable.
 *
 * <p>Note that albeit the returned function is immutable, its default return value may be changed.
 *
 * @param key the only key of the returned function.
 * @param value the only value of the returned function.
 * @return a type-specific immutable function containing just the pair { @code &lt;key,value> }.
 */

public static KEY_VALUE_GENERIC FUNCTION KEY_VALUE_GENERIC singleton(final
KEY_GENERIC_TYPE key, VALUE_GENERIC_TYPE value) {
    return new Singleton KEY_VALUE_GENERIC_DIAMOND(key, value);
}

#if KEYS_PRIMITIVE || VALUES_PRIMITIVE

/** Returns a type-specific immutable function containing only the specified pair. The returned function is
serializable and cloneable.
 *
 * <p>Note that albeit the returned function is immutable, its default return value may be changed.
 *
 * @param key the only key of the returned function.
 * @param value the only value of the returned function.
 * @return a type-specific immutable function containing just the pair { @code &lt;key,value> }.
 */

public static KEY_VALUE_GENERIC FUNCTION KEY_VALUE_GENERIC singleton(final
KEY_GENERIC_CLASS key, final VALUE_GENERIC_CLASS value) {
    return new Singleton KEY_VALUE_GENERIC_DIAMOND(KEY_CLASS2TYPE(key),

```

```

VALUE_CLASS2TYPE(value));
}

#endif

/** A synchronized wrapper class for functions. */

public static class SynchronizedFunction KEY_VALUE_GENERIC implements FUNCTION
KEY_VALUE_GENERIC, java.io.Serializable {

    private static final long serialVersionUID = -7046029254386353129L;

    protected final FUNCTION KEY_VALUE_GENERIC function;
    protected final Object sync;

    protected SynchronizedFunction(final FUNCTION KEY_VALUE_GENERIC f, final Object sync) {
        if (f == null) throw new NullPointerException();
        this.function = f;
        this.sync = sync;
    }

    protected SynchronizedFunction(final FUNCTION KEY_VALUE_GENERIC f) {
        if (f == null) throw new NullPointerException();
        this.function = f;
        this.sync = this;
    }

#ifdef JDK_PRIMITIVE_FUNCTION

#ifdef KEY_WIDENED
    /** {@inheritDoc}
     * @deprecated Please use the corresponding type-specific method instead. */
    @Deprecated
#endif
    @Override
    public VALUE_GENERIC_TYPE_WIDENED
    JDK_PRIMITIVE_FUNCTION_APPLY(KEY_GENERIC_TYPE_WIDENED operand) { synchronized(sync) {
    return function.JDK_PRIMITIVE_FUNCTION_APPLY(operand); } }

#endif

#ifdef KEYS_PRIMITIVE || VALUES_PRIMITIVE
    /** {@inheritDoc}
     * @deprecated Please use the corresponding type-specific method instead. */
    @Deprecated
#endif
    @Override

```

```

public VALUE_GENERIC_CLASS apply(final KEY_GENERIC_CLASS key) { synchronized (sync) { return
function.apply(key); } }

@Override
public int size() { synchronized(sync) { return function.size(); } }

@Override
public VALUE_GENERIC_TYPE defaultReturnValue() { synchronized(sync) { return
function.defaultReturnValue(); } }

@Override
public void defaultReturnValue(final VALUE_GENERIC_TYPE defRetVal) { synchronized(sync) {
function.defaultReturnValue(defRetVal); } }

@Override
public boolean containsKey(final KEY_TYPE k) { synchronized(sync) { return function.containsKey(k); } }

#if KEYS_PRIMITIVE

@Deprecated
@Override
public boolean containsKey(final Object k) { synchronized(sync) { return function.containsKey(k); } }

#endif

@Override
public VALUE_GENERIC_TYPE put(final KEY_GENERIC_TYPE k, final VALUE_GENERIC_TYPE v) {
synchronized(sync) { return function.put(k, v); } }

@Override
public VALUE_GENERIC_TYPE GET_VALUE(final KEY_TYPE k) { synchronized(sync) { return
function.GET_VALUE(k); } }

@Override
public VALUE_GENERIC_TYPE REMOVE_VALUE(final KEY_TYPE k) { synchronized(sync) { return
function.REMOVE_VALUE(k); } }

@Override
public void clear() { synchronized(sync) { function.clear(); } }

#if KEYS_PRIMITIVE || VALUES_PRIMITIVE

/** { @inheritDoc }
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public VALUE_GENERIC_CLASS put(final KEY_GENERIC_CLASS k, final VALUE_GENERIC_CLASS v) {
synchronized(sync) { return function.put(k, v); } }

```

```

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public VALUE_GENERIC_CLASS get(final Object k) { synchronized(sync) { return function.get(k); } }

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public VALUE_GENERIC_CLASS remove(final Object k) { synchronized(sync) { return function.remove(k); } }

#endif

@Override
public int hashCode() { synchronized(sync) { return function.hashCode(); } }

@Override
public boolean equals(final Object o) {
    if (o == this) return true;
    synchronized(sync) { return function.equals(o); }
}

@Override
public String toString() { synchronized(sync) { return function.toString(); } }

private void writeObject(java.io.ObjectOutputStream s) throws java.io.IOException {
    synchronized(sync) { s.defaultWriteObject(); }
}
}

/** Returns a synchronized type-specific function backed by the given type-specific function.
 *
 * @param f the function to be wrapped in a synchronized function.
 * @return a synchronized view of the specified function.
 * @see java.util.Collections#synchronizedMap(java.util.Map)
 */
public static KEY_VALUE_GENERIC FUNCTION KEY_VALUE_GENERIC synchronize(final FUNCTION
KEY_VALUE_GENERIC f) { return new SynchronizedFunction KEY_VALUE_GENERIC_DIAMOND(f); }

/** Returns a synchronized type-specific function backed by the given type-specific function, using an assigned
object to synchronize.
 *
 * @param f the function to be wrapped in a synchronized function.
 * @param sync an object that will be used to synchronize the access to the function.
 * @return a synchronized view of the specified function.
 * @see java.util.Collections#synchronizedMap(java.util.Map)

```

```

*/

public static KEY_VALUE_GENERIC FUNCTION KEY_VALUE_GENERIC synchronize(final FUNCTION
KEY_VALUE_GENERIC f, final Object sync) { return new SynchronizedFunction
KEY_VALUE_GENERIC_DIAMOND(f, sync); }

/** An unmodifiable wrapper class for functions. */

public static class UnmodifiableFunction KEY_VALUE_GENERIC extends ABSTRACT_FUNCTION
KEY_VALUE_GENERIC implements java.io.Serializable {

private static final long serialVersionUID = -7046029254386353129L;

protected final FUNCTION KEY_VALUE_GENERIC function;

protected UnmodifiableFunction(final FUNCTION KEY_VALUE_GENERIC f) {
if (f == null) throw new NullPointerException();
this.function = f;
}

@Override
public int size() { return function.size(); }

@Override
public VALUE_GENERIC_TYPE defaultReturnValue() { return function.defaultReturnValue(); }

@Override
public void defaultReturnValue(final VALUE_GENERIC_TYPE defRetVal) { throw new
UnsupportedOperationException(); }

@Override
public boolean containsKey(final KEY_TYPE k) { return function.containsKey(k); }

@Override
public VALUE_GENERIC_TYPE put(final KEY_GENERIC_TYPE k, final VALUE_GENERIC_TYPE v) { throw
new UnsupportedOperationException(); }

@Override
public VALUE_GENERIC_TYPE GET_VALUE(final KEY_TYPE k) { return function.GET_VALUE(k); }

@Override
public VALUE_GENERIC_TYPE REMOVE_VALUE(final KEY_TYPE k) { throw new
UnsupportedOperationException(); }

@Override
public void clear() { throw new UnsupportedOperationException(); }

```

```

#if KEYS_PRIMITIVE || VALUES_PRIMITIVE
/** { @inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public VALUE_GENERIC_CLASS put(final KEY_GENERIC_CLASS k, final VALUE_GENERIC_CLASS v) {
throw new UnsupportedOperationException(); }

/** { @inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public VALUE_GENERIC_CLASS get(final Object k) { return function.get(k); }

/** { @inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public VALUE_GENERIC_CLASS remove(final Object k) { throw new UnsupportedOperationException(); }
#endif

@Override
public int hashCode() { return function.hashCode(); }

@Override
public boolean equals(Object o) { return o == this || function.equals(o); }

@Override
public String toString() { return function.toString(); }
}

/** Returns an unmodifiable type-specific function backed by the given type-specific function.
 *
 * @param f the function to be wrapped in an unmodifiable function.
 * @return an unmodifiable view of the specified function.
 * @see java.util.Collections#unmodifiableMap(java.util.Map)
 */
public static KEY_VALUE_GENERIC FUNCTION KEY_VALUE_GENERIC unmodifiable(final FUNCTION
KEY_VALUE_GENERIC f) { return new UnmodifiableFunction KEY_VALUE_GENERIC_DIAMOND(f); }

#if KEYS_PRIMITIVE || VALUES_PRIMITIVE
/** An adapter for mapping generic total functions to partial primitive functions. */

public static class PrimitiveFunction KEY_VALUE_GENERIC implements FUNCTION
KEY_VALUE_GENERIC {
protected final java.util.function.Function<? super KEY_GENERIC_CLASS, ? extends

```

```

VALUE_GENERIC_CLASS> function;

    protected PrimitiveFunction(java.util.function.Function<? super KEY_GENERIC_CLASS, ? extends
VALUE_GENERIC_CLASS> function) {
        this.function = function;
    }

#if KEYS_PRIMITIVE
    @Override
    public boolean containsKey(KEY_GENERIC_TYPE key) { return function.apply(KEY2OBJ(key)) != null; }
#endif

    SUPPRESS_WARNINGS_KEY_UNCHECKED
#if KEYS_PRIMITIVE
    @Deprecated
#endif
    @Override
    public boolean containsKey(Object key) {
#if KEYS_PRIMITIVE
        if (key == null) return false;
#endif
        return function.apply(KEY_CLASS_CAST (key)) != null;
    }

    SUPPRESS_WARNINGS_KEY_UNCHECKED
    @Override
    public VALUE_GENERIC_TYPE GET_VALUE(KEY_TYPE key) {
        VALUE_GENERIC_CLASS v = function.apply(KEY_GENERIC_CAST KEY2OBJ(key));
#if VALUES_PRIMITIVE
        if (v == null) return defaultReturnValue();
#else
        if (v == null) return null;
#endif
        return VALUE_CLASS2TYPE(v);
    }

    SUPPRESS_WARNINGS_KEY_UNCHECKED
    @Deprecated
    @Override
    public VALUE_GENERIC_CLASS get(Object key) {
#if KEYS_PRIMITIVE
        if (key == null) return null;
#endif
        return function.apply(KEY_CLASS_CAST key);
    }

    @Deprecated
    @Override

```

```

public VALUE_GENERIC_CLASS put(final KEY_GENERIC_CLASS key, final VALUE_GENERIC_CLASS
value) { throw new UnsupportedOperationException(); }
}

```

```

/** Returns a (partial) type-specific function based on the given total generic function.
 * <p>The returned function contains all keys which are not mapped to { @code null}. If the function already
 * is a primitive function, it is returned without changes.
 * <p><strong>Warning</strong>: If the given function is a &ldquo;widened&rdquo; primitive function (e.g. an
 * { @code Int2IntFunction} given to { @code Short2ShortFunctions}), it still is wrapped into a proxy,
 * decreasing performance.
 *
 * @param f the function to be converted to a type-specific function.
 * @return a primitive view of the specified function.
 * @throws NullPointerException if { @code f} is null.
 * @see PrimitiveFunction
 * @since 8.1.0
 */

```

```

SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
public static KEY_VALUE_GENERIC FUNCTION KEY_VALUE_GENERIC primitive(final
java.util.function.Function<? super KEY_GENERIC_CLASS, ? extends VALUE_GENERIC_CLASS> f) {
    java.util.Objects.requireNonNull(f);

    if (f instanceof FUNCTION) return (FUNCTION KEY_VALUE_GENERIC) f;
#if defined JDK_PRIMITIVE_FUNCTION
    if (f instanceof JDK_PRIMITIVE_FUNCTION)
#if KEYS_PRIMITIVE
#if VALUE_WIDENED
        return key -> VALUE_NARROWING(((JDK_PRIMITIVE_FUNCTION KEY_VALUE_GENERIC)
f).JDK_PRIMITIVE_FUNCTION_APPLY(key));
#else
        return ((JDK_PRIMITIVE_FUNCTION KEY_VALUE_GENERIC) f)::JDK_PRIMITIVE_FUNCTION_APPLY;
#endif
#else
        return key -> VALUE_NARROWING(((JDK_PRIMITIVE_FUNCTION KEY_VALUE_GENERIC)
f).JDK_PRIMITIVE_FUNCTION_APPLY((K)(key)));
#endif
#endif
    return new PrimitiveFunction KEY_VALUE_GENERIC_DIAMOND(f);
}
#endif
}

```

Found in path(s):

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-
a775574/drv/Functions.drv

No license file was found, but licenses were detected in source scan.

/*

```
* Copyright (C) 2002-2017 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

```
package PACKAGE;
```

```
/** An abstract class facilitating the creation of type-specific { @linkplain java.util.ListIterator list iterators }.
*
* @deprecated As of fastutil 8 this class is no longer necessary, as its previous abstract
* methods are now default methods of the type-specific interface.
*/
```

```
@Deprecated
```

```
public abstract class KEY_ABSTRACT_LIST_ITERATOR KEY_GENERIC extends
KEY_ABSTRACT_BIDI_ITERATOR KEY_GENERIC implements KEY_LIST_ITERATOR KEY_GENERIC {
    protected KEY_ABSTRACT_LIST_ITERATOR() {}
}
```

```
Found in path(s):
```

```
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-
a775574/drv/AbstractListIterator.drv
```

```
No license file was found, but licenses were detected in source scan.
```

```
/*
* Copyright (C) 2007-2017 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
```

* limitations under the License.

*/

package PACKAGE;

import java.util.Collection;

import java.util.NoSuchElementException;

/** A simple, brute-force implementation of a set based on a backing array.

*

* <p>The main purpose of this

* implementation is that of wrapping cleanly the brute-force approach to the storage of a very

* small number of items: just put them into an array and scan linearly to find an item.

*/

public class ARRAY_SET KEY_GENERIC extends ABSTRACT_SET KEY_GENERIC implements
java.io.Serializable, Cloneable {

private static final long serialVersionUID = 1L;

/** The backing array (valid up to { @link #size}, excluded). */

private transient KEY_TYPE[] a;

/** The number of valid entries in { @link #a}. */

private int size;

/** Creates a new array set using the given backing array. The resulting set will have as many elements as the array.

*

* <p>It is responsibility of the caller that the elements of { @code a} are distinct.

*

* @param a the backing array.

*/

public ARRAY_SET(final KEY_TYPE[] a) {

 this.a = a;

 size = a.length;

}

/** Creates a new empty array set.

*/

public ARRAY_SET() { this.a = ARRAYS.EMPTY_ARRAY; }

/** Creates a new empty array set of given initial capacity.

*

* @param capacity the initial capacity.

*/

public ARRAY_SET(final int capacity) { this.a = new KEY_TYPE[capacity]; }

/** Creates a new array set copying the contents of a given collection.

* @param c a collection.

```

*/
public ARRAY_SET(COLLECTION KEY_GENERIC c) {
    this(c.size ());
    addAll(c);
}

/** Creates a new array set copying the contents of a given set.
 * @param c a collection.
 */
public ARRAY_SET(final Collection<? extends KEY_GENERIC_CLASS> c) {
    this(c.size());
    addAll(c);
}

/** Creates a new array set using the given backing array and the given number of elements of the array.
 *
 * <p>It is responsibility of the caller that the first { @code size} elements of { @code a} are distinct.
 *
 * @param a the backing array.
 * @param size the number of valid elements in { @code a}.
 */
public ARRAY_SET(final KEY_TYPE[] a, final int size) {
    this.a = a;
    this.size = size;
    if (size > a.length) throw new IllegalArgumentException("The provided size (" + size + ") is larger than or equal to
the array size (" + a.length + ")");
}

private int findKey(final KEY_TYPE o) {
    for(int i = size; i-- != 0;) if (KEY_EQUALS(a[i], o)) return i;
    return -1;
}

@Override
SUPPRESS_WARNINGS_KEY_UNCHECKED
public KEY_ITERATOR KEY_GENERIC iterator() {
    return new KEY_ITERATOR KEY_GENERIC () {
        int next = 0;

        @Override
        public boolean hasNext() { return next < size; }

        @Override
        public KEY_GENERIC_TYPE NEXT_KEY() {
            if (! hasNext()) throw new NoSuchElementException();
            return KEY_GENERIC_CAST a[next++];
        }
    }
}

```

```

@Override
public void remove() {
    final int tail = size-- - next--;
    System.arraycopy(a, next + 1, a, next, tail);
#if KEYS_REFERENCE
    a[size] = null;
#endif
}
};
}

@Override
public boolean contains(final KEY_TYPE k) { return findKey(k) != -1; }

@Override
public int size() { return size; }

@Override
public boolean remove(final KEY_TYPE k) {
    final int pos = findKey(k);
    if (pos == -1) return false;
    final int tail = size - pos - 1;
    for(int i = 0; i < tail; i++) a[pos + i] = a[pos + i + 1];
    size--;
#if KEYS_REFERENCE
    a[size] = null;
#endif
    return true;
}

@Override
public boolean add(final KEY_GENERIC_TYPE k) {
    final int pos = findKey(k);
    if (pos != -1) return false;
    if (size == a.length) {
        final KEY_TYPE[] b = new KEY_TYPE[size == 0 ? 2 : size * 2];
        for(int i = size; i-- != 0;) b[i] = a[i];
        a = b;
    }
    a[size++] = k;
    return true;
}

@Override
public void clear() {
#if KEYS_REFERENCE
    java.util.Arrays.fill(a, 0, size, null);
#endif
}

```

```

#endif
    size = 0;
}

@Override
public boolean isEmpty() { return size == 0; }

/** Returns a deep copy of this set.
 *
 * <p>This method performs a deep copy of this array set; the data stored in the
 * set, however, is not cloned. Note that this makes a difference only for object keys.
 *
 * @return a deep copy of this set.
 */
@Override
SUPPRESS_WARNINGS_KEY_UNCHECKED
public ARRAY_SET KEY_GENERIC clone() {
    ARRAY_SET KEY_GENERIC c;
    try {
        c = (ARRAY_SET KEY_GENERIC)super.clone();
    }
    catch(CloneNotSupportedException cantHappen) {
        throw new InternalError();
    }
    c.a = a.clone();
    return c;
}

private void writeObject(java.io.ObjectOutputStream s) throws java.io.IOException {
    s.defaultWriteObject();
    for(int i = 0; i < size; i++) s.WRITE_KEY(a[i]);
}

private void readObject(java.io.ObjectInputStream s) throws java.io.IOException, ClassNotFoundException {
    s.defaultReadObject();
    a = new KEY_TYPE[size];
    for(int i = 0; i < size; i++) a[i] = s.READ_KEY();
}
}

```

Found in path(s):

```

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-
a775574/drv/ArraySet.drv

```

No license file was found, but licenses were detected in source scan.

```

/*
 * Copyright (C) 2002-2017 Sebastiano Vigna
 *

```

* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

```
package PACKAGE;
```

```
import it.unimi.dsi.fastutil.Hash;  
import it.unimi.dsi.fastutil.HashCommon;  
import static it.unimi.dsi.fastutil.HashCommon.arraySize;  
import static it.unimi.dsi.fastutil.HashCommon.maxFill;
```

```
import java.util.Map;  
import java.util.Arrays;  
import java.util.NoSuchElementException;  
import java.util.function.Consumer;
```

```
import VALUE_PACKAGE.VALUE_COLLECTION;  
import VALUE_PACKAGE.VALUE_ABSTRACT_COLLECTION;
```

```
#if VALUES_PRIMITIVE  
import VALUE_PACKAGE.VALUE_ITERATOR;  
#endif
```

```
#if VALUE_CLASS_Boolean  
import it.unimi.dsi.fastutil.booleans.BooleanConsumer;  
#endif
```

```
#ifdef Linked
```

```
import java.util.Comparator;
```

```
#if VALUES_PRIMITIVE  
import VALUE_PACKAGE.VALUE_LIST_ITERATOR;  
#else  
import it.unimi.dsi.fastutil.objects.ObjectIterator;  
#endif
```

```

import it.unimi.dsi.fastutil.objects.AbstractObjectSortedSet;
import it.unimi.dsi.fastutil.objects.ObjectListIterator;
import it.unimi.dsi.fastutil.objects.ObjectBidirectionalIterator;
import it.unimi.dsi.fastutil.objects.ObjectSortedSet;

/** A type-specific linked hash map with with a fast, small-footprint implementation.
 *
 * <p>Instances of this class use a hash table to represent a map. The table is
 * filled up to a specified <em>load factor</em>, and then doubled in size to
 * accommodate new entries. If the table is emptied below <em>one fourth</em>
 * of the load factor, it is halved in size; however, the table is never reduced to a
 * size smaller than that at creation time: this approach makes it
 * possible to create maps with a large capacity in which insertions and
 * deletions do not cause immediately rehashing. Moreover, halving is
 * not performed when deleting entries from an iterator, as it would interfere
 * with the iteration process.
 *
 * <p>Note that { @link #clear() } does not modify the hash table size.
 * Rather, a family of { @linkplain #trim() trimming
 * methods } lets you control the size of the table; this is particularly useful
 * if you reuse instances of this class.
 *
 * <p>Iterators generated by this map will enumerate pairs in the same order in which they
 * have been added to the map (addition of pairs whose key is already present
 * in the map does not change the iteration order). Note that this order has nothing in common with the natural
 * order of the keys. The order is kept by means of a doubly linked list, represented
 * <i>via</i> an array of longs parallel to the table.
 *
 * <p>This class implements the interface of a sorted map, so to allow easy
 * access of the iteration order: for instance, you can get the first key
 * in iteration order with { @code firstKey() } without having to create an
 * iterator; however, this class partially violates the { @link java.util.SortedMap }
 * contract because all submap methods throw an exception and { @link
 * #comparator() } returns always { @code null }.
 *
 * <p>Additional methods, such as { @code getAndMoveToFirst() }, make it easy
 * to use instances of this class as a cache (e.g., with LRU policy).
 *
 * <p>The iterators provided by the views of this class using are type-specific
 * { @linkplain java.util.ListIterator list iterators }, and can be started at any
 * element <em>which is a key of the map</em>, or
 * a { @link NoSuchElementException } exception will be thrown.
 * If, however, the provided element is not the first or last key in the
 * map, the first access to the list index will require linear time, as in the worst case
 * the entire key set must be scanned in iteration order to retrieve the positional
 * index of the starting key. If you use just the methods of a type-specific { @link
 * it.unimi.dsi.fastutil.BidirectionalIterator },
 * however, all operations will be performed in constant time.

```

```
*
* @see Hash
* @see HashCommon
*/
```

```
public class OPEN_HASH_MAP_KEY_VALUE_GENERIC extends ABSTRACT_SORTED_MAP
KEY_VALUE_GENERIC implements java.io.Serializable, Cloneable, Hash {
```

```
#else
```

```
import it.unimi.dsi.fastutil.objects.AbstractObjectSet;
import it.unimi.dsi.fastutil.objects.ObjectIterator;
```

```
#ifdef Custom
```

```
/** A type-specific hash map with a fast, small-footprint implementation whose { @linkplain
it.unimi.dsi.fastutil.Hash.Strategy hashing strategy }
```

```
* is specified at creation time.
```

```
*
```

```
* <p>Instances of this class use a hash table to represent a map. The table is
* filled up to a specified <em>load factor</em>, and then doubled in size to
* accommodate new entries. If the table is emptied below <em>one fourth</em>
* of the load factor, it is halved in size; however, the table is never reduced to a
* size smaller than that at creation time: this approach makes it
* possible to create maps with a large capacity in which insertions and
* deletions do not cause immediately rehashing. Moreover, halving is
* not performed when deleting entries from an iterator, as it would interfere
* with the iteration process.
```

```
*
```

```
* <p>Note that { @link #clear() } does not modify the hash table size.
```

```
* Rather, a family of { @linkplain #trim() trimming
* methods } lets you control the size of the table; this is particularly useful
* if you reuse instances of this class.
```

```
*
```

```
* @see Hash
```

```
* @see HashCommon
```

```
*/
```

```
public class OPEN_HASH_MAP_KEY_VALUE_GENERIC extends ABSTRACT_MAP
KEY_VALUE_GENERIC implements java.io.Serializable, Cloneable, Hash {
```

```
#else
```

```
/** A type-specific hash map with a fast, small-footprint implementation.
```

```
*
```

```
* <p>Instances of this class use a hash table to represent a map. The table is
* filled up to a specified <em>load factor</em>, and then doubled in size to
* accommodate new entries. If the table is emptied below <em>one fourth</em>
```

- * of the load factor, it is halved in size; however, the table is never reduced to a
- * size smaller than that at creation time: this approach makes it
- * possible to create maps with a large capacity in which insertions and
- * deletions do not cause immediately rehashing. Moreover, halving is
- * not performed when deleting entries from an iterator, as it would interfere
- * with the iteration process.
- *
- * <p>Note that { @link #clear() } does not modify the hash table size.
- * Rather, a family of { @linkplain #trim() trimming
- * methods } lets you control the size of the table; this is particularly useful
- * if you reuse instances of this class.
- *
- * @see Hash
- * @see HashCommon
- */

```
public class OPEN_HASH_MAP KEY_VALUE_GENERIC extends ABSTRACT_MAP
KEY_VALUE_GENERIC implements java.io.Serializable, Cloneable, Hash {
```

```
#endif
```

```
#endif
```

```
private static final long serialVersionUID = 0L;
private static final boolean ASSERTS = ASSERTS_VALUE;
```

```
/** The array of keys. */
protected transient KEY_GENERIC_TYPE[] key;
```

```
/** The array of values. */
protected transient VALUE_GENERIC_TYPE[] value;
```

```
/** The mask for wrapping a position counter. */
protected transient int mask;
```

```
/** Whether this map contains the key zero. */
protected transient boolean containsNullKey;
```

```
#ifdef Custom
```

```
/** The hash strategy of this custom map. */
protected STRATEGY KEY_SUPER_GENERIC strategy;
```

```
#endif
```

```
#ifdef Linked
```

```
/** The index of the first entry in iteration order. It is valid iff { @link #size } is nonzero; otherwise, it contains -1. */
protected transient int first = -1;
```

```
/** The index of the last entry in iteration order. It is valid iff { @link #size } is nonzero; otherwise, it contains -1. */
protected transient int last = -1;
```

```

/** For each entry, the next and the previous entry in iteration order,
 * stored as { @code ((prev & 0xFFFFFFFFFL) << 32) | (next & 0xFFFFFFFFFL)}.
 * The first entry contains predecessor -1, and the last entry
 * contains successor -1. */
protected transient long[] link;
#endif

/** The current table size. */
protected transient int n;

/** Threshold after which we rehash. It must be the table size times { @link #f}. */
protected transient int maxFill;

/** We never resize below this threshold, which is the construction-time {#n}. */
protected final transient int minN;

/** Number of entries in the set (including the key zero, if present). */
protected int size;

/** The acceptable load factor. */
protected final float f;

#ifdef Linked
/** Cached set of entries. */
protected transient FastSortedEntrySet KEY_VALUE_GENERIC entries;

/** Cached set of keys. */
protected transient SORTED_SET KEY_GENERIC keys;
#else
/** Cached set of entries. */
protected transient FastEntrySet KEY_VALUE_GENERIC entries;

/** Cached set of keys. */
protected transient SET KEY_GENERIC keys;
#endif

/** Cached collection of values. */
protected transient VALUE_COLLECTION VALUE_GENERIC values;

#ifdef Custom
/** Creates a new hash map.
 *
 * <p>The actual table size will be the least power of two greater than { @code expected}/{ @code f}.
 *
 * @param expected the expected number of elements in the hash map.
 * @param f the load factor.
 * @param strategy the strategy.

```

```

*/
SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
public OPEN_HASH_MAP(final int expected, final float f, final STRATEGY KEY_SUPER_GENERIC strategy) {
    this.strategy = strategy;
#else
/** Creates a new hash map.
 *
 * <p>The actual table size will be the least power of two greater than { @code expected}/{ @code f}.
 *
 * @param expected the expected number of elements in the hash map.
 * @param f the load factor.
 */
SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
public OPEN_HASH_MAP(final int expected, final float f) {
#endif
    if (f <= 0 || f > 1) throw new IllegalArgumentException("Load factor must be greater than 0 and smaller than or
equal to 1");
    if (expected < 0) throw new IllegalArgumentException("The expected number of elements must be nonnegative");

    this.f = f;

    minN = n = arraySize(expected, f);
    mask = n - 1;
    maxFill = maxFill(n, f);
    key = KEY_GENERIC_ARRAY_CAST new KEY_TYPE[n + 1];
    value = VALUE_GENERIC_ARRAY_CAST new VALUE_TYPE[n + 1];
#ifdef Linked
    link = new long[n + 1];
#endif
}

#ifdef Custom
/** Creates a new hash map with { @link Hash#DEFAULT_LOAD_FACTOR} as load factor.
 *
 * @param expected the expected number of elements in the hash map.
 * @param strategy the strategy.
 */

public OPEN_HASH_MAP(final int expected, final STRATEGY KEY_SUPER_GENERIC strategy) {
    this(expected, DEFAULT_LOAD_FACTOR, strategy);
}
#else
/** Creates a new hash map with { @link Hash#DEFAULT_LOAD_FACTOR} as load factor.
 *
 * @param expected the expected number of elements in the hash map.
 */

```

```

public OPEN_HASH_MAP(final int expected) {
    this(expected, DEFAULT_LOAD_FACTOR);
}
#endif

#ifdef Custom
/** Creates a new hash map with initial expected { @link Hash#DEFAULT_INITIAL_SIZE } entries
 * and { @link Hash#DEFAULT_LOAD_FACTOR } as load factor.
 * @param strategy the strategy.
 */

public OPEN_HASH_MAP(final STRATEGY KEY_SUPER_GENERIC strategy) {
    this(DEFAULT_INITIAL_SIZE, DEFAULT_LOAD_FACTOR, strategy);
}
#else
/** Creates a new hash map with initial expected { @link Hash#DEFAULT_INITIAL_SIZE } entries
 * and { @link Hash#DEFAULT_LOAD_FACTOR } as load factor.
 */

public OPEN_HASH_MAP() {
    this(DEFAULT_INITIAL_SIZE, DEFAULT_LOAD_FACTOR);
}
#endif

#ifdef Custom
/** Creates a new hash map copying a given one.
 *
 * @param m a { @link Map } to be copied into the new hash map.
 * @param f the load factor.
 * @param strategy the strategy.
 */

public OPEN_HASH_MAP(final Map<? extends KEY_GENERIC_CLASS, ? extends
VALUE_GENERIC_CLASS> m, final float f, final STRATEGY KEY_SUPER_GENERIC strategy) {
    this(m.size(), f, strategy);
    putAll(m);
}
#else
/** Creates a new hash map copying a given one.
 *
 * @param m a { @link Map } to be copied into the new hash map.
 * @param f the load factor.
 */

public OPEN_HASH_MAP(final Map<? extends KEY_GENERIC_CLASS, ? extends
VALUE_GENERIC_CLASS> m, final float f) {

```

```

    this(m.size(), f);
    putAll(m);
}
#endif

#ifdef Custom
/** Creates a new hash map with {@link Hash#DEFAULT_LOAD_FACTOR} as load factor copying a given one.
 *
 * @param m a {@link Map} to be copied into the new hash map.
 * @param strategy the strategy.
 */

public OPEN_HASH_MAP(final Map<? extends KEY_GENERIC_CLASS, ? extends
VALUE_GENERIC_CLASS> m, final STRATEGY KEY_SUPER_GENERIC strategy) {
    this(m, DEFAULT_LOAD_FACTOR, strategy);
}
#else
/** Creates a new hash map with {@link Hash#DEFAULT_LOAD_FACTOR} as load factor copying a given one.
 *
 * @param m a {@link Map} to be copied into the new hash map.
 */

public OPEN_HASH_MAP(final Map<? extends KEY_GENERIC_CLASS, ? extends
VALUE_GENERIC_CLASS> m) {
    this(m, DEFAULT_LOAD_FACTOR);
}
#endif

#ifdef Custom
/** Creates a new hash map copying a given type-specific one.
 *
 * @param m a type-specific map to be copied into the new hash map.
 * @param f the load factor.
 * @param strategy the strategy.
 */

public OPEN_HASH_MAP(final MAP KEY_VALUE_GENERIC m, final float f, final STRATEGY
KEY_SUPER_GENERIC strategy) {
    this(m.size(), f, strategy);
    putAll(m);
}
#else
/** Creates a new hash map copying a given type-specific one.
 *
 * @param m a type-specific map to be copied into the new hash map.
 * @param f the load factor.
 */

```

```

public OPEN_HASH_MAP(final MAP KEY_VALUE_GENERIC m, final float f) {
    this(m.size(), f);
    putAll(m);
}
#endif

#ifdef Custom
/** Creates a new hash map with {@link Hash#DEFAULT_LOAD_FACTOR} as load factor copying a given type-
specific one.
*
* @param m a type-specific map to be copied into the new hash map.
* @param strategy the strategy.
*/

public OPEN_HASH_MAP(final MAP KEY_VALUE_GENERIC m, final STRATEGY KEY_SUPER_GENERIC
strategy) {
    this(m, DEFAULT_LOAD_FACTOR, strategy);
}
#else
/** Creates a new hash map with {@link Hash#DEFAULT_LOAD_FACTOR} as load factor copying a given type-
specific one.
*
* @param m a type-specific map to be copied into the new hash map.
*/

public OPEN_HASH_MAP(final MAP KEY_VALUE_GENERIC m) {
    this(m, DEFAULT_LOAD_FACTOR);
}
#endif

#ifdef Custom
/** Creates a new hash map using the elements of two parallel arrays.
*
* @param k the array of keys of the new hash map.
* @param v the array of corresponding values in the new hash map.
* @param f the load factor.
* @param strategy the strategy.
* @throws IllegalArgumentException if {@code k} and {@code v} have different lengths.
*/

public OPEN_HASH_MAP(final KEY_GENERIC_TYPE[] k, final VALUE_GENERIC_TYPE[] v, final float f,
final STRATEGY KEY_SUPER_GENERIC strategy) {
    this(k.length, f, strategy);
    if (k.length != v.length) throw new IllegalArgumentException("The key array and the value array have different
lengths (" + k.length + " and " + v.length + ")");
    for(int i = 0; i < k.length; i++) this.put(k[i], v[i]);
}

```

```

#else
/** Creates a new hash map using the elements of two parallel arrays.
 *
 * @param k the array of keys of the new hash map.
 * @param v the array of corresponding values in the new hash map.
 * @param f the load factor.
 * @throws IllegalArgumentException if {@code k} and {@code v} have different lengths.
 */

public OPEN_HASH_MAP(final KEY_GENERIC_TYPE[] k, final VALUE_GENERIC_TYPE[] v, final float f) {
    this(k.length, f);
    if (k.length != v.length) throw new IllegalArgumentException("The key array and the value array have different
lengths (" + k.length + " and " + v.length + ")");
    for(int i = 0; i < k.length; i++) this.put(k[i], v[i]);
}
#endif

#ifdef Custom
/** Creates a new hash map with {@link Hash#DEFAULT_LOAD_FACTOR} as load factor using the elements of
two parallel arrays.
 *
 * @param k the array of keys of the new hash map.
 * @param v the array of corresponding values in the new hash map.
 * @param strategy the strategy.
 * @throws IllegalArgumentException if {@code k} and {@code v} have different lengths.
 */

public OPEN_HASH_MAP(final KEY_GENERIC_TYPE[] k, final VALUE_GENERIC_TYPE[] v, final
STRATEGY KEY_SUPER_GENERIC strategy) {
    this(k, v, DEFAULT_LOAD_FACTOR, strategy);
}
#else
/** Creates a new hash map with {@link Hash#DEFAULT_LOAD_FACTOR} as load factor using the elements of
two parallel arrays.
 *
 * @param k the array of keys of the new hash map.
 * @param v the array of corresponding values in the new hash map.
 * @throws IllegalArgumentException if {@code k} and {@code v} have different lengths.
 */

public OPEN_HASH_MAP(final KEY_GENERIC_TYPE[] k, final VALUE_GENERIC_TYPE[] v) {
    this(k, v, DEFAULT_LOAD_FACTOR);
}
#endif

#ifdef Custom

```

```

/** Returns the hashing strategy.
 *
 * @return the hashing strategy of this custom hash map.
 */

public STRATEGY KEY_SUPER_GENERIC strategy() {
    return strategy;
}
#endif

private int realSize() {
    return containsNullKey ? size - 1 : size;
}

private void ensureCapacity(final int capacity) {
    final int needed = arraySize(capacity, f);
    if (needed > n) rehash(needed);
}

private void tryCapacity(final long capacity) {
    final int needed = (int)Math.min(1 << 30, Math.max(2, HashCommon.nextPowerOfTwo((long)Math.ceil(capacity /
f))));
    if (needed > n) rehash(needed);
}

private VALUE_GENERIC_TYPE removeEntry(final int pos) {
    final VALUE_GENERIC_TYPE oldValue = value[pos];
#ifdef VALUES_REFERENCE
    value[pos] = null;
#endif
    size--;
#ifdef Linked
    fixPointers(pos);
#endif

    shiftKeys(pos);
    if (n > minN && size < maxFill / 4 && n > DEFAULT_INITIAL_SIZE) rehash(n / 2);
    return oldValue;
}

private VALUE_GENERIC_TYPE removeNullEntry() {
    containsNullKey = false;
#ifdef KEYS_REFERENCE
    key[n] = null;
#endif
    final VALUE_GENERIC_TYPE oldValue = value[n];
#ifdef VALUES_REFERENCE
    value[n] = null;

```

```

#endif
    size--;
#ifdef Linked
    fixPointers(n);
#endif
    if (n > minN && size < maxFill / 4 && n > DEFAULT_INITIAL_SIZE) rehash(n / 2);
    return oldValue;
}

@Override
public void putAll(Map<? extends KEY_GENERIC_CLASS,? extends VALUE_GENERIC_CLASS> m) {
    if (f <= .5) ensureCapacity(m.size()); // The resulting map will be sized for m.size() elements
    else tryCapacity(size() + m.size()); // The resulting map will be tentatively sized for size() + m.size() elements
    super.putAll(m);
}

SUPPRESS_WARNINGS_KEY_UNCHECKED
private int find(final KEY_GENERIC_TYPE k) {
    if (KEY_EQUALS_NULL(k)) return containsNullKey ? n : -(n + 1);

    KEY_GENERIC_TYPE curr;
    final KEY_GENERIC_TYPE[] key = this.key;
    int pos;

    // The starting point.
    if (KEY_IS_NULL(curr = key[pos = KEY2INTHASH(k) & mask])) return -(pos + 1);
    if (KEY_EQUALS_NOT_NULL(k, curr)) return pos;
    // There's always an unused entry.
    while(true) {
        if (KEY_IS_NULL(curr = key[pos = (pos + 1) & mask])) return -(pos + 1);
        if (KEY_EQUALS_NOT_NULL(k, curr)) return pos;
    }
}

private void insert(final int pos, final KEY_GENERIC_TYPE k, final VALUE_GENERIC_TYPE v) {
    if (pos == n) containsNullKey = true;
    key[pos] = k;
    value[pos] = v;
#ifdef Linked
    if (size == 0) {
        first = last = pos;
        // Special case of SET_UPPER_LOWER(link[pos], -1, -1);
        link[pos] = -1L;
    }
    else {
        SET_NEXT(link[last], pos);
    }
}

```

```

    SET_UPPER_LOWER(link[pos], last, -1);
    last = pos;
}
#endif

if (size++ >= maxFill) rehash(arraySize(size + 1, f));
if (ASSERTS) checkTable();
}

@Override
public VALUE_GENERIC_TYPE put(final KEY_GENERIC_TYPE k, final VALUE_GENERIC_TYPE v) {
    final int pos = find(k);
    if (pos < 0) {
        insert(-pos - 1, k, v);
        return defRetVal;
    }
    final VALUE_GENERIC_TYPE oldValue = value[pos];
    value[pos] = v;
    return oldValue;
}

#if VALUE_CLASS_Byte || VALUE_CLASS_Short || VALUE_CLASS_Char || VALUE_CLASS_Integer ||
VALUE_CLASS_Long || VALUE_CLASS_Float || VALUE_CLASS_Double

private VALUE_GENERIC_TYPE addToValue(final int pos, final VALUE_GENERIC_TYPE incr) {
    final VALUE_GENERIC_TYPE oldValue = value[pos];
#if VALUE_CLASS_Byte || VALUE_CLASS_Short || VALUE_CLASS_Char
    value[pos] = (VALUE_TYPE)(oldValue + incr);
#else
    value[pos] = oldValue + incr;
#endif
    return oldValue;
}

/** Adds an increment to value currently associated with a key.
 *
 * <p>Note that this method respects the { @linkplain #defaultReturnValue() default return value } semantics: when
 * called with a key that does not currently appears in the map, the key
 * will be associated with the default return value plus
 * the given increment.
 *
 * @param k the key.
 * @param incr the increment.
 * @return the old value, or the { @linkplain #defaultReturnValue() default return value } if no value was present for
the given key.
 */
public VALUE_GENERIC_TYPE addTo(final KEY_GENERIC_TYPE k, final VALUE_GENERIC_TYPE incr) {

```

```

int pos;

if (KEY_EQUALS_NULL(k)) {
    if (containsNullKey) return addToValue(n, incr);
    pos = n;
    containsNullKey = true;
}
else {
    KEY_GENERIC_TYPE curr;
    final KEY_GENERIC_TYPE[] key = this.key;

    // The starting point.
    if (! KEY_IS_NULL(curr = key[pos = KEY2INTHASH(k) & mask])) {
        if (KEY_EQUALS_NOT_NULL(curr, k)) return addToValue(pos, incr);
        while(! KEY_IS_NULL(curr = key[pos = (pos + 1) & mask]))
            if (KEY_EQUALS_NOT_NULL(curr, k)) return addToValue(pos, incr);
    }
}

key[pos] = k;

#if VALUE_CLASS_Byte || VALUE_CLASS_Short || VALUE_CLASS_Char
    value[pos] = (VALUE_TYPE)(defRetValue + incr);
#else
    value[pos] = defRetValue + incr;
#endif

#ifdef Linked
    if (size == 0) {
        first = last = pos;
        // Special case of SET_UPPER_LOWER(link[pos], -1, -1);
        link[pos] = -1L;
    }
    else {
        SET_NEXT(link[last], pos);
        SET_UPPER_LOWER(link[pos], last, -1);
        last = pos;
    }
#endif

    if (size++ >= maxFill) rehash(arraySize(size + 1, f));
    if (ASSERTS) checkTable();
    return defRetValue;
}

#endif

/** Shifts left entries with the specified hash code, starting at the specified position,

```

```

* and empties the resulting free entry.
*
* @param pos a starting position.
*/
protected final void shiftKeys(int pos) {
    // Shift entries with the same hash.
    int last, slot;
    KEY_GENERIC_TYPE curr;
    final KEY_GENERIC_TYPE[] key = this.key;

    for(;;) {
        pos = ((last = pos) + 1) & mask;

        for(;;) {
            if (KEY_IS_NULL(curr = key[pos])) {
                key[last] = KEY_NULL;
                #if VALUES_REFERENCE
                value[last] = null;
                #endif
                return;
            }
            slot = KEY2INTHASH(curr) & mask;
            if (last <= pos ? last >= slot || slot > pos : last >= slot && slot > pos) break;
            pos = (pos + 1) & mask;
        }

        key[last] = curr;
        value[last] = value[pos];
        #ifdef Linked
        fixPointers(pos, last);
        #endif
    }
}

@Override
SUPPRESS_WARNINGS_KEY_UNCHECKED
public VALUE_GENERIC_TYPE REMOVE_VALUE(final KEY_TYPE k) {
    if (KEY_EQUALS_NULL(KEY_GENERIC_CAST k)) {
        if (containsNullKey) return removeNullEntry();
        return defRetValue;
    }

    KEY_GENERIC_TYPE curr;
    final KEY_GENERIC_TYPE[] key = this.key;
    int pos;

    // The starting point.
    if (KEY_IS_NULL(curr = key[pos = KEY2INTHASH_CAST(k) & mask])) return defRetValue;

```

```

if (KEY_EQUALS_NOT_NULL_CAST(k, curr)) return removeEntry(pos);
while(true) {
    if (KEY_IS_NULL(curr = key[pos = (pos + 1) & mask])) return defRetVal;
    if (KEY_EQUALS_NOT_NULL_CAST(k, curr)) return removeEntry(pos);
}
}

#ifdef Linked

private VALUE_GENERIC_TYPE setValue(final int pos, final VALUE_GENERIC_TYPE v) {
    final VALUE_GENERIC_TYPE oldValue = value[pos];
    value[pos] = v;
    return oldValue;
}

/** Removes the mapping associated with the first key in iteration order.
 * @return the value previously associated with the first key in iteration order.
 * @throws NoSuchElementException is this map is empty.
 */
public VALUE_GENERIC_TYPE REMOVE_FIRST_VALUE() {
    if (size == 0) throw new NoSuchElementException();
    final int pos = first;
    // Abbreviated version of fixPointers(pos)
    first = GET_NEXT(link[pos]);
    if (0 <= first) {
        // Special case of SET_PREV(link[first], -1)
        link[first] |= (-1 & 0xFFFFFFFFL) << 32;
    }
    size--;
    final VALUE_GENERIC_TYPE v = value[pos];
    if (pos == n) {
        containsNullKey = false;
#ifdef KEYS_REFERENCE
        key[n] = null;
#endif
#ifdef VALUES_REFERENCE
        value[n] = null;
#endif
    }
    else shiftKeys(pos);
    if (n > minN && size < maxFill / 4 && n > DEFAULT_INITIAL_SIZE) rehash(n / 2);
    return v;
}

/** Removes the mapping associated with the last key in iteration order.
 * @return the value previously associated with the last key in iteration order.
 * @throws NoSuchElementException is this map is empty.
 */

```

```

*/
public VALUE_GENERIC_TYPE REMOVE_LAST_VALUE() {
    if (size == 0) throw new NoSuchElementException();
    final int pos = last;
    // Abbreviated version of fixPointers(pos)
    last = GET_PREV(link[pos]);
    if (0 <= last) {
        // Special case of SET_NEXT(link[last], -1)
        link[last] |= -1 & 0xFFFFFFFFFL;
    }
    size--;
    final VALUE_GENERIC_TYPE v = value[pos];
    if (pos == n) {
        containsNullKey = false;
#ifdef KEYS_REFERENCE
        key[n] = null;
#endif
#ifdef VALUES_REFERENCE
        value[n] = null;
#endif
    }
    else shiftKeys(pos);
    if (n > minN && size < maxFill / 4 && n > DEFAULT_INITIAL_SIZE) rehash(n / 2);
    return v;
}

private void moveIndexToFirst(final int i) {
    if (size == 1 || first == i) return;
    if (last == i) {
        last = GET_PREV(link[i]);
        // Special case of SET_NEXT(link[last], -1);
        link[last] |= -1 & 0xFFFFFFFFFL;
    }
    else {
        final long linki = link[i];
        final int prev = GET_PREV(linki);
        final int next = GET_NEXT(linki);
        COPY_NEXT(link[prev], linki);
        COPY_PREV(link[next], linki);
    }
    SET_PREV(link[first], i);
    SET_UPPER_LOWER(link[i], -1, first);
    first = i;
}

private void moveIndexToLast(final int i) {
    if (size == 1 || last == i) return;
    if (first == i) {

```

```

first = GET_NEXT(link[i]);
// Special case of SET_PREV(link[first], -1);
link[first] |= (-1 & 0xFFFFFFFFL) << 32;
}
else {
final long linki = link[i];
final int prev = GET_PREV(linki);
final int next = GET_NEXT(linki);
COPY_NEXT(link[prev], linki);
COPY_PREV(link[next], linki);
}
SET_NEXT(link[last], i);
SET_UPPER_LOWER(link[i], last, -1);
last = i;
}

/** Returns the value to which the given key is mapped; if the key is present, it is moved to the first position of the
iteration order.
*
* @param k the key.
* @return the corresponding value, or the {@linkplain #defaultReturnValue() default return value} if no value was
present for the given key.
*/
public VALUE_GENERIC_TYPE getAndMoveToFirst(final KEY_GENERIC_TYPE k) {
if (KEY_EQUALS_NULL(k)) {
if (containsNullKey) {
moveIndexToFirst(n);
return value[n];
}

return defRetVal;
}

KEY_GENERIC_TYPE curr;
final KEY_GENERIC_TYPE[] key = this.key;
int pos;

// The starting point.
if (KEY_IS_NULL(curr = key[pos = KEY2INTHASH(k) & mask])) return defRetVal;
if (KEY_EQUALS_NOT_NULL(k, curr)) {
moveIndexToFirst(pos);
return value[pos];
}

// There's always an unused entry.
while(true) {
if (KEY_IS_NULL(curr = key[pos = (pos + 1) & mask])) return defRetVal;
if (KEY_EQUALS_NOT_NULL(k, curr)) {

```

```

    moveIndexToFirst(pos);
    return value[pos];
}
}
}

/** Returns the value to which the given key is mapped; if the key is present, it is moved to the last position of the
iteration order.
*
* @param k the key.
* @return the corresponding value, or the {@linkplain #defaultReturnValue() default return value} if no value was
present for the given key.
*/
public VALUE_GENERIC_TYPE getAndMoveToLast(final KEY_GENERIC_TYPE k) {
    if (KEY_EQUALS_NULL(k)) {
        if (containsNullKey) {
            moveIndexToLast(n);
            return value[n];
        }

        return defRetValue;
    }

    KEY_GENERIC_TYPE curr;
    final KEY_GENERIC_TYPE[] key = this.key;
    int pos;

    // The starting point.
    if (KEY_IS_NULL(curr = key[pos = KEY2INTHASH(k) & mask])) return defRetValue;
    if (KEY_EQUALS_NOT_NULL(k, curr)) {
        moveIndexToLast(pos);
        return value[pos];
    }

    // There's always an unused entry.
    while(true) {
        if (KEY_IS_NULL(curr = key[pos = (pos + 1) & mask])) return defRetValue;
        if (KEY_EQUALS_NOT_NULL(k, curr)) {
            moveIndexToLast(pos);
            return value[pos];
        }
    }
}

/** Adds a pair to the map; if the key is already present, it is moved to the first position of the iteration order.
*
* @param k the key.
* @param v the value.

```

* @return the old value, or the { @linkplain #defaultReturnValue() default return value } if no value was present for the given key.

*/

```
public VALUE_GENERIC_TYPE putAndMoveToFirst(final KEY_GENERIC_TYPE k, final  
VALUE_GENERIC_TYPE v) {
```

```
    int pos;
```

```
    if (KEY_EQUALS_NULL(k)) {
```

```
        if (containsNullKey) {
```

```
            moveIndexToFirst(n);
```

```
            return setValue(n, v);
```

```
        }
```

```
        containsNullKey = true;
```

```
        pos = n;
```

```
    }
```

```
    else {
```

```
        KEY_GENERIC_TYPE curr;
```

```
        final KEY_GENERIC_TYPE[] key = this.key;
```

```
        // The starting point.
```

```
        if (! KEY_IS_NULL(curr = key[pos = KEY2INTHASH(k) & mask])) {
```

```
            if (KEY_EQUALS_NOT_NULL(curr, k)) {
```

```
                moveIndexToFirst(pos);
```

```
                return setValue(pos, v);
```

```
            }
```

```
            while(! KEY_IS_NULL(curr = key[pos = (pos + 1) & mask]))
```

```
                if (KEY_EQUALS_NOT_NULL(curr, k)) {
```

```
                    moveIndexToFirst(pos);
```

```
                    return setValue(pos, v);
```

```
                }
```

```
            }
```

```
        }
```

```
        key[pos] = k;
```

```
        value[pos] = v;
```

```
        if (size == 0) {
```

```
            first = last = pos;
```

```
            // Special case of SET_UPPER_LOWER(link[pos], -1, -1);
```

```
            link[pos] = -1L;
```

```
        }
```

```
        else {
```

```
            SET_PREV(link[first], pos);
```

```
            SET_UPPER_LOWER(link[pos], -1, first);
```

```
            first = pos;
```

```
        }
```

```
        if (size++ >= maxFill) rehash(arraySize(size, f));
```

```

if (ASSERTS) checkTable();
return defRetVal;
}

/** Adds a pair to the map; if the key is already present, it is moved to the last position of the iteration order.
 *
 * @param k the key.
 * @param v the value.
 * @return the old value, or the { @linkplain #defaultReturnValue() default return value } if no value was present for
the given key.
 */
public VALUE_GENERIC_TYPE putAndMoveToLast(final KEY_GENERIC_TYPE k, final
VALUE_GENERIC_TYPE v) {
    int pos;

    if (KEY_EQUALS_NULL(k)) {
        if (containsNullKey) {
            moveIndexToLast(n);
            return setValue(n, v);
        }
        containsNullKey = true;
        pos = n;
    }
    else {
        KEY_GENERIC_TYPE curr;
        final KEY_GENERIC_TYPE[] key = this.key;

        // The starting point.
        if (! KEY_IS_NULL(curr = key[pos = KEY2INTHASH(k) & mask])) {
            if (KEY_EQUALS_NOT_NULL(curr, k)) {
                moveIndexToLast(pos);
                return setValue(pos, v);
            }
            while(! KEY_IS_NULL(curr = key[pos = (pos + 1) & mask]))
                if (KEY_EQUALS_NOT_NULL(curr, k)) {
                    moveIndexToLast(pos);
                    return setValue(pos, v);
                }
        }
    }

    key[pos] = k;
    value[pos] = v;

    if (size == 0) {
        first = last = pos;
        // Special case of SET_UPPER_LOWER(link[pos], -1, -1);
        link[pos] = -1L;
    }
}

```

```

    }
    else {
        SET_NEXT(link[last], pos);
        SET_UPPER_LOWER(link[pos], last, -1);
        last = pos;
    }

    if (size++ >= maxFill) rehash(arraySize(size, f));
    if (ASSERTS) checkTable();
    return defRetVal;
}

#endif

@Override
SUPPRESS_WARNINGS_KEY_UNCHECKED
public VALUE_GENERIC_TYPE GET_VALUE(final KEY_TYPE k) {
    if (KEY_EQUALS_NULL(KEY_GENERIC_CAST k)) return containsNullKey ? value[n] : defRetVal;

    KEY_GENERIC_TYPE curr;
    final KEY_GENERIC_TYPE[] key = this.key;
    int pos;

    // The starting point.
    if (KEY_IS_NULL(curr = key[pos = KEY2INTHASH_CAST(k) & mask])) return defRetVal;
    if (KEY_EQUALS_NOT_NULL_CAST(k, curr)) return value[pos];
    // There's always an unused entry.
    while(true) {
        if (KEY_IS_NULL(curr = key[pos = (pos + 1) & mask])) return defRetVal;
        if (KEY_EQUALS_NOT_NULL_CAST(k, curr)) return value[pos];
    }
}

@Override
SUPPRESS_WARNINGS_KEY_UNCHECKED
public boolean containsKey(final KEY_TYPE k) {
    if (KEY_EQUALS_NULL(KEY_GENERIC_CAST k)) return containsNullKey;

    KEY_GENERIC_TYPE curr;
    final KEY_GENERIC_TYPE[] key = this.key;
    int pos;

    // The starting point.
    if (KEY_IS_NULL(curr = key[pos = KEY2INTHASH_CAST(k) & mask])) return false;
    if (KEY_EQUALS_NOT_NULL_CAST(k, curr)) return true;
    // There's always an unused entry.
    while(true) {
        if (KEY_IS_NULL(curr = key[pos = (pos + 1) & mask])) return false;

```

```

    if (KEY_EQUALS_NOT_NULL_CAST(k, curr)) return true;
  }
}

```

@Override

```

public boolean containsValue(final VALUE_TYPE v) {
    final VALUE_GENERIC_TYPE value[] = this.value;
    final KEY_GENERIC_TYPE key[] = this.key;
    if (containsNullKey && VALUE_EQUALS(value[n], v)) return true;
    for(int i = n; i-- != 0;) if (! KEY_IS_NULL(key[i]) && VALUE_EQUALS(value[i], v)) return true;
    return false;
}

```

#if KEYS_PRIMITIVE || VALUES_PRIMITIVE

/** {@inheritDoc} */

@Override

SUPPRESS_WARNINGS_KEY_UNCHECKED

```

public VALUE_GENERIC_TYPE getOrDefault(final KEY_TYPE k, final VALUE_GENERIC_TYPE
defaultValue) {
    if (KEY_EQUALS_NULL(KEY_GENERIC_CAST k)) return containsNullKey ? value[n] : defaultValue;

```

KEY_GENERIC_TYPE curr;

final KEY_GENERIC_TYPE[] key = this.key;

int pos;

// The starting point.

if (KEY_IS_NULL(curr = key[pos = KEY2INTHASH_CAST(k) & mask])) return defaultValue;

if (KEY_EQUALS_NOT_NULL_CAST(k, curr)) return value[pos];

// There's always an unused entry.

while(true) {

if (KEY_IS_NULL(curr = key[pos = (pos + 1) & mask])) return defaultValue;

if (KEY_EQUALS_NOT_NULL_CAST(k, curr)) return value[pos];

}

}

/** {@inheritDoc} */

@Override

```

public VALUE_GENERIC_TYPE putIfAbsent(final KEY_GENERIC_TYPE k, final VALUE_GENERIC_TYPE
v) {
    final int pos = find(k);
    if (pos >= 0) return value[pos];
    insert(-pos - 1, k, v);
    return defRetVal;
}

```

```

/** {@inheritDoc} */
@Override
SUPPRESS_WARNINGS_KEY_UNCHECKED
public boolean remove(final KEY_TYPE k, final VALUE_TYPE v) {
    if (KEY_EQUALS_NULL(KEY_GENERIC_CAST k)) {
        if (containsNullKey && VALUE_EQUALS(v, value[n])) {
            removeNullEntry();
            return true;
        }
        return false;
    }

    KEY_GENERIC_TYPE curr;
    final KEY_GENERIC_TYPE[] key = this.key;
    int pos;

    // The starting point.
    if (KEY_IS_NULL(curr = key[pos = KEY2INTHASH_CAST(k) & mask])) return false;
    if (KEY_EQUALS_NOT_NULL_CAST(k, curr) && VALUE_EQUALS(v, value[pos])) {
        removeEntry(pos);
        return true;
    }
    while(true) {
        if (KEY_IS_NULL(curr = key[pos = (pos + 1) & mask])) return false;
        if (KEY_EQUALS_NOT_NULL_CAST(k, curr) && VALUE_EQUALS(v, value[pos])) {
            removeEntry(pos);
            return true;
        }
    }
}

/** {@inheritDoc} */
@Override
public boolean replace(final KEY_GENERIC_TYPE k, final VALUE_GENERIC_TYPE oldValue, final
VALUE_GENERIC_TYPE v) {
    final int pos = find(k);
    if (pos < 0 || ! VALUE_EQUALS(oldValue, value[pos])) return false;
    value[pos] = v;
    return true;
}

/** {@inheritDoc} */
@Override
public VALUE_GENERIC_TYPE replace(final KEY_GENERIC_TYPE k, final VALUE_GENERIC_TYPE v) {
    final int pos = find(k);
    if (pos < 0) return defRetValue;
    final VALUE_GENERIC_TYPE oldValue = value[pos];
    value[pos] = v;
}

```

```

return oldValue;
}

#ifdef JDK_PRIMITIVE_FUNCTION

/** {@inheritDoc} */
@Override
public VALUE_GENERIC_TYPE COMPUTE_IF_ABSENT_JDK(final KEY_GENERIC_TYPE k, final
JDK_PRIMITIVE_FUNCTION KEY_SUPER_GENERIC VALUE_EXTENDS_GENERIC mappingFunction) {
    java.util.Objects.requireNonNull(mappingFunction);
    final int pos = find(k);
    if (pos >= 0) return value[pos];
    final VALUE_GENERIC_TYPE newValue =
VALUE_NARROWING(mappingFunction.JDK_PRIMITIVE_FUNCTION_APPLY(k));
    insert(-pos - 1, k, newValue);
    return newValue;
}

#endif

#if KEYS_PRIMITIVE && VALUES_PRIMITIVE

/** {@inheritDoc} */
@Override
public VALUE_GENERIC_TYPE COMPUTE_IF_ABSENT_NULLABLE(final KEY_GENERIC_TYPE k, final
JDK_KEY_TO_GENERIC_FUNCTION<? extends VALUE_GENERIC_CLASS> mappingFunction) {
    java.util.Objects.requireNonNull(mappingFunction);
    final int pos = find(k);
    if (pos >= 0) return value[pos];
    final VALUE_GENERIC_CLASS newValue = mappingFunction.apply(k);
    if (newValue == null) return defRetValue;
    final VALUE_GENERIC_TYPE v = VALUE_CLASS2TYPE(newValue);
    insert(-pos - 1, k, v);
    return v;
}

#endif

/** {@inheritDoc} */
@Override
public VALUE_GENERIC_TYPE COMPUTE_IF_PRESENT(final KEY_GENERIC_TYPE k, final
java.util.function.BiFunction<? super KEY_GENERIC_CLASS, ? super VALUE_GENERIC_CLASS, ? extends
VALUE_GENERIC_CLASS> remappingFunction) {
    java.util.Objects.requireNonNull(remappingFunction);
    final int pos = find(k);
    if (pos < 0) return defRetValue;
    final VALUE_GENERIC_CLASS newValue = remappingFunction.apply(KEY2OBJ(k),
VALUE2OBJ(value[pos]));

```

```

if (newValue == null) {
    if (KEY_EQUALS_NULL(k)) removeNullEntry();
    else removeEntry(pos);
    return defRetValue;
}
return value[pos] = VALUE_CLASS2TYPE(newValue);
}

/** {@inheritDoc} */
@Override
public VALUE_GENERIC_TYPE COMPUTE(final KEY_GENERIC_TYPE k, final
java.util.function.BiFunction<? super KEY_GENERIC_CLASS, ? super VALUE_GENERIC_CLASS, ? extends
VALUE_GENERIC_CLASS> remappingFunction) {
    java.util.Objects.requireNonNull(remappingFunction);
    final int pos = find(k);
    final VALUE_GENERIC_CLASS newValue = remappingFunction.apply(KEY2OBJ(k), pos >= 0 ?
VALUE2OBJ(value[pos]) : null);
    if (newValue == null) {
        if (pos >= 0) {
            if (KEY_EQUALS_NULL(k)) removeNullEntry();
            else removeEntry(pos);
        }
        return defRetValue;
    }

    VALUE_GENERIC_TYPE newVal = VALUE_CLASS2TYPE(newValue);
    if (pos < 0) {
        insert(-pos - 1, k, newVal);
        return newVal;
    }

    return value[pos] = newVal;
}

/** {@inheritDoc} */
@Override
public VALUE_GENERIC_TYPE MERGE(final KEY_GENERIC_TYPE k, final VALUE_GENERIC_TYPE v,
final java.util.function.BiFunction<? super VALUE_GENERIC_CLASS, ? super VALUE_GENERIC_CLASS, ?
extends VALUE_GENERIC_CLASS> remappingFunction) {
    java.util.Objects.requireNonNull(remappingFunction);

    final int pos = find(k);
    #if VALUES_PRIMITIVE
    if (pos < 0) {
    #else
    if (pos < 0 || value[pos] == null) {
        if (v == null) return defRetValue;
    #endif
}

```

```

insert(-pos - 1, k, v);
return v;
}

final VALUE_GENERIC_CLASS newValue = remappingFunction.apply(VALUE2OBJ(value[pos]),
VALUE2OBJ(v));
if (newValue == null) {
if (KEY_EQUALS_NULL(k)) removeNullEntry();
else removeEntry(pos);
return defRetValue;
}

return value[pos] = VALUE_CLASS2TYPE(newValue);
}

#endif

/* Removes all elements from this map.
*
* <p>To increase object reuse, this method does not change the table size.
* If you want to reduce the table size, you must use {@link #trim()}.
*
*/
@Override
public void clear() {
if (size == 0) return;
size = 0;
containsNullKey = false;

Arrays.fill(key, KEY_NULL);
#if VALUES_REFERENCE
Arrays.fill(value, null);
#endif

#ifdef Linked
first = last = -1;
#endif
}

@Override
public int size() {
return size;
}

@Override
public boolean isEmpty() {
return size == 0;
}

```

```
}
```

```
/** The entry class for a hash map does not record key and value, but  
 * rather the position in the hash table of the corresponding entry. This  
 * is necessary so that calls to {@link java.util.Map.Entry#setValue(Object)} are reflected in  
 * the map */
```

```
final class MapEntry implements MAP.Entry KEY_VALUE_GENERIC, Map.Entry<KEY_GENERIC_CLASS,  
VALUE_GENERIC_CLASS> {
```

```
// The table index this entry refers to, or -1 if this entry has been deleted.
```

```
int index;
```

```
MapEntry(final int index) {  
    this.index = index;  
}
```

```
MapEntry() {}
```

```
@Override  
public KEY_GENERIC_TYPE ENTRY_GET_KEY() {  
    return key[index];  
}
```

```
@Override  
public VALUE_GENERIC_TYPE ENTRY_GET_VALUE() {  
    return value[index];  
}
```

```
@Override  
public VALUE_GENERIC_TYPE setValue(final VALUE_GENERIC_TYPE v) {  
    final VALUE_GENERIC_TYPE oldValue = value[index];  
    value[index] = v;  
    return oldValue;  
}
```

```
#if KEYS_PRIMITIVE
```

```
/** {@inheritDoc}
```

```
 * @deprecated Please use the corresponding type-specific method instead. */
```

```
@Deprecated
```

```
@Override
```

```
public KEY_GENERIC_CLASS getKey() {  
    return KEY2OBJ(key[index]);  
}
```

```
#endif
```

```
#if VALUES_PRIMITIVE
```

```
/** {@inheritDoc}
```

```

    * @deprecated Please use the corresponding type-specific method instead. */
    @Deprecated
    @Override
    public VALUE_GENERIC_CLASS getValue() {
        return VALUE2OBJ(value[index]);
    }

    /** {@inheritDoc}
     * @deprecated Please use the corresponding type-specific method instead. */
    @Deprecated
    @Override
    public VALUE_GENERIC_CLASS setValue(final VALUE_GENERIC_CLASS v) {
        return VALUE2OBJ(setValue(VALUE_CLASS2TYPE(v)));
    }
}

#endif

    @SuppressWarnings("unchecked")
    @Override
    public boolean equals(final Object o) {
        if (!(o instanceof Map.Entry)) return false;
        Map.Entry<KEY_GENERIC_CLASS, VALUE_GENERIC_CLASS> e = (Map.Entry<KEY_GENERIC_CLASS,
VALUE_GENERIC_CLASS>)o;

        return KEY_EQUALS(key[index], KEY_CLASS2TYPE(e.getKey())) && VALUE_EQUALS(value[index],
VALUE_CLASS2TYPE(e.getValue()));
    }

    @Override
    public int hashCode() {
        return KEY2JAVAHASH(key[index]) ^ VALUE2JAVAHASH(value[index]);
    }

    @Override
    public String toString() {
        return key[index] + "=>" + value[index];
    }
}

#ifndef Linked

/** Modifies the {@link #link} vector so that the given entry is removed.
 * This method will complete in constant time.
 *
 * @param i the index of an entry.
 */
protected void fixPointers(final int i) {
    if (size == 0) {

```

```

    first = last = -1;
    return;
}
if (first == i) {
    first = GET_NEXT(link[i]);
    if (0 <= first) {
        // Special case of SET_PREV(link[first], -1)
        link[first] |= (-1 & 0xFFFFFFFFL) << 32;
    }
    return;
}
if (last == i) {
    last = GET_PREV(link[i]);
    if (0 <= last) {
        // Special case of SET_NEXT(link[last], -1)
        link[last] |= -1 & 0xFFFFFFFFL;
    }
    return;
}
final long linki = link[i];
final int prev = GET_PREV(linki);
final int next = GET_NEXT(linki);
COPY_NEXT(link[prev], linki);
COPY_PREV(link[next], linki);
}

/** Modifies the {@link #link} vector for a shift from s to d.
 * <p>This method will complete in constant time.
 *
 * @param s the source position.
 * @param d the destination position.
 */
protected void fixPointers(int s, int d) {
    if (size == 1) {
        first = last = d;
        // Special case of SET_UPPER_LOWER(link[d], -1, -1)
        link[d] = -1L;
        return;
    }
    if (first == s) {
        first = d;
        SET_PREV(link[GET_NEXT(link[s])], d);
        link[d] = link[s];
        return;
    }
    if (last == s) {
        last = d;

```

```

    SET_NEXT(link[GET_PREV(link[s])], d);
    link[d] = link[s];
    return;
}
final long links = link[s];
final int prev = GET_PREV(links);
final int next = GET_NEXT(links);
SET_NEXT(link[prev], d);
SET_PREV(link[next], d);
link[d] = links;
}

/** Returns the first key of this map in iteration order.
 *
 * @return the first key in iteration order.
 */
@Override
public KEY_GENERIC_TYPE FIRST_KEY() {
    if (size == 0) throw new NoSuchElementException();
    return key[first];
}

/** Returns the last key of this map in iteration order.
 *
 * @return the last key in iteration order.
 */
@Override
public KEY_GENERIC_TYPE LAST_KEY() {
    if (size == 0) throw new NoSuchElementException();
    return key[last];
}

/** {@inheritDoc}
 * <p>This implementation just throws an {@link UnsupportedOperationException}.*/
@Override
public SORTED_MAP KEY_VALUE_GENERIC tailMap(KEY_GENERIC_TYPE from) { throw new
UnsupportedOperationException(); }

/** {@inheritDoc}
 * <p>This implementation just throws an {@link UnsupportedOperationException}.*/
@Override
public SORTED_MAP KEY_VALUE_GENERIC headMap(KEY_GENERIC_TYPE to) { throw new
UnsupportedOperationException(); }

/** {@inheritDoc}
 * <p>This implementation just throws an {@link UnsupportedOperationException}.*/

```

```

@Override
public SORTED_MAP KEY_VALUE_GENERIC subMap(KEY_GENERIC_TYPE from, KEY_GENERIC_TYPE
to) { throw new UnsupportedOperationException(); }

/** {@inheritDoc}
 * <p>This implementation just returns {@code null}.*/
@Override
public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() { return null; }

/** A list iterator over a linked map.
 *
 * <p>This class provides a list iterator over a linked hash map. The constructor runs in constant time.
 */
private class MapIterator {
    /** The entry that will be returned by the next call to {@link java.util.ListIterator#previous()} (or {@code null} if
no previous entry exists). */
    int prev = -1;
    /** The entry that will be returned by the next call to {@link java.util.ListIterator#next()} (or {@code null} if no
next entry exists). */
    int next = -1;
    /** The last entry that was returned (or -1 if we did not iterate or used {@link java.util.Iterator#remove()}). */
    int curr = -1;
    /** The current index (in the sense of a {@link java.util.ListIterator}). Note that this value is not meaningful when
this iterator has been created using the nonempty constructor.*/
    int index = -1;

    protected MapIterator() {
        next = first;
        index = 0;
    }

    private MapIterator(final KEY_GENERIC_TYPE from) {
        if (KEY_EQUALS_NULL(from)) {
            if (OPEN_HASH_MAP.this.containsNullKey) {
                next = GET_NEXT(link[n]);
                prev = n;
                return;
            }
            else throw new NoSuchElementException("The key " + from + " does not belong to this map.");
        }

        if (KEY_EQUALS(key[last], from)) {
            prev = last;
            index = size;
            return;
        }
    }

```

```

// The starting point.
int pos = KEY2INTHASH(from) & mask;

// There's always an unused entry.
while(! KEY_IS_NULL(key[pos])) {
    if (KEY_EQUALS_NOT_NULL(key[pos], from)) {
        // Note: no valid index known.
        next = GET_NEXT(link[pos]);
        prev = pos;
        return;
    }
    pos = (pos + 1) & mask;
}
throw new NoSuchElementException("The key " + from + " does not belong to this map.");
}

public boolean hasNext() { return next != -1; }

public boolean hasPrevious() { return prev != -1; }

private final void ensureIndexKnown() {
    if (index >= 0) return;
    if (prev == -1) {
        index = 0;
        return;
    }
    if (next == -1) {
        index = size;
        return;
    }
    int pos = first;
    index = 1;
    while(pos != prev) {
        pos = GET_NEXT(link[pos]);
        index++;
    }
}

public int nextIndex() {
    ensureIndexKnown();
    return index;
}

public int previousIndex() {
    ensureIndexKnown();
    return index - 1;
}

```

```

public int nextEntry() {
    if (! hasNext()) throw new NoSuchElementException();

    curr = next;
    next = GET_NEXT(link[curr]);
    prev = curr;

    if (index >= 0) index++;

    return curr;
}

public int previousEntry() {
    if (! hasPrevious()) throw new NoSuchElementException();

    curr = prev;
    prev = GET_PREV(link[curr]);
    next = curr;

    if (index >= 0) index--;

    return curr;
}

public void remove() {
    ensureIndexKnown();
    if (curr == -1) throw new IllegalStateException();

    if (curr == prev) {
        /* If the last operation was a next(), we are removing an entry that preceeds
           the current index, and thus we must decrement it. */
        index--;
        prev = GET_PREV(link[curr]);
    }
    else
        next = GET_NEXT(link[curr]);

    size--;
    /* Now we manually fix the pointers. Because of our knowledge of next
       and prev, this is going to be faster than calling fixPointers(). */
    if (prev == -1) first = next;
    else
        SET_NEXT(link[prev], next);
    if (next == -1) last = prev;
    else
        SET_PREV(link[next], prev);

    int last, slot, pos = curr;

```

```

curr = -1;

if (pos == n) {
    OPEN_HASH_MAP.this.containsNullKey = false;
#if KEYS_REFERENCE
    key[n] = null;
#endif
#if VALUES_REFERENCE
    value[n] = null;
#endif
}
else {
    KEY_GENERIC_TYPE curr;
    final KEY_GENERIC_TYPE[] key = OPEN_HASH_MAP.this.key;
    // We have to horribly duplicate the shiftKeys() code because we need to update next/prev.
    for(;;) {
        pos = ((last = pos) + 1) & mask;
        for(;;) {
            if (KEY_IS_NULL(curr = key[pos])) {
                key[last] = KEY_NULL;
#if VALUES_REFERENCE
                value[last] = null;
#endif
            }
            return;
        }
        slot = KEY2INTHASH(curr) & mask;
        if (last <= pos ? last >= slot || slot > pos : last >= slot && slot > pos) break;
        pos = (pos + 1) & mask;
    }

    key[last] = curr;
    value[last] = value[pos];
    if (next == pos) next = last;
    if (prev == pos) prev = last;
    fixPointers(pos, last);
}
}

public int skip(final int n) {
    int i = n;
    while(i-- != 0 && hasNext()) nextEntry();
    return n - i - 1;
}

public int back(final int n) {
    int i = n;
    while(i-- != 0 && hasPrevious()) previousEntry();
}

```

```

return n - i - 1;
}

public void set(@SuppressWarnings("unused") MAP.Entry KEY_VALUE_GENERIC ok) {
    throw new UnsupportedOperationException();
}

public void add(@SuppressWarnings("unused") MAP.Entry KEY_VALUE_GENERIC ok) {
    throw new UnsupportedOperationException();
}
}

private class EntryIterator extends MapIterator implements ObjectListIterator<MAP.Entry
KEY_VALUE_GENERIC> {
    private MapEntry entry;

    public EntryIterator() {}

    public EntryIterator(KEY_GENERIC_TYPE from) {
        super(from);
    }

    @Override
    public MapEntry next() {
        return entry = new MapEntry(nextEntry());
    }

    @Override
    public MapEntry previous() {
        return entry = new MapEntry(previousEntry());
    }

    @Override
    public void remove() {
        super.remove();
        entry.index = -1; // You cannot use a deleted entry.
    }
}

private class FastEntryIterator extends MapIterator implements ObjectListIterator<MAP.Entry
KEY_VALUE_GENERIC> {
    final MapEntry entry = new MapEntry();

    public FastEntryIterator() {}

    public FastEntryIterator(KEY_GENERIC_TYPE from) {
        super(from);
    }
}

```

```

@Override
public MapEntry next() {
    entry.index = nextEntry();
    return entry;
}

@Override
public MapEntry previous() {
    entry.index = previousEntry();
    return entry;
}
}

#else

/** An iterator over a hash map. */

private class MapIterator {
    /** The index of the last entry returned, if positive or zero; initially, {@link #n}. If negative, the last
        entry returned was that of the key of index {@code - pos - 1} from the {@link #wrapped} list. */
    int pos = n;
    /** The index of the last entry that has been returned (more precisely, the value of {@link #pos} if {@link #pos} is
        positive,
        or {@link Integer#MIN_VALUE} if {@link #pos} is negative). It is -1 if either
        we did not return an entry yet, or the last returned entry has been removed. */
    int last = -1;
    /** A downward counter measuring how many entries must still be returned. */
    int c = size;
    /** A boolean telling us whether we should return the entry with the null key. */
    boolean mustReturnNullKey = OPEN_HASH_MAP.this.containsNullKey;
    /** A lazily allocated list containing keys of entries that have wrapped around the table because of removals. */
    ARRAY_LIST KEY_GENERIC wrapped;

    public boolean hasNext() {
        return c != 0;
    }

    public int nextEntry() {
        if (!hasNext()) throw new NoSuchElementException();

        c--;
        if (mustReturnNullKey) {
            mustReturnNullKey = false;
            return last = n;
        }

        final KEY_GENERIC_TYPE key[] = OPEN_HASH_MAP.this.key;

```

```

for(;;) {
    if (--pos < 0) {
        // We are just enumerating elements from the wrapped list.
        last = Integer.MIN_VALUE;
        final KEY_GENERIC_TYPE k = wrapped.GET_KEY(- pos - 1);
        int p = KEY2INTHASH(k) & mask;
        while (! KEY_EQUALS_NOT_NULL(k, key[p])) p = (p + 1) & mask;
        return p;
    }
    if (! KEY_IS_NULL(key[pos])) return last = pos;
}

/** Shifts left entries with the specified hash code, starting at the specified position,
 * and empties the resulting free entry.
 *
 * @param pos a starting position.
 */
private void shiftKeys(int pos) {
    // Shift entries with the same hash.
    int last, slot;
    KEY_GENERIC_TYPE curr;
    final KEY_GENERIC_TYPE[] key = OPEN_HASH_MAP.this.key;

    for(;;) {
        pos = ((last = pos) + 1) & mask;

        for(;;) {
            if (KEY_IS_NULL(curr = key[pos])) {
                key[last] = KEY_NULL;
                #if VALUES_REFERENCE
                value[last] = null;
            #endif
            return;
        }
        slot = KEY2INTHASH(curr) & mask;
        if (last <= pos ? last >= slot || slot > pos : last >= slot && slot > pos) break;
        pos = (pos + 1) & mask;
    }

    if (pos < last) { // Wrapped entry.
        if (wrapped == null) wrapped = new ARRAY_LIST KEY_GENERIC_DIAMOND(2);
        wrapped.add(key[pos]);
    }

    key[last] = curr;
    value[last] = value[pos];
}

```

```

    }
}

public void remove() {
    if (last == -1) throw new IllegalStateException();
    if (last == n) {
        containsNullKey = false;
#ifdef KEYS_REFERENCE
        key[n] = null;
#endif
#ifdef VALUES_REFERENCE
        value[n] = null;
#endif
    }
    else if (pos >= 0) shiftKeys(last);
    else {
        // We're removing wrapped entries.
#ifdef KEYS_REFERENCE
        OPEN_HASH_MAP.this.REMOVE_VALUE(wrapped.set(- pos - 1, null));
#else
        OPEN_HASH_MAP.this.REMOVE_VALUE(wrapped.GET_KEY(- pos - 1));
#endif
    }
    last = -1; // Note that we must not decrement size
    return;
}

size--;
last = -1; // You can no longer remove this entry.
if (ASSERTS) checkTable();
}

public int skip(final int n) {
    int i = n;
    while(i-- != 0 && hasNext()) nextEntry();
    return n - i - 1;
}
}

private class EntryIterator extends MapIterator implements ObjectIterator<MAP.Entry KEY_VALUE_GENERIC>
{
    private MapEntry entry;

    @Override
    public MapEntry next() {
        return entry = new MapEntry(nextEntry());
    }
}

```

```

@Override
public void remove() {
    super.remove();
    entry.index = -1; // You cannot use a deleted entry.
}
}

private class FastEntryIterator extends MapIterator implements ObjectIterator<MAP.Entry
KEY_VALUE_GENERIC> {
    private final MapEntry entry = new MapEntry();

    @Override
    public MapEntry next() {
        entry.index = nextEntry();
        return entry;
    }
}

#endif

#ifdef Linked
private final class MapEntrySet extends AbstractObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC>
implements FastSortedEntrySet KEY_VALUE_GENERIC {

    @Override
    public ObjectBidirectionalIterator<MAP.Entry KEY_VALUE_GENERIC> iterator() { return new EntryIterator(); }

    @Override
    public Comparator<? super MAP.Entry KEY_VALUE_GENERIC> comparator() { return null; }

    @Override
    public ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> subSet(MAP.Entry KEY_VALUE_GENERIC
fromElement, MAP.Entry KEY_VALUE_GENERIC toElement) { throw new UnsupportedOperationException(); }

    @Override
    public ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> headSet(MAP.Entry KEY_VALUE_GENERIC
toElement) { throw new UnsupportedOperationException(); }

    @Override
    public ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> tailSet(MAP.Entry KEY_VALUE_GENERIC
fromElement) { throw new UnsupportedOperationException(); }

    @Override
    public MAP.Entry KEY_VALUE_GENERIC first() {
        if (size == 0) throw new NoSuchElementException();
        return new MapEntry(OPEN_HASH_MAP.this.first());
    }
}

```

```

@Override
public MAP.Entry KEY_VALUE_GENERIC last() {
    if (size == 0) throw new NoSuchElementException();
    return new MapEntry(OPEN_HASH_MAP.this.last);
}

#else
private final class MapEntrySet extends AbstractObjectSet<MAP.Entry KEY_VALUE_GENERIC> implements
FastEntrySet KEY_VALUE_GENERIC {

    @Override
    public ObjectIterator<MAP.Entry KEY_VALUE_GENERIC> iterator() { return new EntryIterator(); }

    @Override
    public ObjectIterator<MAP.Entry KEY_VALUE_GENERIC> fastIterator() { return new FastEntryIterator(); }
#endif

    @Override
    SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
    public boolean contains(final Object o) {
        if (!(o instanceof Map.Entry)) return false;
        final Map.Entry<?,?> e = (Map.Entry<?,?>)o;
#if KEYS_PRIMITIVE
        if (e.getKey() == null || !(e.getKey() instanceof KEY_CLASS)) return false;
#endif
#if VALUES_PRIMITIVE
        if (e.getValue() == null || !(e.getValue() instanceof VALUE_CLASS)) return false;
#endif
        final KEY_GENERIC_TYPE k = KEY_OBJ2TYPE(KEY_GENERIC_CAST e.getKey());
        final VALUE_GENERIC_TYPE v = VALUE_OBJ2TYPE(VALUE_GENERIC_CAST e.getValue());

        if (KEY_EQUALS_NULL(k)) return OPEN_HASH_MAP.this.containsNullKey && VALUE_EQUALS(value[n],
v);

        KEY_GENERIC_TYPE curr;
        final KEY_GENERIC_TYPE[] key = OPEN_HASH_MAP.this.key;
        int pos;

        // The starting point.
        if (KEY_IS_NULL(curr = key[pos = KEY2INTHASH(k) & mask])) return false;
        if (KEY_EQUALS_NOT_NULL(k, curr)) return VALUE_EQUALS(value[pos], v);
        // There's always an unused entry.
        while(true) {
            if (KEY_IS_NULL(curr = key[pos = (pos + 1) & mask])) return false;
            if (KEY_EQUALS_NOT_NULL(k, curr)) return VALUE_EQUALS(value[pos], v);
        }
    }
}

```

```

@Override
SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
public boolean remove(final Object o) {
    if (!(o instanceof Map.Entry)) return false;
    final Map.Entry<?,?> e = (Map.Entry<?,?>)o;
#if KEYS_PRIMITIVE
    if (e.getKey() == null || !(e.getKey() instanceof KEY_CLASS)) return false;
#endif
#if VALUES_PRIMITIVE
    if (e.getValue() == null || !(e.getValue() instanceof VALUE_CLASS)) return false;
#endif
    final KEY_GENERIC_TYPE k = KEY_OBJ2TYPE(KEY_GENERIC_CAST e.getKey());
    final VALUE_GENERIC_TYPE v = VALUE_OBJ2TYPE(VALUE_GENERIC_CAST e.getValue());

    if (KEY_EQUALS_NULL(k) {
        if (containsNullKey && VALUE_EQUALS(value[n], v)) {
            removeNullEntry();
            return true;
        }
        return false;
    }

    KEY_GENERIC_TYPE curr;
    final KEY_GENERIC_TYPE[] key = OPEN_HASH_MAP.this.key;
    int pos;

    // The starting point.
    if (KEY_IS_NULL(curr = key[pos = KEY2INTHASH(k) & mask])) return false;
    if (KEY_EQUALS_NOT_NULL(curr, k)) {
        if (VALUE_EQUALS(value[pos], v)) {
            removeEntry(pos);
            return true;
        }
        return false;
    }

    while(true) {
        if (KEY_IS_NULL(curr = key[pos = (pos + 1) & mask])) return false;
        if (KEY_EQUALS_NOT_NULL(curr, k)) {
            if (VALUE_EQUALS(value[pos], v)) {
                removeEntry(pos);
                return true;
            }
        }
    }
}

```

```
@Override
public int size() {
    return size;
}
```

```
@Override
public void clear() {
    OPEN_HASH_MAP.this.clear();
}
```

```
#ifdef Linked
```

```
/** Returns a type-specific list iterator on the elements in this set, starting from a given element of the set.
 * Please see the class documentation for implementation details.
 *
 * @param from an element to start from.
 * @return a type-specific list iterator starting at the given element.
 * @throws IllegalArgumentException if { @code from } does not belong to the set.
 */
```

```
@Override
public ObjectListIterator<MAP.Entry KEY_VALUE_GENERIC> iterator(final MAP.Entry
KEY_VALUE_GENERIC from) {
    return new EntryIterator(from.ENTRY_GET_KEY());
}
```

```
/** Returns a type-specific fast list iterator on the elements in this set, starting from the first element.
 * Please see the class documentation for implementation details.
 *
 * @return a type-specific list iterator starting at the first element.
 */
```

```
@Override
public ObjectListIterator<MAP.Entry KEY_VALUE_GENERIC> fastIterator() {
    return new FastEntryIterator();
}
```

```
/** Returns a type-specific fast list iterator on the elements in this set, starting from a given element of the set.
 * Please see the class documentation for implementation details.
 *
 * @param from an element to start from.
 * @return a type-specific list iterator starting at the given element.
 * @throws IllegalArgumentException if { @code from } does not belong to the set.
 */
```

```
@Override
public ObjectListIterator<MAP.Entry KEY_VALUE_GENERIC> fastIterator(final MAP.Entry
KEY_VALUE_GENERIC from) {
    return new FastEntryIterator(from.ENTRY_GET_KEY());
}
```

```
/** { @inheritDoc } */
```

```

@Override
public void forEach(final Consumer<? super MAP.Entry KEY_VALUE_GENERIC> consumer) {
    for(int i = size, curr, next = first; i-- != 0;) {
        curr = next;
        next = GET_NEXT(link[curr]);
        consumer.accept(new ABSTRACT_MAP.BasicEntry KEY_VALUE_GENERIC(key[curr], value[curr]));
    }
}

/** {@inheritDoc} */
@Override
public void fastForEach(final Consumer<? super MAP.Entry KEY_VALUE_GENERIC> consumer) {
    final ABSTRACT_MAP.BasicEntry KEY_VALUE_GENERIC entry = new ABSTRACT_MAP.BasicEntry
KEY_VALUE_GENERIC_DIAMOND();
    for(int i = size, curr, next = first; i-- != 0;) {
        curr = next;
        next = GET_NEXT(link[curr]);
        entry.key = key[curr];
        entry.value = value[curr];
        consumer.accept(entry);
    }
}

#else

/** {@inheritDoc} */
@Override
public void forEach(final Consumer<? super MAP.Entry KEY_VALUE_GENERIC> consumer) {
    if (containsNullKey) consumer.accept(new ABSTRACT_MAP.BasicEntry KEY_VALUE_GENERIC(key[n],
value[n]));
    for(int pos = n; pos-- != 0;)
        if (! KEY_IS_NULL(key[pos])) consumer.accept(new ABSTRACT_MAP.BasicEntry
KEY_VALUE_GENERIC(key[pos], value[pos]));
}

/** {@inheritDoc} */
@Override
public void fastForEach(final Consumer<? super MAP.Entry KEY_VALUE_GENERIC> consumer) {
    final ABSTRACT_MAP.BasicEntry KEY_VALUE_GENERIC entry = new ABSTRACT_MAP.BasicEntry
KEY_VALUE_GENERIC_DIAMOND();
    if (containsNullKey) {
        entry.key = key[n];
        entry.value = value[n];
        consumer.accept(entry);
    }
    for(int pos = n; pos-- != 0;)
        if (! KEY_IS_NULL(key[pos])) {
            entry.key = key[pos];

```

```

        entry.value = value[pos];
        consumer.accept(entry);
    }
}

#endif

}

#ifdef Linked
@Override
public FastSortedEntrySet KEY_VALUE_GENERIC ENTRYSET() {
    if (entries == null) entries = new MapEntrySet();
#else
@Override
public FastEntrySet KEY_VALUE_GENERIC ENTRYSET() {
    if (entries == null) entries = new MapEntrySet();
#endif
    return entries;
}

/** An iterator on keys.
 *
 * <p>We simply override the {@link java.util.ListIterator#next()}/{@link java.util.ListIterator#previous()}
methods
 * (and possibly their type-specific counterparts) so that they return keys
 * instead of entries.
 */

#ifdef Linked
private final class KeyIterator extends MapIterator implements KEY_LIST_ITERATOR KEY_GENERIC {
    public KeyIterator(final KEY_GENERIC_TYPE k) { super(k); }

    @Override
    public KEY_GENERIC_TYPE PREV_KEY() { return key[previousEntry()]; }

#else
private final class KeyIterator extends MapIterator implements KEY_ITERATOR KEY_GENERIC {
#endif
    public KeyIterator() { super(); }

    @Override
    public KEY_GENERIC_TYPE NEXT_KEY() { return key[nextEntry()]; }
}

#endif

```

```

private final class KeySet extends ABSTRACT_SORTED_SET KEY_GENERIC {

    @Override
    public KEY_LIST_ITERATOR KEY_GENERIC iterator(final KEY_GENERIC_TYPE from) { return new
KeyIterator(from); }

    @Override
    public KEY_LIST_ITERATOR KEY_GENERIC iterator() { return new KeyIterator(); }
#else
private final class KeySet extends ABSTRACT_SET KEY_GENERIC {

    @Override
    public KEY_ITERATOR KEY_GENERIC iterator() { return new KeyIterator(); }
#endif

    /** {@inheritDoc} */
    @Override
#ifdef JDK_PRIMITIVE_KEY_CONSUMER
    public void forEach(final JDK_PRIMITIVE_KEY_CONSUMER consumer) {
#else
    public void forEach(final KEY_CONSUMER KEY_SUPER_GENERIC consumer) {
#endif
        if (containsNullKey) consumer.accept(key[n]);
        for(int pos = n; pos-- != 0;) {
            final KEY_GENERIC_TYPE k = key[pos];
            if (! KEY_IS_NULL(k)) consumer.accept(k);
        }
    }

    @Override
    public int size() { return size; }

    @Override
    public boolean contains(KEY_TYPE k) { return containsKey(k); }

    @Override
    public boolean remove(KEY_TYPE k) {
        final int oldSize = size;
        OPEN_HASH_MAP.this.REMOVE_VALUE(k);
        return size != oldSize;
    }

    @Override
    public void clear() { OPEN_HASH_MAP.this.clear();}

#ifdef Linked
    @Override
    public KEY_GENERIC_TYPE FIRST() {

```

```

    if (size == 0) throw new NoSuchElementException();
    return key[first];
}

@Override
public KEY_GENERIC_TYPE LAST() {
    if (size == 0) throw new NoSuchElementException();
    return key[last];
}

@Override
public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() { return null; }

@Override
public SORTED_SET KEY_GENERIC tailSet(KEY_GENERIC_TYPE from) { throw new
UnsupportedOperationException(); }

@Override
public SORTED_SET KEY_GENERIC headSet(KEY_GENERIC_TYPE to) { throw new
UnsupportedOperationException(); }

@Override
public SORTED_SET KEY_GENERIC subSet(KEY_GENERIC_TYPE from, KEY_GENERIC_TYPE to) { throw
new UnsupportedOperationException(); }
#endif
}

#ifndef Linked
@Override
public SORTED_SET KEY_GENERIC keySet() {
#else
@Override
public SET KEY_GENERIC keySet() {
#endif
    if (keys == null) keys = new KeySet();
    return keys;
}

/** An iterator on values.
 *
 * <p>We simply override the { @link java.util.ListIterator#next()}/{ @link java.util.ListIterator#previous()}
methods
 * (and possibly their type-specific counterparts) so that they return values
 * instead of entries.
 */

```

```

#ifdef Linked
private final class ValueIterator extends MapIterator implements VALUE_LIST_ITERATOR VALUE_GENERIC {
    @Override
    public VALUE_GENERIC_TYPE PREV_VALUE() { return value[previousEntry()]; }

#else
private final class ValueIterator extends MapIterator implements VALUE_ITERATOR VALUE_GENERIC {
#endif
public ValueIterator() { super(); }

    @Override
    public VALUE_GENERIC_TYPE NEXT_VALUE() { return value[nextEntry()]; }
}

    @Override
public VALUE_COLLECTION VALUE_GENERIC values() {
    if (values == null) values = new VALUE_ABSTRACT_COLLECTION VALUE_GENERIC() {
        @Override
        public VALUE_ITERATOR VALUE_GENERIC iterator() { return new ValueIterator(); }
        @Override
        public int size() { return size; }
        @Override
        public boolean contains(VALUE_TYPE v) { return containsValue(v); }
        @Override
        public void clear() { OPEN_HASH_MAP.this.clear(); }

        /** {@inheritDoc} */
        @Override
#ifdef JDK_PRIMITIVE_VALUE_CONSUMER
        public void forEach(final JDK_PRIMITIVE_VALUE_CONSUMER consumer) {
#else
        public void forEach(final VALUE_CONSUMER VALUE_SUPER_GENERIC consumer) {
#endif
            if (containsNullKey) consumer.accept(value[n]);
            for(int pos = n; pos-- != 0;)
                if (! KEY_IS_NULL(key[pos])) consumer.accept(value[pos]);
            }

        };

    return values;
}

/** Reshapes the map, making the table as small as possible.
 *
 * <p>This method reshapes the table to the smallest size satisfying the
 * load factor. It can be used when the set will not be changed anymore, so

```

```

* to optimize access speed and size.
*
* <p>If the table size is already the minimum possible, this method
* does nothing.
*
* @return true if there was enough memory to trim the map.
* @see #trim(int)
*/

public boolean trim() {
    final int l = arraySize(size, f);
    if (l >= n || size > maxFill(l, f)) return true;
    try {
        rehash(l);
    }
    catch(OutOfMemoryError cantDoIt) { return false; }
    return true;
}

/** Reshapes this map if the table is too large.
*
* <p>Let <var>N</var> be the smallest table size that can hold
* <code>max(n,{ @link #size()})</code> entries, still satisfying the load factor. If the current
* table size is smaller than or equal to <var>N</var>, this method does
* nothing. Otherwise, it reshapes this map in a table of size
* <var>N</var>.
*
* <p>This method is useful when reusing maps. { @linkplain #clear() Clearing a
* map} leaves the table size untouched. If you are reusing a map
* many times, you can call this method with a typical
* size to avoid keeping around a very large table just
* because of a few large transient maps.
*
* @param n the threshold for the trimming.
* @return true if there was enough memory to trim the map.
* @see #trim()
*/

public boolean trim(final int n) {
    final int l = HashCommon.nextPowerOfTwo((int)Math.ceil(n / f));
    if (l >= n || size > maxFill(l, f)) return true;
    try {
        rehash(l);
    }
    catch(OutOfMemoryError cantDoIt) { return false; }
    return true;
}

```

```

/** Rehashes the map.
 *
 * <p>This method implements the basic rehashing strategy, and may be
 * overridden by subclasses implementing different rehashing strategies (e.g.,
 * disk-based rehashing). However, you should not override this method
 * unless you understand the internal workings of this class.
 *
 * @param newN the new size
 */

SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
protected void rehash(final int newN) {
    final KEY_GENERIC_TYPE key[] = this.key;
    final VALUE_GENERIC_TYPE value[] = this.value;

    final int mask = newN - 1; // Note that this is used by the hashing macro
    final KEY_GENERIC_TYPE newKey[] = KEY_GENERIC_ARRAY_CAST new KEY_TYPE[newN + 1];
    final VALUE_GENERIC_TYPE newValue[] = VALUE_GENERIC_ARRAY_CAST new VALUE_TYPE[newN +
1];

#ifdef Linked
    int i = first, prev = -1, newPrev = -1, t, pos;
    final long link[] = this.link;
    final long newLink[] = new long[newN + 1];
    first = -1;

    for(int j = size; j-- != 0;) {
        if (KEY_EQUALS_NULL(key[i])) pos = newN;
        else {
            pos = KEY2INTHASH(key[i]) & mask;
            while (! KEY_IS_NULL(newKey[pos])) pos = (pos + 1) & mask;
        }

        newKey[pos] = key[i];
        newValue[pos] = value[i];

        if (prev != -1) {
            SET_NEXT(newLink[newPrev], pos);
            SET_PREV(newLink[pos], newPrev);
            newPrev = pos;
        }
        else {
            newPrev = first = pos;
            // Special case of SET(newLink[pos], -1, -1);
            newLink[pos] = -1L;
        }
    }

```

```

    t = i;
    i = GET_NEXT(link[i]);
    prev = t;
}

this.link = newLink;
this.last = newPrev;
if (newPrev != -1)
    // Special case of SET_NEXT(newLink[newPrev], -1);
    newLink[newPrev] |= -1 & 0xFFFFFFFFL;
#else
int i = n, pos;

for(int j = realSize(); j-- != 0;) {
    while(KEY_IS_NULL(key[--i]));

    if (! KEY_IS_NULL(newKey[pos = KEY2INTHASH(key[i] & mask)])
        while (! KEY_IS_NULL(newKey[pos = (pos + 1) & mask]));

    newKey[pos] = key[i];
    newValue[pos] = value[i];
}

newValue[newN] = value[n];

#endif

n = newN;
this.mask = mask;
maxFill = maxFill(n, f);
this.key = newKey;
this.value = newValue;
}

/** Returns a deep copy of this map.
 *
 * <p>This method performs a deep copy of this hash map; the data stored in the
 * map, however, is not cloned. Note that this makes a difference only for object keys.
 *
 * @return a deep copy of this map.
 */
@Override
SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
public OPEN_HASH_MAP KEY_VALUE_GENERIC clone() {
    OPEN_HASH_MAP KEY_VALUE_GENERIC c;
    try {
        c = (OPEN_HASH_MAP KEY_VALUE_GENERIC)super.clone();
    }

```

```

    }
    catch(CloneNotSupportedException cantHappen) {
        throw new InternalError();
    }

    c.keys = null;
    c.values = null;
    c.entries = null;
    c.containsNullKey = containsNullKey;

    c.key = key.clone();
    c.value = value.clone();
#ifdef Linked
    c.link = link.clone();
#endif
#ifdef Custom
    c.strategy = strategy;
#endif
    return c;
}

/** Returns a hash code for this map.
 *
 * This method overrides the generic method provided by the superclass.
 * Since { @code equals()} is not overridden, it is important
 * that the value returned by this method is the same value as
 * the one returned by the overridden method.
 *
 * @return a hash code for this map.
 */

@Override
public int hashCode() {
    int h = 0;
    for(int j = realSize(), i = 0, t = 0; j-- != 0;) {
        while(KEY_IS_NULL(key[i])) i++;
#ifdef KEYS_REFERENCE
        if (this != key[i])
#endif
            t = KEY2JAVAHASH_NOT_NULL(key[i]);
#ifdef VALUES_REFERENCE
        if (this != value[i])
#endif
            t ^= VALUE2JAVAHASH(value[i]);
        h += t;
        i++;
    }
}

```

```

// Zero / null keys have hash zero.
if (containsNullKey) h += VALUE2JAVAHASH(value[n]);
return h;
}

```

```

private void writeObject(java.io.ObjectOutputStream s) throws java.io.IOException {
    final KEY_GENERIC_TYPE key[] = this.key;
    final VALUE_GENERIC_TYPE value[] = this.value;
    final MapIterator i = new MapIterator();

    s.defaultWriteObject();

    for(int j = size, e; j-- != 0;) {
        e = i.nextEntry();
        s.WRITE_KEY(key[e]);
        s.WRITE_VALUE(value[e]);
    }
}

```

SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED

```

private void readObject(java.io.ObjectInputStream s) throws java.io.IOException, ClassNotFoundException {
    s.defaultReadObject();

```

```

    n = arraySize(size, f);
    maxFill = maxFill(n, f);
    mask = n - 1;

```

```

    final KEY_GENERIC_TYPE key[] = this.key = KEY_GENERIC_ARRAY_CAST new KEY_TYPE[n + 1];
    final VALUE_GENERIC_TYPE value[] = this.value = VALUE_GENERIC_ARRAY_CAST new
VALUE_TYPE[n + 1];

```

```

#ifdef Linked

```

```

    final long link[] = this.link = new long[n + 1];
    int prev = -1;
    first = last = -1;

```

```

#endif

```

```

    KEY_GENERIC_TYPE k;
    VALUE_GENERIC_TYPE v;

```

```

    for(int i = size, pos; i-- != 0;) {

```

```

        k = KEY_GENERIC_CAST s.READ_KEY();
        v = VALUE_GENERIC_CAST s.READ_VALUE();

```

```

if (KEY_EQUALS_NULL(k)) {
    pos = n;
    containsNullKey = true;
}
else {
    pos = KEY2INTHASH(k) & mask;
    while (! KEY_IS_NULL(key[pos])) pos = (pos + 1) & mask;
}

key[pos] = k;
value[pos] = v;

#ifdef Linked
if (first != -1) {
    SET_NEXT(link[prev], pos);
    SET_PREV(link[pos], prev);
    prev = pos;
}
else {
    prev = first = pos;
    // Special case of SET_PREV(newLink[pos], -1);
    link[pos] |= (-1L & 0xFFFFFFFFL) << 32;
}
#endif

#ifdef Linked
last = prev;
if (prev != -1)
    // Special case of SET_NEXT(link[prev], -1);
    link[prev] |= -1 & 0xFFFFFFFFL;
#endif

if (ASSERTS) checkTable();
}

#ifdef ASSERTS_CODE
private void checkTable() {
    assert (n & -n) == n : "Table length is not a power of two: " + n;
    assert n == key.length - 1;
    int n = key.length - 1;
    while(n-- != 0)
        if (! KEY_IS_NULL(key[n]) && ! containsKey(key[n]))
            throw new AssertionError("Hash table has key " + key[n] + " marked as occupied, but the key does not belong to the table");
}

```

```

#if KEYS_PRIMITIVE
    java.util.HashSet<KEY_GENERIC_CLASS> s = new java.util.HashSet<KEY_GENERIC_CLASS> ();
#else
    java.util.HashSet<Object> s = new java.util.HashSet<Object>();
#endif

    for(int i = key.length; i-- != 0;)
        if (! KEY_IS_NULL(key[i]) && ! s.add(key[i])) throw new AssertionError("Key " + key[i] + " appears twice at
position " + i);

#ifdef Linked
    KEY_BIDI_ITERATOR KEY_GENERIC i = keySet().iterator();
    KEY_GENERIC_TYPE k;
    n = size();
    while(n-- != 0)
        if (! containsKey(k = i.NEXT_KEY()))
            throw new AssertionError("Linked hash table forward enumerates key " + k + ", but the key does not belong to the
table");

    if (i.hasNext()) throw new AssertionError("Forward iterator not exhausted");

    n = size();
    if (n > 0) {
        i = keySet().iterator(LAST_KEY());
        while(n-- != 0)
            if (! containsKey(k = i.PREV_KEY()))
                throw new AssertionError("Linked hash table backward enumerates key " + k + ", but the key does not belong to the
table");

        if (i.hasPrevious()) throw new AssertionError("Previous iterator not exhausted");
    }
#endif
}
#else
private void checkTable() {}
#endif

#ifdef TEST

private static long seed = System.currentTimeMillis();
private static java.util.Random r = new java.util.Random(seed);

private static KEY_TYPE genKey() {
#ifdef KEY_CLASS_Byte || KEY_CLASS_Short || KEY_CLASS_Character
    return (KEY_TYPE)(r.nextInt());
#elif KEYS_PRIMITIVE
    return r.NEXT_KEY();
#endif
}

```

```

#elif !KEY_CLASS_Reference
#ifdef Custom
    int i = r.nextInt(3);
    byte a[] = new byte[i];
    while(i-- != 0) a[i] = (byte)r.nextInt();
    return a;
#else
    return Integer.toBinaryString(r.nextInt());
#endif
#else
    return new java.io.Serializable() {};
#endif
}

private static VALUE_TYPE genValue() {
#if VALUE_CLASS_Byte || VALUE_CLASS_Short || VALUE_CLASS_Character
    return (VALUE_TYPE)(r.nextInt());
#elif VALUES_PRIMITIVE
    return r.NEXT_VALUE();
#elif !VALUE_CLASS_Reference
    return Integer.toBinaryString(r.nextInt());
#else
    return new java.io.Serializable() {};
#endif
}

private static final class ArrayComparator implements java.util.Comparator {
public int compare(Object a, Object b) {
    byte[] aa = (byte[])a;
    byte[] bb = (byte[])b;
    int length = Math.min(aa.length, bb.length);
    for(int i = 0; i < length; i++) {
        if (aa[i] < bb[i]) return -1;
        if (aa[i] > bb[i]) return 1;
    }
    return aa.length == bb.length ? 0 : (aa.length < bb.length ? -1 : 1);
}
}

private static final class MockMap extends java.util.TreeMap {
private java.util.List list = new java.util.ArrayList();

public MockMap(java.util.Comparator c) { super(c); }

public Object put(Object k, Object v) {
    if (!containsKey(k)) list.add(k);
    return super.put(k, v);
}
}

```

```

public void putAll(Map m) {
    java.util.Iterator i = m.entrySet().iterator();
    while(i.hasNext()) {
        Map.Entry e = (Map.Entry)i.next();
        put(e.getKey(), e.getValue());
    }
}

public Object remove(Object k) {
    if (containsKey(k)) {
        int i = list.size();
        while(i-- != 0) if (comparator().compare(list.get(i), k) == 0) {
            list.remove(i);
            break;
        }
    }
    return super.remove(k);
}

private void justRemove(Object k) { super.remove(k); }
private java.util.Set justEntrySet() { return super.entrySet(); }
private java.util.Set justKeySet() { return super.keySet(); }

public java.util.Set keySet() {
    return new java.util.AbstractSet() {
        final java.util.Set keySet = justKeySet();

        public boolean contains(Object k) { return keySet.contains(k); }
        public int size() { return keySet.size(); }
        public java.util.Iterator iterator() {
            return new java.util.Iterator() {
                final java.util.Iterator iterator = list.iterator();
                Object curr;
                public Object next() { return curr = iterator.next(); }
                public boolean hasNext() { return iterator.hasNext(); }
                public void remove() {
                    justRemove(curr);
                    iterator.remove();
                }
            };
        }
    };
}

public java.util.Set entrySet() {

```

```

return new java.util.AbstractSet() {
    final java.util.Set entrySet = justEntrySet();

    public boolean contains(Object k) { return entrySet.contains(k); }
    public int size() { return entrySet.size(); }
    public java.util.Iterator iterator() {
        return new java.util.Iterator() {
            final java.util.Iterator iterator = list.iterator();
            Object curr;
            public Object next() {
                curr = iterator.next();
            }
        };
    }
};

#if VALUE_CLASS_Reference
#if KEY_CLASS_Reference
    return new ABSTRACT_MAP.BasicEntry((Object)curr, (Object)get(curr)) {
#else
    return new ABSTRACT_MAP.BasicEntry((KEY_CLASS)curr, (Object)get(curr)) {
#endif
#else
#if KEY_CLASS_Reference
    return new ABSTRACT_MAP.BasicEntry((Object)curr, (VALUE_CLASS)get(curr)) {
#else
    return new ABSTRACT_MAP.BasicEntry((KEY_CLASS)curr, (VALUE_CLASS)get(curr)) {
#endif
#endif
        public VALUE_TYPE setValue(VALUE_TYPE v) {
            return VALUE_OBJ2TYPE(put(getKey(), VALUE2OBJ(v)));
        }
    };
}

public boolean hasNext() { return iterator.hasNext(); }
public void remove() {
    justRemove(((Map.Entry)curr).getKey());
    iterator.remove();
}
};
}
};
}
}

private static java.text.NumberFormat format = new java.text.DecimalFormat("#,###.00");
private static java.text.FieldPosition fp = new java.text.FieldPosition(0);

private static String format(double d) {
    StringBuffer s = new StringBuffer();
    return format.format(d, s, fp).toString();
}
}

```

```

private static void speedTest(int n, float f, boolean comp) {
#ifdef Custom
    int i, j;
    OPEN_HASH_MAP m;
#ifdef Linked
    java.util.LinkedHashMap t;
#else
    java.util.HashMap t;
#endif
    KEY_TYPE k[] = new KEY_TYPE[n];
    KEY_TYPE nk[] = new KEY_TYPE[n];
    VALUE_TYPE v[] = new VALUE_TYPE[n];
    long ns;

    for(i = 0; i < n; i++) {
        k[i] = genKey();
        nk[i] = genKey();
        v[i] = genValue();
    }

    double totPut = 0, totYes = 0, totNo = 0, totIter = 0, totRemYes = 0, totRemNo = 0, d;

    if (comp) { for(j = 0; j < 20; j++) {

#ifdef Linked
        t = new java.util.LinkedHashMap(16);
#else
        t = new java.util.HashMap(16);
#endif

        /* We put pairs to t. */
        ns = System.nanoTime();
        for(i = 0; i < n; i++) t.put(KEY2OBJ(k[i]), VALUE2OBJ(v[i]));
        d = (System.nanoTime() - ns) / (double)n;
        if (j > 2) totPut += d;
        System.out.print("Put: " + format(d) + "ns ");

        /* We check for pairs in t. */
        ns = System.nanoTime();
        for(i = 0; i < n; i++) t.containsKey(KEY2OBJ(k[i]));
        d = (System.nanoTime() - ns) / (double)n;
        if (j > 2) totYes += d;
        System.out.print("Yes: " + format(d) + "ns ");

        /* We check for pairs not in t. */
        ns = System.nanoTime();
        for(i = 0; i < n; i++) t.containsKey(KEY2OBJ(nk[i]));

```

```

d = (System.nanoTime() - ns) / (double)n;
if (j > 2) totNo += d;
System.out.print("No: " + format(d) + "ns ");

/* We iterate on t. */
ns = System.nanoTime();
for(java.util.Iterator it = t.entrySet().iterator(); it.hasNext(); it.next());
d = (System.nanoTime() - ns) / (double)n;
if (j > 2) totIter += d;
System.out.print("Iter: " + format(d) + "ns ");

/* We delete pairs not in t. */
ns = System.nanoTime();
for(i = 0; i < n; i++) t.remove(KEY2OBJ(nk[i]));
d = (System.nanoTime() - ns) / (double)n;
if (j > 2) totRemNo += d;
System.out.print("RemNo: " + format(d) + "ns ");

/* We delete pairs in t. */
ns = System.nanoTime();
for(i = 0; i < n; i++) t.remove(KEY2OBJ(k[i]));
d = (System.nanoTime() - ns) / (double)n;
if (j > 2) totRemYes += d;
System.out.print("RemYes: " + format(d) + "ns ");

System.out.println();
}

System.out.println();
System.out.println("java.util Put: " + format(totPut/(j-3)) + "ns Yes: " + format(totYes/(j-3)) + "ns No: " +
format(totNo/(j-3)) + "ns Iter: " + format(totIter/(j-3)) + "ns RemNo: " + format(totRemNo/(j-3)) + "ns RemYes: " +
format(totRemYes/(j-3)) + "K/s");

System.out.println();

totPut = totYes = totNo = totIter = totRemYes = totRemNo = 0;

}

for(j = 0; j < 20; j++) {

m = new OPEN_HASH_MAP(16, f);

/* We put pairs to m. */
ns = System.nanoTime();
for(i = 0; i < n; i++) m.put(k[i], v[i]);
d = (System.nanoTime() - ns) / (double)n;
if (j > 2) totPut += d;

```

```

System.out.print("Put: " + format(d) + "ns ");

/* We check for pairs in m. */
ns = System.nanoTime();
for(i = 0; i < n; i++) m.containsKey(k[i]);
d = (System.nanoTime() - ns) / (double)n;
if (j > 2) totYes += d;
System.out.print("Yes: " + format(d) + "ns ");

/* We check for pairs not in m. */
ns = System.nanoTime();
for(i = 0; i < n; i++) m.containsKey(nk[i]);
d = (System.nanoTime() - ns) / (double)n;
if (j > 2) totNo += d;
System.out.print("No: " + format(d) + "ns ");

/* We iterate on m. */
ns = System.nanoTime();
for(java.util.Iterator it = m.entrySet().iterator(); it.hasNext(); it.next());
d = (System.nanoTime() - ns) / (double)n;
if (j > 2) totIter += d;
System.out.print("Iter: " + format(d) + "ns ");

/* We delete pairs not in m. */
ns = System.nanoTime();
for(i = 0; i < n; i++) m.remove(nk[i]);
d = (System.nanoTime() - ns) / (double)n;
if (j > 2) totRemNo += d;
System.out.print("RemNo: " + format(d) + "ns ");

/* We delete pairs in m. */
ns = System.nanoTime();
for(i = 0; i < n; i++) m.remove(k[i]);
d = (System.nanoTime() - ns) / (double)n;
if (j > 2) totRemYes += d;
System.out.print("RemYes: " + format(d) + "ns ");

System.out.println();
}

System.out.println();
System.out.println("fastutil Put: " + format(totPut/(j-3)) + "ns Yes: " + format(totYes/(j-3)) + "ns No: " +
format(totNo/(j-3)) + "ns Iter: " + format(totIter/(j-3)) + "ns RemNo: " + format(totRemNo/(j-3)) + "ns RemYes: " +
format(totRemYes/(j-3)) + "ns");

System.out.println();
#endif

```

```

}

private static boolean valEquals(Object o1, Object o2) {
    return o1 == null ? o2 == null : o1.equals(o2);
}

private static void fatal(String msg) {
    System.out.println(msg);
    System.exit(1);
}

private static void ensure(boolean cond, String msg) {
    if (cond) return;
    fatal(msg);
}

protected static void runTest(int n, float f) {
    #if !defined(Custom) || KEYS_REFERENCE

    #ifdef Custom
        OPEN_HASH_MAP m = new OPEN_HASH_MAP(Hash.DEFAULT_INITIAL_SIZE, f,
it.unimi.dsi.fastutil.bytes.ByteArrays.HASH_STRATEGY);
    #else
        OPEN_HASH_MAP m = new OPEN_HASH_MAP(Hash.DEFAULT_INITIAL_SIZE, f);
    #endif

    #ifdef Linked
    #ifdef Custom
        Map t = new MockMap(new ArrayComparator());
    #else
        Map t = new java.util.LinkedHashMap();
    #endif
    #else
    #ifdef Custom
        Map t = new java.util.TreeMap(new ArrayComparator());
    #else
        Map t = new java.util.HashMap();
    #endif
    #endif

    /* First of all, we fill t with random data. */

    for(int i=0; i<n; i++) t.put(KEY2OBJ(genKey()), VALUE2OBJ(genValue()));

    /* Now we add to m the same data */

    m.putAll(t);

```

```

if (!m.equals(t)) System.out.println("Error (" + seed + "): !m.equals(t) after insertion");
if (!t.equals(m)) System.out.println("Error (" + seed + "): !t.equals(m) after insertion");

/* Now we check that m actually holds that data. */

for(java.util.Iterator i=t.entrySet().iterator(); i.hasNext();) {
    java.util.Map.Entry e = (java.util.Map.Entry)i.next();
    if (!valEquals(e.getValue(), m.get(e.getKey())))
        System.out.println("Error (" + seed + "): m and t differ on an entry (" + e + ") after insertion (iterating on t)");
}

/* Now we check that m actually holds that data, but iterating on m. */

for(java.util.Iterator i=m.entrySet().iterator(); i.hasNext();) {
    java.util.Map.Entry e = (java.util.Map.Entry)i.next();
    if (!valEquals(e.getValue(), t.get(e.getKey())))
        System.out.println("Error (" + seed + "): m and t differ on an entry (" + e + ") after insertion (iterating on m)");
}

/* Now we check that m actually holds the same keys. */

for(java.util.Iterator i=t.keySet().iterator(); i.hasNext();) {
    Object o = i.next();
    if (!m.containsKey(o)) {
        System.out.println("Error (" + seed + "): m and t differ on a key (" + o + ") after insertion (iterating on t)");
        System.exit(1);
    }
    if (!m.keySet().contains(o)) {
        System.out.println("Error (" + seed + "): m and t differ on a key (" + o + ", in keySet()) after insertion (iterating on t)");
        System.exit(1);
    }
}

/* Now we check that m actually holds the same keys, but iterating on m. */

for(java.util.Iterator i=m.keySet().iterator(); i.hasNext();) {
    Object o = i.next();
    if (!t.containsKey(o)) {
        System.out.println("Error (" + seed + "): m and t differ on a key after insertion (iterating on m)");
        System.exit(1);
    }
    if (!t.keySet().contains(o)) {
        System.out.println("Error (" + seed + "): m and t differ on a key (in keySet()) after insertion (iterating on m)");
        System.exit(1);
    }
}

```

```

/* Now we check that m actually hold the same values. */

for(java.util.Iterator i=t.values().iterator(); i.hasNext();) {
    Object o = i.next();
    if (!m.containsValue(o)) {
        System.out.println("Error (" + seed + "): m and t differ on a value after insertion (iterating on t)");
        System.exit(1);
    }
    if (!m.values().contains(o)) {
        System.out.println("Error (" + seed + "): m and t differ on a value (in values()) after insertion (iterating on t)");
        System.exit(1);
    }
}

/* Now we check that m actually hold the same values, but iterating on m. */

for(java.util.Iterator i=m.values().iterator(); i.hasNext();) {
    Object o = i.next();
    if (!t.containsValue(o)) {
        System.out.println("Error (" + seed + "): m and t differ on a value after insertion (iterating on m)");
        System.exit(1);
    }
    if (!t.values().contains(o)) {
        System.out.println("Error (" + seed + "): m and t differ on a value (in values()) after insertion (iterating on m)");
        System.exit(1);
    }
}

/* Now we check that inquiries about random data give the same answer in m and t. For
   m we use the polymorphic method. */

for(int i=0; i<n; i++) {
    KEY_TYPE T = genKey();
    if (m.containsKey(KEY2OBJ(T)) != t.containsKey(KEY2OBJ(T))) {
        System.out.println("Error (" + seed + "): divergence in keys between t and m (polymorphic method)");
        System.exit(1);
    }
}

#if (KEYS_REFERENCE) && ! (VALUES_REFERENCE)
    if ((m.GET_VALUE(T) != VALUE_NULL) != ((t.get(KEY2OBJ(T)) == null ? VALUE_NULL :
VALUE_OBJ2TYPE(t.get(KEY2OBJ(T)))) != VALUE_NULL) ||
        t.get(KEY2OBJ(T)) != null &&
        ! VALUE2OBJ(m.GET_VALUE(T)).equals(t.get(KEY2OBJ(T))))
#else
    if ((m.get(T) != VALUE_NULL) != ((t.get(KEY2OBJ(T)) == null ? VALUE_NULL :
VALUE_OBJ2TYPE(t.get(KEY2OBJ(T)))) != VALUE_NULL) ||
        t.get(KEY2OBJ(T)) != null &&
        ! m.get(KEY2OBJ(T)).equals(t.get(KEY2OBJ(T))))

```

```

#endif
{
    System.out.println("Error (" + seed + "): divergence between t and m (polymorphic method)");
    System.exit(1);
}
}

/* Again, we check that inquiries about random data give the same answer in m and t, but
for m we use the standard method. */

for(int i=0; i<n; i++) {
    KEY_TYPE T = genKey();
    if (!valEquals(m.get(KEY2OBJ(T)), t.get(KEY2OBJ(T)))) {
        System.out.println("Error (" + seed + "): divergence between t and m (standard method)");
        System.exit(1);
    }
}

/* Now we put and remove random data in m and t, checking that the result is the same. */

for(int i=0; i<20*n; i++) {
    KEY_TYPE T = genKey();
    VALUE_TYPE U = genValue();
    if (!valEquals(m.put(KEY2OBJ(T), VALUE2OBJ(U)), t.put(KEY2OBJ(T), VALUE2OBJ(U)))) {
        System.out.println("Error (" + seed + "): divergence in put() between t and m");
        System.exit(1);
    }
    T = genKey();
    if (!valEquals(m.remove(KEY2OBJ(T)), t.remove(KEY2OBJ(T)))) {
        System.out.println("Error (" + seed + "): divergence in remove() between t and m");
        System.exit(1);
    }
}

if (!m.equals(t)) System.out.println("Error (" + seed + "): !m.equals(t) after removal");
if (!t.equals(m)) System.out.println("Error (" + seed + "): !t.equals(m) after removal");

/* Now we check that m actually holds the same data. */

for(java.util.Iterator i=t.entrySet().iterator(); i.hasNext();) {
    java.util.Map.Entry e = (java.util.Map.Entry)i.next();
    if (!valEquals(e.getValue(), m.get(e.getKey()))) {
        System.out.println("Error (" + seed + "): m and t differ on an entry (" + e + ") after removal (iterating on t)");
        System.exit(1);
    }
}
}

```

```

/* Now we check that m actually holds that data, but iterating on m. */

for(java.util.Iterator i=m.entrySet().iterator(); i.hasNext();) {
    java.util.Map.Entry e = (java.util.Map.Entry)i.next();
    if (!valEquals(e.getValue(), t.get(e.getKey()))) {
        System.out.println("Error (" + seed + "): m and t differ on an entry (" + e + ") after removal (iterating on m)");
        System.exit(1);
    }
}

/* Now we check that m actually holds the same keys. */

for(java.util.Iterator i=t.keySet().iterator(); i.hasNext();) {
    Object o = i.next();
    if (!m.containsKey(o)) {
        System.out.println("Error (" + seed + "): m and t differ on a key (" + o + ") after removal (iterating on t)");
        System.exit(1);
    }
    if (!m.keySet().contains(o)) {
        System.out.println("Error (" + seed + "): m and t differ on a key (" + o + ", in keySet()) after removal (iterating on t)");
        System.exit(1);
    }
}

/* Now we check that m actually holds the same keys, but iterating on m. */

for(java.util.Iterator i=m.keySet().iterator(); i.hasNext();) {
    Object o = i.next();
    if (!t.containsKey(o)) {
        System.out.println("Error (" + seed + "): m and t differ on a key after removal (iterating on m)");
        System.exit(1);
    }
    if (!t.keySet().contains(o)) {
        System.out.println("Error (" + seed + "): m and t differ on a key (in keySet()) after removal (iterating on m)");
        System.exit(1);
    }
}

/* Now we check that m actually hold the same values. */

for(java.util.Iterator i=t.values().iterator(); i.hasNext();) {
    Object o = i.next();
    if (!m.containsValue(o)) {
        System.out.println("Error (" + seed + "): m and t differ on a value after removal (iterating on t)");
        System.exit(1);
    }
    if (!m.values().contains(o)) {

```

```

    System.out.println("Error (" + seed + "): m and t differ on a value (in values()) after removal (iterating on t)");
    System.exit(1);
}
}

/* Now we check that m actually hold the same values, but iterating on m. */

for(java.util.Iterator i=m.values().iterator(); i.hasNext();) {
    Object o = i.next();
    if (!t.containsValue(o)) {
        System.out.println("Error (" + seed + "): m and t differ on a value after removal (iterating on m)");
        System.exit(1);
    }
    if (!t.values().contains(o)) {
        System.out.println("Error (" + seed + "): m and t differ on a value (in values()) after removal (iterating on m)");
        System.exit(1);
    }
}

int h = m.hashCode();

/* Now we save and read m. */

try {
    java.io.File ff = new java.io.File("it.unimi.dsi.fastutil.test");
    java.io.OutputStream os = new java.io.FileOutputStream(ff);
    java.io.ObjectOutputStream oos = new java.io.ObjectOutputStream(os);

    oos.writeObject(m);
    oos.close();

    java.io.InputStream is = new java.io.FileInputStream(ff);
    java.io.ObjectInputStream ois = new java.io.ObjectInputStream(is);

    m = (OPEN_HASH_MAP)ois.readObject();
    ois.close();
    ff.delete();
}
catch(Exception e) {
    e.printStackTrace();
    System.exit(1);
}

#if !KEY_CLASS_Reference && !VALUE_CLASS_Reference
    if (m.hashCode() != h) System.out.println("Error (" + seed + "): hashCode() changed after save/read");

```

```

/* Now we check that m actually holds that data. */

for(java.util.Iterator i=t.keySet().iterator(); i.hasNext();) {
    Object o = i.next();
    if (!valEquals(m.get(o),t.get(o))) {
        System.out.println("Error (" + seed + "): m and t differ on an entry after save/read");
        System.exit(1);
    }
}
#else
    m.clear();
    m.putAll(t);
#endif

/* Now we put and remove random data in m and t, checking that the result is the same. */

for(int i=0; i<20*n; i++) {
    KEY_TYPE T = genKey();
    VALUE_TYPE U = genValue();
    if (!valEquals(m.put(KEY2OBJ(T), VALUE2OBJ(U)), t.put(KEY2OBJ(T), VALUE2OBJ(U)))) {
        System.out.println("Error (" + seed + "): divergence in put() between t and m after save/read");
        System.exit(1);
    }
    T = genKey();
    if (!valEquals(m.remove(KEY2OBJ(T)), t.remove(KEY2OBJ(T)))) {
        System.out.println("Error (" + seed + "): divergence in remove() between t and m after save/read");
        System.exit(1);
    }
}

if (!m.equals(t)) System.out.println("Error (" + seed + "): !m.equals(t) after post-save/read removal");
if (!t.equals(m)) System.out.println("Error (" + seed + "): !t.equals(m) after post-save/read removal");

#ifdef Linked

/* Now we play with iterators. */

{
    java.util.ListIterator i, j;
    Object J;
    Map.Entry E, F;
    i = (java.util.ListIterator)m.entrySet().iterator();
    j = new java.util.LinkedList(t.entrySet()).listIterator();
}

```

```

for(int k = 0; k < 2*n; k++) {
    ensure(i.hasNext() == j.hasNext(), "Error (" + seed + "): divergence in hasNext()");
    ensure(i.hasPrevious() == j.hasPrevious(), "Error (" + seed + "): divergence in hasPrevious()");

    if (r.nextFloat() < .8 && i.hasNext()) {
#ifdef Custom
        ensure(m.strategy().equals((E=(java.util.Map.Entry)i.next()).getKey(), J = (F=(Map.Entry)j.next()).getKey()), "Error (" + seed + "): divergence in next()");
#else
        ensure((E=(java.util.Map.Entry)i.next()).getKey().equals(J = (F=(Map.Entry)j.next()).getKey()), "Error (" + seed + "): divergence in next()");
#endif

        if (r.nextFloat() < 0.3) {
            i.remove();
            j.remove();
            t.remove(J);
        }
        else if (r.nextFloat() < 0.3) {
            Object U = VALUE2OBJ(genValue());
            E.setValue(U);
            t.put(F.getKey(), U);
        }
        }
    else if (r.nextFloat() < .2 && i.hasPrevious()) {
#ifdef Custom
        ensure(m.strategy().equals((E=(java.util.Map.Entry)i.previous()).getKey(), J = (F=(Map.Entry)j.previous()).getKey()), "Error (" + seed + "): divergence in previous()");
#else
        ensure((E=(java.util.Map.Entry)i.previous()).getKey().equals(J = (F=(Map.Entry)j.previous()).getKey()), "Error (" + seed + "): divergence in previous()");
#endif

        if (r.nextFloat() < 0.3) {
            i.remove();
            j.remove();
            t.remove(J);
        }
        else if (r.nextFloat() < 0.3) {
            Object U = VALUE2OBJ(genValue());
            E.setValue(U);
            t.put(F.getKey(), U);
        }
        }

    ensure(i.nextIndex() == j.nextIndex(), "Error (" + seed + "): divergence in nextIndex()");
    ensure(i.previousIndex() == j.previousIndex(), "Error (" + seed + "): divergence in previousIndex()");

```

```

    }

}

if (t.size() > 0) {
    java.util.ListIterator i, j;
    Object J;
    j = new java.util.LinkedList(t.keySet()).listIterator();
    int e = r.nextInt(t.size());
    Object from;
    do from = j.next(); while(e-- != 0);

    i = (java.util.ListIterator)((SORTED_SET)m.keySet()).iterator(KEY_OBJ2TYPE(from));

    for(int k = 0; k < 2*n; k++) {
        ensure(i.hasNext() == j.hasNext(), "Error (" + seed + "): divergence in hasNext() (iterator with starting point " +
        from + ")");
        ensure(i.hasPrevious() == j.hasPrevious(), "Error (" + seed + "): divergence in hasPrevious() (iterator with starting
        point " + from + ")");

        if (r.nextFloat() < .8 && i.hasNext()) {
#ifdef Custom
            ensure(m.strategy().equals(i.next(), J = j.next()), "Error (" + seed + "): divergence in next() (iterator with starting
            point " + from + ")");
#else
            ensure(i.next().equals(J = j.next()), "Error (" + seed + "): divergence in next() (iterator with starting point " + from +
            ")");
#endif

            if (r.nextFloat() < 0.5) {
                i.remove();
                j.remove();
                t.remove(J);
            }
        }
        else if (r.nextFloat() < .2 && i.hasPrevious()) {
#ifdef Custom
            ensure(m.strategy().equals(i.previous(), J = j.previous()), "Error (" + seed + "): divergence in previous() (iterator
            with starting point " + from + ")");
#else
            ensure(i.previous().equals(J = j.previous()), "Error (" + seed + "): divergence in previous() (iterator with starting
            point " + from + ")");
#endif
        }

        if (r.nextFloat() < 0.5) {
            i.remove();
            j.remove();
            t.remove(J);
        }
    }
}

```

```

    }
}

    ensure(i.nextIndex() == j.nextIndex(), "Error (" + seed + "): divergence in nextIndex() (iterator with starting point "
+ from + ")");
    ensure(i.previousIndex() == j.previousIndex(), "Error (" + seed + "): divergence in previousIndex() (iterator with
starting point " + from + ")");

}

}

/* Now we check that m actually holds that data. */

ensure(m.equals(t), "Error (" + seed + "): ! m.equals(t) after iteration");
ensure(t.equals(m), "Error (" + seed + "): ! t.equals(m) after iteration");

#endif

/* Now we take out of m everything, and check that it is empty. */

for(java.util.Iterator i=t.keySet().iterator(); i.hasNext();) m.remove(i.next());

if (!m.isEmpty()) {
    System.out.println("Error (" + seed + "): m is not empty (as it should be)");
    System.exit(1);
}

#ifdef NumericEnhancements
#if VALUE_CLASS_Byte || VALUE_CLASS_Character || VALUE_CLASS_Short || VALUE_CLASS_Integer ||
VALUE_CLASS_Long
/* Now we check that increment works properly, using random data */

{
    t.clear();
    m.clear();

    for(int k = 0; k < 2*n; k++) {
        KEY_TYPE T = genKey();
        VALUE_TYPE U = genValue();

        VALUE_TYPE rU = m.increment(T, U);
        VALUE_GENERIC_CLASS tU = (VALUE_GENERIC_CLASS) t.get(KEY2OBJ(T));
        if (null == tU) {
            ensure(m.defaultReturnValue() == rU, "Error (" + seed + "): map increment does not return proper starting value.");

```

```

    t.put(KEY2OBJ(T), VALUE2OBJ((VALUE_TYPE) (m.defaultReturnValue() + U)));
}
else {
    t.put(KEY2OBJ(T), VALUE2OBJ((VALUE_TYPE) (((VALUE_TYPE) tU) + U)));
}
}

// Maps should contain identical values
ensure(new java.util.HashMap(m).equals(new java.util.HashMap(t)),
    "Error(" + seed + "): incremented maps are not equal.");
}
#endif
#endif

#if (KEY_CLASS_Integer || KEY_CLASS_Long) && (VALUE_CLASS_Integer || VALUE_CLASS_Long)
    m = new OPEN_HASH_MAP(n, f);
    t.clear();
    int x;

    /* Now we torture-test the hash table. This part is implemented only for integers and longs. */

    int p = m.key.length;

    for(int i=0; i<p; i++) {
        for (int j=0; j<20; j++) {
            m.put(i+(r.nextInt() % 10)*p, 1);
            m.remove(i+(r.nextInt() % 10)*p);
        }

        for (int j=-10; j<10; j++) m.remove(i+j*p);
    }

    t.putAll(m);

    /* Now all table entries are REMOVED. */

    for(int i=0; i<(p*f)/10; i++) {
        for (int j=0; j<10; j++) {
            if (!valEquals(m.put(KEY2OBJ(x = i+(r.nextInt() % 10)*p), VALUE2OBJ(1)), t.put(KEY2OBJ(x),
                VALUE2OBJ(1))))
                System.out.println("Error (" + seed + "): m and t differ on an entry during torture-test insertion.");
        }
    }

    if (!m.equals(t)) System.out.println("Error (" + seed + "): !m.equals(t) after torture-test insertion");
    if (!t.equals(m)) System.out.println("Error (" + seed + "): !t.equals(m) after torture-test insertion");

```

```

for(int i=0; i<p/10; i++) {
  for (int j=0; j<10; j++) {
    if (!valEquals(m.remove(KEY2OBJ(x = i+(r.nextInt() % 10)*p)), t.remove(KEY2OBJ(x))))
      System.out.println("Error (" + seed + "): m and t differ on an entry during torture-test removal.");
  }
}

if (!m.equals(t)) System.out.println("Error (" + seed + "): !m.equals(t) after torture-test removal");
if (!t.equals(m)) System.out.println("Error (" + seed + "): !t.equals(m) after torture-test removal");

if (!m.equals(m.clone())) System.out.println("Error (" + seed + "): !m.equals(m.clone()) after torture-test removal");
if (!((OPEN_HASH_MAP)m.clone()).equals(m)) System.out.println("Error (" + seed + "): !m.clone().equals(m)
after torture-test removal");

m.trim();

if (!m.equals(t)) System.out.println("Error (" + seed + "): !m.equals(t) after trim()");
if (!t.equals(m)) System.out.println("Error (" + seed + "): !t.equals(m) after trim()");
#endif

System.out.println("Test OK");
return;

#endif
}

public static void main(String args[]) {
  float f = Hash.DEFAULT_LOAD_FACTOR;
  int n = Integer.parseInt(args[1]);
  if (args.length>2) f = Float.parseFloat(args[2]);
  if (args.length > 3) r = new java.util.Random(seed = Long.parseLong(args[3]));

  try {
    if ("speedTest".equals(args[0]) || "speedComp".equals(args[0])) speedTest(n, f, "speedComp".equals(args[0]));
    else if ("test".equals(args[0])) runTest(n, f);
  } catch(Throwable e) {
    e.printStackTrace(System.err);
    System.err.println("seed: " + seed);
  }

}

#endif

}

```

Found in path(s):

```
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-
a775574/drv/OpenHashMap.drv
```

No license file was found, but licenses were detected in source scan.

```
/*
```

```
* Copyright (C) 2002-2017 Sebastiano Vigna
```

```
*
```

```
* Licensed under the Apache License, Version 2.0 (the "License");
```

```
* you may not use this file except in compliance with the License.
```

```
* You may obtain a copy of the License at
```

```
*
```

```
* http://www.apache.org/licenses/LICENSE-2.0
```

```
*
```

```
* Unless required by applicable law or agreed to in writing, software
```

```
* distributed under the License is distributed on an "AS IS" BASIS,
```

```
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
```

```
* See the License for the specific language governing permissions and
```

```
* limitations under the License.
```

```
*/
```

```
package PACKAGE;
```

```
import it.unimi.dsi.fastutil.objects.AbstractObjectSortedSet;
```

```
import it.unimi.dsi.fastutil.objects.ObjectBidirectionalIterator;
```

```
import it.unimi.dsi.fastutil.objects.ObjectListIterator;
```

```
import it.unimi.dsi.fastutil.objects.ObjectSortedSet;
```

```
import VALUE_PACKAGE.VALUE_COLLECTION;
```

```
import VALUE_PACKAGE.VALUE_ABSTRACT_COLLECTION;
```

```
import VALUE_PACKAGE.VALUE_ITERATOR;
```

```
import java.util.Comparator;
```

```
import java.util.Iterator;
```

```
import java.util.Map;
```

```
import java.util.SortedMap;
```

```
import java.util.NoSuchElementException;
```

```
#if VALUES_PRIMITIVE
```

```
import VALUE_PACKAGE.VALUE_LIST_ITERATOR;
```

```
#endif
```

```
/** A type-specific AVL tree map with a fast, small-footprint implementation.
```

```
*
```

```
* <p>The iterators provided by the views of this class are type-specific { @linkplain
```

```
* it.unimi.dsi.fastutil.BidirectionalIterator bidirectional iterators }.
```

```
* Moreover, the iterator returned by { @code iterator() } can be safely cast
```

```
* to a type-specific { @linkplain java.util.ListIterator list iterator }.
```

```
*/
```

```

public class AVL_TREE_MAP KEY_VALUE_GENERIC extends ABSTRACT_SORTED_MAP
KEY_VALUE_GENERIC implements java.io.Serializable, Cloneable {

    /** A reference to the root entry. */
    protected transient Entry KEY_VALUE_GENERIC tree;

    /** Number of entries in this map. */
    protected int count;

    /** The first key in this map. */
    protected transient Entry KEY_VALUE_GENERIC firstEntry;

    /** The last key in this map. */
    protected transient Entry KEY_VALUE_GENERIC lastEntry;

    /** Cached set of entries. */
    protected transient ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> entries;

    /** Cached set of keys. */
    protected transient SORTED_SET KEY_GENERIC keys;

    /** Cached collection of values. */
    protected transient VALUE_COLLECTION VALUE_GENERIC values;

    /** The value of this variable remembers, after a { @code put() }
     * or a { @code remove() }, whether the <em>domain</em> of the map
     * has been modified. */
    protected transient boolean modified;

    /** This map's comparator, as provided in the constructor. */
    protected Comparator<? super KEY_GENERIC_CLASS> storedComparator;

    /** This map's actual comparator; it may differ from { @link #storedComparator } because it is
     always a type-specific comparator, so it could be derived from the former by wrapping. */
    protected transient KEY_COMPARATOR KEY_SUPER_GENERIC actualComparator;

    private static final long serialVersionUID = -7046029254386353129L;

    {
        allocatePaths();
    }

    /** Creates a new empty tree map.
     */

    public AVL_TREE_MAP() {
        tree = null;
    }

```

```

count = 0;
}

/** Generates the comparator that will be actually used.
 *
 * <p>When a given {@link Comparator} is specified and stored in {@link
 * #storedComparator}, we must check whether it is type-specific. If it is
 * so, we can use it directly, and we store it in {@link #actualComparator}. Otherwise,
 * we adapt it using a helper static method.
 */
private void setActualComparator() {
#if KEY_CLASS_Object
    actualComparator = storedComparator;
#else
    actualComparator = COMPARATORS.AS_KEY_COMPARATOR(storedComparator);
#endif
}

/** Creates a new empty tree map with the given comparator.
 *
 * @param c a (possibly type-specific) comparator.
 */
public AVL_TREE_MAP(final Comparator<? super KEY_GENERIC_CLASS> c) {
    this();
    storedComparator = c;
    setActualComparator();
}

/** Creates a new tree map copying a given map.
 *
 * @param m a {@link Map} to be copied into the new tree map.
 */
public AVL_TREE_MAP(final Map<? extends KEY_GENERIC_CLASS, ? extends VALUE_GENERIC_CLASS>
m) {
    this();
    putAll(m);
}

/** Creates a new tree map copying a given sorted map (and its {@link Comparator}).
 *
 * @param m a {@link SortedMap} to be copied into the new tree map.
 */
public AVL_TREE_MAP(final SortedMap<KEY_GENERIC_CLASS,VALUE_GENERIC_CLASS> m) {

```

```

    this(m.comparator());
    putAll(m);
}

/** Creates a new tree map copying a given map.
 *
 * @param m a type-specific map to be copied into the new tree map.
 */

public AVL_TREE_MAP(final MAP KEY_VALUE_EXTENDS_GENERIC m) {
    this();
    putAll(m);
}

/** Creates a new tree map copying a given sorted map (and its {@link Comparator}).
 *
 * @param m a type-specific sorted map to be copied into the new tree map.
 */

public AVL_TREE_MAP(final SORTED_MAP KEY_VALUE_GENERIC m) {
    this(m.comparator());
    putAll(m);
}

/** Creates a new tree map using the elements of two parallel arrays and the given comparator.
 *
 * @param k the array of keys of the new tree map.
 * @param v the array of corresponding values in the new tree map.
 * @param c a (possibly type-specific) comparator.
 * @throws IllegalArgumentException if {@code k} and {@code v} have different lengths.
 */

public AVL_TREE_MAP(final KEY_GENERIC_TYPE[] k, final VALUE_GENERIC_TYPE v[], final
Comparator<? super KEY_GENERIC_CLASS> c) {
    this(c);
    if (k.length != v.length) throw new IllegalArgumentException("The key array and the value array have different
lengths (" + k.length + " and " + v.length + ")");
    for(int i = 0; i < k.length; i++) this.put(k[i], v[i]);
}

/** Creates a new tree map using the elements of two parallel arrays.
 *
 * @param k the array of keys of the new tree map.
 * @param v the array of corresponding values in the new tree map.
 * @throws IllegalArgumentException if {@code k} and {@code v} have different lengths.
 */

public AVL_TREE_MAP(final KEY_GENERIC_TYPE[] k, final VALUE_GENERIC_TYPE v[]) {

```

```

    this(k, v, null);
}

/*
 * The following methods implements some basic building blocks used by
 * all accessors. They are (and should be maintained) identical to those used in AVLTreeSet.drv.
 *
 * The put()/remove() code is derived from Ben Pfaff's GNU libavl
 * (http://www.msu.edu/~pfaffben/avl/). If you want to understand what's
 * going on, you should have a look at the literate code contained therein
 * first.
 */

/** Compares two keys in the right way.
 *
 * <p>This method uses the {@link #actualComparator} if it is non-{@code null}.
 * Otherwise, it resorts to primitive type comparisons or to {@link Comparable#compareTo(Object) compareTo()}.
 *
 * @param k1 the first key.
 * @param k2 the second key.
 * @return a number smaller than, equal to or greater than 0, as usual
 * (i.e., when  $k1 < k2$ ,  $k1 = k2$  or  $k1 > k2$ , respectively).
 */

SUPPRESS_WARNINGS_KEY_UNCHECKED
final int compare(final KEY_GENERIC_TYPE k1, final KEY_GENERIC_TYPE k2) {
    return actualComparator == null ? KEY_CMP(k1, k2) : actualComparator.compare(k1, k2);
}

/** Returns the entry corresponding to the given key, if it is in the tree; {@code null}, otherwise.
 *
 * @param k the key to search for.
 * @return the corresponding entry, or {@code null} if no entry with the given key exists.
 */

final Entry KEY_VALUE_GENERIC findKey(final KEY_GENERIC_TYPE k) {
    Entry KEY_VALUE_GENERIC e = tree;
    int cmp;

    while (e != null && (cmp = compare(k, e.key)) != 0) e = cmp < 0 ? e.left() : e.right();

    return e;
}

/** Locates a key.

```

```

*
* @param k a key.
* @return the last entry on a search for the given key; this will be
* the given key, if it present; otherwise, it will be either the smallest greater key or the greatest smaller key.
*/

final Entry KEY_VALUE_GENERIC locateKey(final KEY_GENERIC_TYPE k) {
    Entry KEY_VALUE_GENERIC e = tree, last = tree;
    int cmp = 0;

    while (e != null && (cmp = compare(k, e.key)) != 0) {
        last = e;
        e = cmp < 0 ? e.left() : e.right();
    }

    return cmp == 0 ? e : last;
}

/** This vector remembers the directions followed during
 * the current insertion. It suffices for about 232 entries. */
private transient boolean dirPath[];

private void allocatePaths() {
    dirPath = new boolean[48];
}

#if VALUES_PRIMITIVE && !VALUE_CLASS_Boolean
/** Adds an increment to value currently associated with a key.
 *
 * <p>Note that this method respects the { @linkplain #defaultReturnValue() default return value } semantics: when
 * called with a key that does not currently appears in the map, the key
 * will be associated with the default return value plus
 * the given increment.
 *
 * @param k the key.
 * @param incr the increment.
 * @return the old value, or the { @linkplain #defaultReturnValue() default return value } if no value was present for
 * the given key.
 */
public VALUE_GENERIC_TYPE addTo(final KEY_GENERIC_TYPE k, final VALUE_GENERIC_TYPE incr) {
    Entry KEY_VALUE_GENERIC e = add(k);
    final VALUE_GENERIC_TYPE oldValue = e.value;
    e.value += incr;
    return oldValue;
}
#endif

@Override

```

```

public VALUE_GENERIC_TYPE put(final KEY_GENERIC_TYPE k, final VALUE_GENERIC_TYPE v) {
    Entry KEY_VALUE_GENERIC e = add(k);
    final VALUE_GENERIC_TYPE oldValue = e.value;
    e.value = v;
    return oldValue;
}

/** Returns a node with key k in the balanced tree, creating one with defRetValue if necessary.
 *
 * @param k the key
 * @return a node with key k. If a node with key k already exists, then that node is returned,
 * otherwise a new node with defRetValue is created ensuring that the tree is balanced
 * after creation of the node.
 */
private Entry KEY_VALUE_GENERIC add(final KEY_GENERIC_TYPE k) {
    /* After execution of this method, modified is true iff a new entry has
    been inserted. */
    modified = false;

    Entry KEY_VALUE_GENERIC e = null;
    if (tree == null) { // The case of the empty tree is treated separately.
        count++;
        e = tree = lastEntry = firstEntry = new Entry KEY_VALUE_GENERIC_DIAMOND(k, defRetValue);
        modified = true;
    }
    else {
        Entry KEY_VALUE_GENERIC p = tree, q = null, y = tree, z = null, w = null;
        int cmp, i = 0;

        while(true) {
            if ((cmp = compare(k, p.key)) == 0) {
                return p;
            }

            if (p.balance() != 0) {
                i = 0;
                z = q;
                y = p;
            }

            if (dirPath[i++] = cmp > 0) {
                if (p.succ()) {
                    count++;
                    e = new Entry KEY_VALUE_GENERIC_DIAMOND(k, defRetValue);

                    modified = true;
                    if (p.right == null) lastEntry = e;
                }
            }
        }
    }
}

```

```

e.left = p;
e.right = p.right;

p.right(e);

break;
}

q = p;
p = p.right;
}
else {
if (p.pred()) {
count++;
e = new Entry KEY_VALUE_GENERIC_DIAMOND(k, defRetVal);

modified = true;
if (p.left == null) firstEntry = e;

e.right = p;
e.left = p.left;

p.left(e);

break;
}

q = p;
p = p.left;
}
}

p = y;
i = 0;

while(p != e) {
if (dirPath[i] p.incBalance();
else p.decBalance();

p = dirPath[i++] ? p.right : p.left;
}

if (y.balance() == -2) {
Entry KEY_VALUE_GENERIC x = y.left;

if (x.balance() == -1) {
w = x;
if (x.succ()) {

```

```

x.succ(false);
y.pred(x);
}
else y.left = x.right;

x.right = y;
x.balance(0);
y.balance(0);
}
else {
assert x.balance() == 1;

w = x.right;
x.right = w.left;
w.left = x;
y.left = w.right;
w.right = y;
if (w.balance() == -1) {
x.balance(0);
y.balance(1);
}
else if (w.balance() == 0) {
x.balance(0);
y.balance(0);
}
else {
x.balance(-1);
y.balance(0);
}
w.balance(0);

if (w.pred()) {
x.succ(w);
w.pred(false);
}
if (w.succ()) {
y.pred(w);
w.succ(false);
}

}
}
else if (y.balance() == +2) {
Entry KEY_VALUE_GENERIC x = y.right;

if (x.balance() == 1) {
w = x;

```

```

if (x.pred()) {
    x.pred(false);
    y.succ(x);
}
else y.right = x.left;

x.left = y;
x.balance(0);
y.balance(0);
}
else {
    assert x.balance() == -1;

    w = x.left;
    x.left = w.right;
    w.right = x;
    y.right = w.left;
    w.left = y;
    if (w.balance() == 1) {
        x.balance(0);
        y.balance(-1);
    }
    else if (w.balance() == 0) {
        x.balance(0);
        y.balance(0);
    }
    else {
        x.balance(1);
        y.balance(0);
    }
    w.balance(0);

    if (w.pred()) {
        y.succ(w);
        w.pred(false);
    }
    if (w.succ()) {
        x.pred(w);
        w.succ(false);
    }

}
}
else return e;

if (z == null) tree = w;
else {

```

```

    if (z.left == y) z.left = w;
    else z.right = w;
  }
}

return e;
}

/** Finds the parent of an entry.
 *
 * @param e a node of the tree.
 * @return the parent of the given node, or { @code null } for the root.
 */

private Entry KEY_VALUE_GENERIC parent(final Entry KEY_VALUE_GENERIC e) {
  if (e == tree) return null;

  Entry KEY_VALUE_GENERIC x, y, p;
  x = y = e;

  while(true) {
    if (y.succ()) {
      p = y.right;
      if (p == null || p.left != e) {
        while(! x.pred()) x = x.left;
        p = x.left;
      }
      return p;
    }
    else if (x.pred()) {
      p = x.left;
      if (p == null || p.right != e) {
        while(! y.succ()) y = y.right;
        p = y.right;
      }
      return p;
    }
  }

  x = x.left;
  y = y.right;
}

/** After execution of this method, { @link #modified } is true iff an entry
has been deleted. */

```

SUPPRESS_WARNINGS_KEY_UNCHECKED

```

@Override
public VALUE_GENERIC_TYPE REMOVE_VALUE(final KEY_TYPE k) {
    modified = false;

    if (tree == null) return defRetVal;

    int cmp;
    Entry KEY_VALUE_GENERIC p = tree, q = null;
    boolean dir = false;
    final KEY_GENERIC_TYPE kk = KEY_GENERIC_CAST k;

    while(true) {
        if ((cmp = compare(kk, p.key)) == 0) break;
        else if (dir = cmp > 0) {
            q = p;
            if ((p = p.right()) == null) return defRetVal;
        }
        else {
            q = p;
            if ((p = p.left()) == null) return defRetVal;
        }
    }

    if (p.left == null) firstEntry = p.next();
    if (p.right == null) lastEntry = p.prev();

    if (p.succ()) {
        if (p.pred()) {
            if (q != null) {
                if (dir) q.succ(p.right);
                else q.pred(p.left);
            }
            else tree = dir ? p.right : p.left;
        }
        else {
            p.prev().right = p.right;

            if (q != null) {
                if (dir) q.right = p.left;
                else q.left = p.left;
            }
            else tree = p.left;
        }
    }
    else {
        Entry KEY_VALUE_GENERIC r = p.right;

        if (r.pred()) {

```

```

r.left = p.left;
r.pred(p.pred());
if (! r.pred()) r.prev().right = r;
if (q != null) {
    if (dir) q.right = r;
    else q.left = r;
}
else tree = r;

r.balance(p.balance());
q = r;
dir = true;

}
else {
    Entry KEY_VALUE_GENERIC s;

    while(true) {
        s = r.left;
        if (s.pred()) break;
        r = s;
    }

    if (s.succ()) r.pred(s);
    else r.left = s.right;

    s.left = p.left;

    if (! p.pred()) {
        p.prev().right = s;
        s.pred(false);
    }

    s.right = p.right;
    s.succ(false);

    if (q != null) {
        if (dir) q.right = s;
        else q.left = s;
    }
    else tree = s;

    s.balance(p.balance());
    q = r;
    dir = false;
}
}

```

```

Entry KEY_VALUE_GENERIC y;

while(q != null) {
    y = q;
    q = parent(y);

    if (! dir) {
        dir = q != null && q.left != y;
        y.incBalance();

        if (y.balance() == 1) break;
        else if (y.balance() == 2) {

            Entry KEY_VALUE_GENERIC x = y.right;
            assert x != null;

            if (x.balance() == -1) {
                Entry KEY_VALUE_GENERIC w;

                assert x.balance() == -1;

                w = x.left;
                x.left = w.right;
                w.right = x;
                y.right = w.left;
                w.left = y;

                if (w.balance() == 1) {
                    x.balance(0);
                    y.balance(-1);
                }
                else if (w.balance() == 0) {
                    x.balance(0);
                    y.balance(0);
                }
                else {
                    assert w.balance() == -1;

                    x.balance(1);
                    y.balance(0);
                }

                w.balance(0);

                if (w.pred()) {
                    y.succ(w);
                    w.pred(false);
                }
            }
        }
    }
}

```

```

if (w.succ()) {
    x.pred(w);
    w.succ(false);
}

if (q != null) {
    if (dir) q.right = w;
    else q.left = w;
}
else tree = w;
}
else {
    if (q != null) {
        if (dir) q.right = x;
        else q.left = x;
    }
    else tree = x;

    if (x.balance() == 0) {
        y.right = x.left;
        x.left = y;
        x.balance(-1);
        y.balance(+1);
        break;
    }
    assert x.balance() == 1;

    if (x.pred()) {
        y.succ(true);
        x.pred(false);
    }
    else y.right = x.left;

    x.left = y;
    y.balance(0);
    x.balance(0);
}
}
}
else {
    dir = q != null && q.left != y;
    y.decBalance();

    if (y.balance() == -1) break;
    else if (y.balance() == -2) {

        Entry KEY_VALUE_GENERIC x = y.left;
        assert x != null;

```

```

if (x.balance() == 1) {
    Entry KEY_VALUE_GENERIC w;

    assert x.balance() == 1;

    w = x.right;
    x.right = w.left;
    w.left = x;
    y.left = w.right;
    w.right = y;

    if (w.balance() == -1) {
        x.balance(0);
        y.balance(1);
    }
    else if (w.balance() == 0) {
        x.balance(0);
        y.balance(0);
    }
    else {
        assert w.balance() == 1;

        x.balance(-1);
        y.balance(0);
    }

    w.balance(0);

    if (w.pred()) {
        x.succ(w);
        w.pred(false);
    }
    if (w.succ()) {
        y.pred(w);
        w.succ(false);
    }

    if (q != null) {
        if (dir) q.right = w;
        else q.left = w;
    }
    else tree = w;
}
else {
    if (q != null) {
        if (dir) q.right = x;
        else q.left = x;
    }
}

```

```

    }
    else tree = x;

    if (x.balance() == 0) {
        y.left = x.right;
        x.right = y;
        x.balance(+1);
        y.balance(-1);
        break;
    }
    assert x.balance() == -1;

    if (x.succ()) {
        y.pred(true);
        x.succ(false);
    }
    else y.left = x.right;

    x.right = y;
    y.balance(0);
    x.balance(0);
    }
    }
    }
    }

    modified = true;
    count--;
    return p.value;
}

@Override
public boolean containsValue(final VALUE_TYPE v) {
    final ValueIterator i = new ValueIterator();
    VALUE_GENERIC_TYPE ev;

    int j = count;
    while(j-- != 0) {
        ev = i.NEXT_VALUE();
        if (VALUE_EQUALS(ev, v)) return true;
    }

    return false;
}

@Override
public void clear() {

```

```

count = 0;
tree = null;
entries = null;
values = null;
keys = null;
firstEntry = lastEntry = null;
}

```

```

/** This class represent an entry in a tree map.

```

```

*

```

```

* <p>We use the only "metadata", i.e., { @link Entry#info}, to store
* information about balance, predecessor status and successor status.

```

```

*

```

```

* <p>Note that since the class is recursive, it can be

```

```

* considered equivalently a tree.

```

```

*/

```

```

private static final class Entry KEY_VALUE_GENERIC extends ABSTRACT_MAP.BasicEntry

```

```

KEY_VALUE_GENERIC implements Cloneable {

```

```

/** If the bit in this mask is true, { @link #right} points to a successor. */

```

```

private static final int SUCC_MASK = 1 << 31;

```

```

/** If the bit in this mask is true, { @link #left} points to a predecessor. */

```

```

private static final int PRED_MASK = 1 << 30;

```

```

/** The bits in this mask hold the node balance info. You can get it just by casting to byte. */

```

```

private static final int BALANCE_MASK = 0xFF;

```

```

/** The pointers to the left and right subtrees. */

```

```

Entry KEY_VALUE_GENERIC left, right;

```

```

/** This integers holds different information in different bits (see { @link #SUCC_MASK}, { @link
#PRED_MASK} and { @link #BALANCE_MASK}). */

```

```

int info;

```

```

Entry() {

```

```

super(KEY_NULL, VALUE_NULL);

```

```

}

```

```

/** Creates a new entry with the given key and value.

```

```

*

```

```

* @param k a key.

```

```

* @param v a value.

```

```

*/

```

```

Entry(final KEY_GENERIC_TYPE k, final VALUE_GENERIC_TYPE v) {

```

```

super(k, v);

```

```

info = SUCC_MASK | PRED_MASK;

```

```

}

```

```

/** Returns the left subtree.

```

```

*

```

```

* @return the left subtree ({ @code null} if the left
* subtree is empty).
*/
Entry KEY_VALUE_GENERIC left() {
return (info & PRED_MASK) != 0 ? null : left;
}

/** Returns the right subtree.
*
* @return the right subtree ({ @code null} if the right
* subtree is empty).
*/
Entry KEY_VALUE_GENERIC right() {
return (info & SUCC_MASK) != 0 ? null : right;
}

/** Checks whether the left pointer is really a predecessor.
* @return true if the left pointer is a predecessor.
*/
boolean pred() {
return (info & PRED_MASK) != 0;
}

/** Checks whether the right pointer is really a successor.
* @return true if the right pointer is a successor.
*/
boolean succ() {
return (info & SUCC_MASK) != 0;
}

/** Sets whether the left pointer is really a predecessor.
* @param pred if true then the left pointer will be considered a predecessor.
*/
void pred(final boolean pred) {
if (pred) info |= PRED_MASK;
else info &= ~PRED_MASK;
}

/** Sets whether the right pointer is really a successor.
* @param succ if true then the right pointer will be considered a successor.
*/
void succ(final boolean succ) {
if (succ) info |= SUCC_MASK;
else info &= ~SUCC_MASK;
}

/** Sets the left pointer to a predecessor.
* @param pred the predecessr.

```

```

*/
void pred(final Entry KEY_VALUE_GENERIC pred) {
    info |= PRED_MASK;
    left = pred;
}

/** Sets the right pointer to a successor.
 * @param succ the successor.
 */
void succ(final Entry KEY_VALUE_GENERIC succ) {
    info |= SUCC_MASK;
    right = succ;
}

/** Sets the left pointer to the given subtree.
 * @param left the new left subtree.
 */
void left(final Entry KEY_VALUE_GENERIC left) {
    info &= ~PRED_MASK;
    this.left = left;
}

/** Sets the right pointer to the given subtree.
 * @param right the new right subtree.
 */
void right(final Entry KEY_VALUE_GENERIC right) {
    info &= ~SUCC_MASK;
    this.right = right;
}

/** Returns the current level of the node.
 * @return the current level of this node.
 */
int balance() {
    return (byte)info;
}

/** Sets the level of this node.
 * @param level the new level of this node.
 */
void balance(int level) {
    info &= ~BALANCE_MASK;
    info |= (level & BALANCE_MASK);
}

/** Increments the level of this node. */
void incBalance() {
    info = info & ~BALANCE_MASK | ((byte)info + 1) & 0xFF;
}

```

```

}

/** Decrements the level of this node. */
protected void decBalance() {
    info = info & ~BALANCE_MASK | ((byte)info - 1) & 0xFF;
}

/** Computes the next entry in the set order.
 *
 * @return the next entry ({@code null}) if this is the last entry).
 */

Entry KEY_VALUE_GENERIC next() {
    Entry KEY_VALUE_GENERIC next = this.right;
    if ((info & SUCC_MASK) == 0) while ((next.info & PRED_MASK) == 0) next = next.left;
    return next;
}

/** Computes the previous entry in the set order.
 *
 * @return the previous entry ({@code null}) if this is the first entry).
 */

Entry KEY_VALUE_GENERIC prev() {
    Entry KEY_VALUE_GENERIC prev = this.left;
    if ((info & PRED_MASK) == 0) while ((prev.info & SUCC_MASK) == 0) prev = prev.right;
    return prev;
}

@Override
public VALUE_GENERIC_TYPE setValue(final VALUE_GENERIC_TYPE value) {
    final VALUE_GENERIC_TYPE oldValue = this.value;
    this.value = value;
    return oldValue;
}

@Override
SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
public Entry KEY_VALUE_GENERIC clone() {
    Entry KEY_VALUE_GENERIC c;
    try {
        c = (Entry KEY_VALUE_GENERIC)super.clone();
    }
    catch(CloneNotSupportedException cantHappen) {
        throw new InternalError();
    }

    c.key = key;

```

```

c.value = value;
c.info = info;

return c;
}

@Override
@SuppressWarnings("unchecked")
public boolean equals(final Object o) {
    if (!(o instanceof Map.Entry)) return false;
    Map.Entry<KEY_GENERIC_CLASS, VALUE_GENERIC_CLASS> e = (Map.Entry<KEY_GENERIC_CLASS,
VALUE_GENERIC_CLASS>)o;

    return KEY_EQUALS(key, KEY_CLASS2TYPE(e.getKey())) && VALUE_EQUALS(value,
VALUE_CLASS2TYPE(e.getValue()));
}

@Override
public int hashCode() {
    return KEY2JAVAHASH_NOT_NULL(key) ^ VALUE2JAVAHASH(value);
}

@Override
public String toString() {
    return key + "=>" + value;
}

/*
public void prettyPrint() {
    prettyPrint(0);
}

public void prettyPrint(int level) {
    if (pred()) {
        for (int i = 0; i < level; i++)
            System.err.print(" ");
        System.err.println("pred: " + left);
    }
    else if (left != null)
        left.prettyPrint(level + 1);
    for (int i = 0; i < level; i++)
        System.err.print(" ");
    System.err.println(key + "=" + value + " (" + balance() + ")");
    if (succ()) {
        for (int i = 0; i < level; i++)
            System.err.print(" ");

```

```

        System.err.println("succ: " + right);
    }
    else if (right != null)
        right.prettyPrint(level + 1);
    }
    */
}

/*
public void prettyPrint() {
    System.err.println("size: " + count);
    if (tree != null) tree.prettyPrint();
}
*/

SUPPRESS_WARNINGS_KEY_UNCHECKED
@Override
public boolean containsKey(final KEY_TYPE k) {
    return findKey(KEY_GENERIC_CAST k) != null;
}

@Override
public int size() {
    return count;
}

@Override
public boolean isEmpty() {
    return count == 0;
}

SUPPRESS_WARNINGS_KEY_UNCHECKED
@Override
public VALUE_GENERIC_TYPE GET_VALUE(final KEY_TYPE k) {
    final Entry KEY_VALUE_GENERIC e = findKey(KEY_GENERIC_CAST k);
    return e == null ? defRetValue : e.value;
}

@Override
public KEY_GENERIC_TYPE FIRST_KEY() {
    if (tree == null) throw new NoSuchElementException();
    return firstEntry.key;
}

@Override
public KEY_GENERIC_TYPE LAST_KEY() {
    if (tree == null) throw new NoSuchElementException();
    return lastEntry.key;
}

```

```

}

/** An abstract iterator on the whole range.
 *
 * <p>This class can iterate in both directions on a threaded tree.
 */

private class TreeIterator {
    /** The entry that will be returned by the next call to {@link java.util.ListIterator#previous()} (or {@code null} if
    no previous entry exists). */
    Entry KEY_VALUE_GENERIC prev;
    /** The entry that will be returned by the next call to {@link java.util.ListIterator#next()} (or {@code null} if no
    next entry exists). */
    Entry KEY_VALUE_GENERIC next;
    /** The last entry that was returned (or {@code null} if we did not iterate or used {@link #remove()}). */
    Entry KEY_VALUE_GENERIC curr;
    /** The current index (in the sense of a {@link java.util.ListIterator}). Note that this value is not meaningful when
    this {@link TreeIterator} has been created using the nonempty constructor.*/
    int index = 0;

    TreeIterator() {
        next = firstEntry;
    }

    TreeIterator(final KEY_GENERIC_TYPE k) {
        if ((next = locateKey(k)) != null) {
            if (compare(next.key, k) <= 0) {
                prev = next;
                next = next.next();
            }
            else prev = next.prev();
        }
    }

    public boolean hasNext() { return next != null; }

    public boolean hasPrevious() { return prev != null; }

    void updateNext() {
        next = next.next();
    }

    Entry KEY_VALUE_GENERIC nextEntry() {
        if (! hasNext()) throw new NoSuchElementException();
        curr = prev = next;
        index++;
        updateNext();
    }
}

```

```

return curr;
}

void updatePrevious() {
    prev = prev.prev();
}

Entry KEY_VALUE_GENERIC previousEntry() {
    if (!hasPrevious()) throw new NoSuchElementException();
    curr = next = prev;
    index--;
    updatePrevious();
    return curr;
}

public int nextIndex() {
    return index;
}

public int previousIndex() {
    return index - 1;
}

public void remove() {
    if (curr == null) throw new IllegalStateException();
    /* If the last operation was a next(), we are removing an entry that precedes
       the current index, and thus we must decrement it. */
    if (curr == prev) index--;
    next = prev = curr;
    updatePrevious();
    updateNext();
    AVL_TREE_MAP.this.REMOVE_VALUE(curr.key);
    curr = null;
}

public int skip(final int n) {
    int i = n;
    while(i-- != 0 && hasNext()) nextEntry();
    return n - i - 1;
}

public int back(final int n) {
    int i = n;
    while(i-- != 0 && hasPrevious()) previousEntry();
    return n - i - 1;
}
}

```

```

/** An iterator on the whole range.
 *
 * <p>This class can iterate in both directions on a threaded tree.
 */

private class EntryIterator extends TreeIterator implements ObjectListIterator<MAP.Entry
KEY_VALUE_GENERIC> {
    EntryIterator() {}

    EntryIterator(final KEY_GENERIC_TYPE k) {
        super(k);
    }

    @Override
    public MAP.Entry KEY_VALUE_GENERIC next() { return nextEntry(); }

    @Override
    public MAP.Entry KEY_VALUE_GENERIC previous() { return previousEntry(); }

    @Override
    public void set(MAP.Entry KEY_VALUE_GENERIC ok) { throw new UnsupportedOperationException(); }

    @Override
    public void add(MAP.Entry KEY_VALUE_GENERIC ok) { throw new UnsupportedOperationException(); }
}

@Override
public ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> ENTRYSET() {
    if (entries == null) entries = new AbstractObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC>() {
        final Comparator<? super MAP.Entry KEY_VALUE_GENERIC> comparator = (Comparator<MAP.Entry
KEY_VALUE_GENERIC>) (x, y) -> AVL_TREE_MAP.this.actualComparator.compare(x.ENTRY_GET_KEY(),
y.ENTRY_GET_KEY());
    };
}

@Override
public Comparator<? super MAP.Entry KEY_VALUE_GENERIC> comparator() { return comparator; }

@Override
public ObjectBidirectionalIterator<MAP.Entry KEY_VALUE_GENERIC> iterator() { return new EntryIterator(); }

@Override
public ObjectBidirectionalIterator<MAP.Entry KEY_VALUE_GENERIC> iterator(final MAP.Entry
KEY_VALUE_GENERIC from) { return new EntryIterator(from.ENTRY_GET_KEY()); }

@Override
    SUPPRESS_WARNINGS_KEY_UNCHECKED
    public boolean contains(final Object o) {

```

```

    if (!(o instanceof Map.Entry)) return false;
    final Map.Entry<?,?> e = (Map.Entry<?,?>)o;
#if KEYS_PRIMITIVE
    if (e.getKey() == null || !(e.getKey() instanceof KEY_CLASS)) return false;
#endif
#if VALUES_PRIMITIVE
    if (e.getValue() == null || !(e.getValue() instanceof VALUE_CLASS)) return false;
#endif
    final Entry KEY_VALUE_GENERIC f = findKey(KEY_OBJ2TYPE(KEY_GENERIC_CAST e.getKey()));
    return e.equals(f);
}

@Override
SUPPRESS_WARNINGS_KEY_UNCHECKED
public boolean remove(final Object o) {
    if (!(o instanceof Map.Entry)) return false;
    final Map.Entry<?,?> e = (Map.Entry<?,?>)o;
#if KEYS_PRIMITIVE
    if (e.getKey() == null || !(e.getKey() instanceof KEY_CLASS)) return false;
#endif
#if VALUES_PRIMITIVE
    if (e.getValue() == null || !(e.getValue() instanceof VALUE_CLASS)) return false;
#endif
    final Entry KEY_VALUE_GENERIC f = findKey(KEY_OBJ2TYPE(KEY_GENERIC_CAST e.getKey()));
    if (f == null || !VALUE_EQUALS(f.ENTRY_GET_VALUE(), VALUE_OBJ2TYPE(e.getValue()))) return false;
    AVL_TREE_MAP.this.REMOVE_VALUE(f.key);
    return true;
}

@Override
public int size() { return count; }

@Override
public void clear() { AVL_TREE_MAP.this.clear(); }

@Override
public MAP.Entry KEY_VALUE_GENERIC first() { return firstEntry; }

@Override
public MAP.Entry KEY_VALUE_GENERIC last() { return lastEntry; }

@Override
public ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> subSet(MAP.Entry KEY_VALUE_GENERIC
from, MAP.Entry KEY_VALUE_GENERIC to) { return subMap(from.ENTRY_GET_KEY(),
to.ENTRY_GET_KEY()).ENTRYSET(); }

@Override
public ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> headSet(MAP.Entry KEY_VALUE_GENERIC

```

```

to) { return headMap(to.ENTRY_GET_KEY()).ENTRYSET(); }

    @Override
    public ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> tailSet(MAP.Entry KEY_VALUE_GENERIC
from) { return tailMap(from.ENTRY_GET_KEY()).ENTRYSET(); }
    };

return entries;
}

/** An iterator on the whole range of keys.
 *
 * <p>This class can iterate in both directions on the keys of a threaded tree. We
 * simply override the { @link java.util.ListIterator#next()}/{ @link java.util.ListIterator#previous()} methods (and
possibly
 * their type-specific counterparts) so that they return keys instead of entries.
 */
private final class KeyIterator extends TreeIterator implements KEY_LIST_ITERATOR KEY_GENERIC {
    public KeyIterator() {}
    public KeyIterator(final KEY_GENERIC_TYPE k) { super(k); }
    @Override
    public KEY_GENERIC_TYPE NEXT_KEY() { return nextEntry().key; }
    @Override
    public KEY_GENERIC_TYPE PREV_KEY() { return previousEntry().key; }
}

/** A keyset implementation using a more direct implementation for iterators. */
private class KeySet extends ABSTRACT_SORTED_MAP KEY_VALUE_GENERIC.KeySet {
    @Override
    public KEY_BIDI_ITERATOR KEY_GENERIC iterator() { return new KeyIterator(); }
    @Override
    public KEY_BIDI_ITERATOR KEY_GENERIC iterator(final KEY_GENERIC_TYPE from) { return new
KeyIterator(from); }
}

/** Returns a type-specific sorted set view of the keys contained in this map.
 *
 * <p>In addition to the semantics of { @link java.util.Map#keySet()}, you can
 * safely cast the set returned by this call to a type-specific sorted
 * set interface.
 *
 * @return a type-specific sorted set view of the keys contained in this map.
 */
@Override
public SORTED_SET KEY_GENERIC keySet() {
    if (keys == null) keys = new KeySet();
    return keys;
}

```

```

/** An iterator on the whole range of values.
 *
 * <p>This class can iterate in both directions on the values of a threaded tree. We
 * simply override the { @link java.util.ListIterator#next()}/{ @link java.util.ListIterator#previous() } methods (and
 possibly
 * their type-specific counterparts) so that they return values instead of entries.
 */
private final class ValueIterator extends TreeIterator implements VALUE_LIST_ITERATOR VALUE_GENERIC {
    @Override
    public VALUE_GENERIC_TYPE NEXT_VALUE() { return nextEntry().value; }
    @Override
    public VALUE_GENERIC_TYPE PREV_VALUE() { return previousEntry().value; }
}

/** Returns a type-specific collection view of the values contained in this map.
 *
 * <p>In addition to the semantics of { @link java.util.Map#values() }, you can
 * safely cast the collection returned by this call to a type-specific collection
 * interface.
 *
 * @return a type-specific collection view of the values contained in this map.
 */
@Override
public VALUE_COLLECTION VALUE_GENERIC values() {
    if (values == null) values = new VALUE_ABSTRACT_COLLECTION VALUE_GENERIC() {
        @Override
        public VALUE_ITERATOR VALUE_GENERIC iterator() { return new ValueIterator(); }
        @Override
        public boolean contains(final VALUE_TYPE k) { return containsValue(k); }
        @Override
        public int size() { return count; }
        @Override
        public void clear() { AVL_TREE_MAP.this.clear(); }
    };

    return values;
}

@Override
public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() { return actualComparator; }

@Override
public SORTED_MAP KEY_VALUE_GENERIC headMap(KEY_GENERIC_TYPE to) { return new
Submap(KEY_NULL, true, to, false); }

@Override
public SORTED_MAP KEY_VALUE_GENERIC tailMap(KEY_GENERIC_TYPE from) { return new

```

```
Submap(from, false, KEY_NULL, true); }
```

```
@Override
```

```
public SORTED_MAP KEY_VALUE_GENERIC subMap(KEY_GENERIC_TYPE from, KEY_GENERIC_TYPE  
to) { return new Submap(from, false, to, false); }
```

```
/** A submap with given range.
```

```
*
```

```
* <p>This class represents a submap. One has to specify the left/right  
* limits (which can be set to  $-\infty$ ; or  $\infty$ ). Since the submap is a  
* view on the map, at a given moment it could happen that the limits of  
* the range are not any longer in the main map. Thus, things such as  
* { @link java.util.SortedMap#firstKey\(\) } or { @link java.util.Collection#size\(\) } must be always computed  
* on-the-fly.
```

```
*/
```

```
private final class Submap extends ABSTRACT_SORTED_MAP KEY_VALUE_GENERIC implements  
java.io.Serializable {
```

```
private static final long serialVersionUID = -7046029254386353129L;
```

```
/** The start of the submap range, unless { @link #bottom } is true. */
```

```
KEY_GENERIC_TYPE from;
```

```
/** The end of the submap range, unless { @link #top } is true. */
```

```
KEY_GENERIC_TYPE to;
```

```
/** If true, the submap range starts from  $-\infty$ . */
```

```
boolean bottom;
```

```
/** If true, the submap range goes to  $\infty$ . */
```

```
boolean top;
```

```
/** Cached set of entries. */
```

```
protected transient ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> entries;
```

```
/** Cached set of keys. */
```

```
protected transient SORTED_SET KEY_GENERIC keys;
```

```
/** Cached collection of values. */
```

```
protected transient VALUE_COLLECTION VALUE_GENERIC values;
```

```
/** Creates a new submap with given key range.
```

```
*
```

```
* @param from the start of the submap range.
```

```
* @param bottom if true, the first parameter is ignored and the range starts from  $-\infty$ ;
```

```
* @param to the end of the submap range.
```

```
* @param top if true, the third parameter is ignored and the range goes to  $\infty$ ;
```

```
*/
```

```
public Submap(final KEY_GENERIC_TYPE from, final boolean bottom, final KEY_GENERIC_TYPE to, final  
boolean top) {
```

```
if (! bottom && ! top && AVL_TREE_MAP.this.compare(from, to) > 0) throw new  
IllegalArgumentException("Start key (" + from + ") is larger than end key (" + to + ")");
```

```
this.from = from;
```

```
this.bottom = bottom;
```

```

this.to = to;
this.top = top;
this.defRetVal = AVL_TREE_MAP.this.defRetVal;
}

@Override
public void clear() {
    final SubmapIterator i = new SubmapIterator();
    while(i.hasNext()) {
        i.nextEntry();
        i.remove();
    }
}

/** Checks whether a key is in the submap range.
 * @param k a key.
 * @return true if is the key is in the submap range.
 */
final boolean in(final KEY_GENERIC_TYPE k) {
    return (bottom || AVL_TREE_MAP.this.compare(k, from) >= 0) &&
        (top || AVL_TREE_MAP.this.compare(k, to) < 0);
}

@Override
public ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> ENTRYSET() {
    if (entries == null) entries = new AbstractObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC>() {
        @Override
        public ObjectBidirectionalIterator<MAP.Entry KEY_VALUE_GENERIC> iterator() { return new
SubmapEntryIterator(); }
        @Override
        public ObjectBidirectionalIterator<MAP.Entry KEY_VALUE_GENERIC> iterator(final MAP.Entry
KEY_VALUE_GENERIC from) { return new SubmapEntryIterator(from.ENTRY_GET_KEY()); }
        @Override
        public Comparator<? super MAP.Entry KEY_VALUE_GENERIC> comparator() { return
AVL_TREE_MAP.this.ENTRYSET().comparator(); }
        @Override
        SUPPRESS_WARNINGS_KEY_UNCHECKED
        public boolean contains(final Object o) {
            if (!(o instanceof Map.Entry)) return false;
            final Map.Entry<?,?> e = (Map.Entry<?,?>)o;
            #if KEYS_PRIMITIVE
                if (e.getKey() == null || !(e.getKey() instanceof KEY_CLASS)) return false;
            #endif
            #if VALUES_PRIMITIVE
                if (e.getValue() == null || !(e.getValue() instanceof VALUE_CLASS)) return false;
            #endif
            final AVL_TREE_MAP.Entry KEY_VALUE_GENERIC f = findKey(KEY_OBJ2TYPE(KEY_GENERIC_CAST
e.getKey()));

```

```

        return f != null && in(f.key) && e.equals(f);
    }
    @Override
    SUPPRESS_WARNINGS_KEY_UNCHECKED
    public boolean remove(final Object o) {
        if (!(o instanceof Map.Entry)) return false;
        final Map.Entry<?,?> e = (Map.Entry<?,?>)o;
    #if KEYS_PRIMITIVE
        if (e.getKey() == null || !(e.getKey() instanceof KEY_CLASS)) return false;
    #endif
    #if VALUES_PRIMITIVE
        if (e.getValue() == null || !(e.getValue() instanceof VALUE_CLASS)) return false;
    #endif
        final AVL_TREE_MAP.Entry KEY_VALUE_GENERIC f = findKey(KEY_OBJ2TYPE(KEY_GENERIC_CAST
e.getKey()));
        if (f != null && in(f.key)) Submap.this.REMOVE_VALUE(f.key);
        return f != null;
    }
    @Override
    public int size() {
        int c = 0;
        for(Iterator<?> i = iterator(); i.hasNext(); i.next()) c++;
        return c;
    }
    @Override
    public boolean isEmpty() { return ! new SubmapIterator().hasNext(); }
    @Override
    public void clear() { Submap.this.clear(); }
    @Override
    public MAP.Entry KEY_VALUE_GENERIC first() { return firstEntry(); }
    @Override
    public MAP.Entry KEY_VALUE_GENERIC last() { return lastEntry(); }
    @Override
    public ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> subSet(MAP.Entry KEY_VALUE_GENERIC
from, MAP.Entry KEY_VALUE_GENERIC to) { return subMap(from.ENTRY_GET_KEY(),
to.ENTRY_GET_KEY()).ENTRYSET(); }
    @Override
    public ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> headSet(MAP.Entry KEY_VALUE_GENERIC
to) { return headMap(to.ENTRY_GET_KEY()).ENTRYSET(); }
    @Override
    public ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> tailSet(MAP.Entry KEY_VALUE_GENERIC
from) { return tailMap(from.ENTRY_GET_KEY()).ENTRYSET(); }
};

    return entries;
}

private class KeySet extends ABSTRACT_SORTED_MAP KEY_VALUE_GENERIC.KeySet {

```

```

@Override
public KEY_BIDI_ITERATOR KEY_GENERIC iterator() { return new SubmapKeyIterator(); }
@Override
public KEY_BIDI_ITERATOR KEY_GENERIC iterator(final KEY_GENERIC_TYPE from) { return new
SubmapKeyIterator(from); }
}

@Override
public SORTED_SET KEY_GENERIC keySet() {
if (keys == null) keys = new KeySet();
return keys;
}

@Override
public VALUE_COLLECTION VALUE_GENERIC values() {
if (values == null) values = new VALUE_ABSTRACT_COLLECTION VALUE_GENERIC() {
@Override
public VALUE_ITERATOR VALUE_GENERIC iterator() { return new SubmapValueIterator(); }
@Override
public boolean contains(final VALUE_TYPE k) { return containsValue(k); }
@Override
public int size() { return Submap.this.size();}
@Override
public void clear() { Submap.this.clear(); }
};

return values;
}

@Override
SUPPRESS_WARNINGS_KEY_UNCHECKED
public boolean containsKey(final KEY_TYPE k) {
return in(KEY_GENERIC_CAST k) && AVL_TREE_MAP.this.containsKey(k);
}

@Override
public boolean containsValue(final VALUE_TYPE v) {
final SubmapIterator i = new SubmapIterator();
VALUE_TYPE ev;

while(i.hasNext()) {
ev = i.nextEntry().value;
if (VALUE_EQUALS(ev, v)) return true;
}

return false;
}

```

```

@Override
SUPPRESS_WARNINGS_KEY_UNCHECKED
public VALUE_GENERIC_TYPE GET_VALUE(final KEY_TYPE k) {
    final AVL_TREE_MAP.Entry KEY_VALUE_GENERIC e;
    final KEY_GENERIC_TYPE kk = KEY_GENERIC_CAST k;
    return in(kk) && (e = findKey(kk)) != null ? e.value : this.defRetValue;
}

```

```

@Override
public VALUE_GENERIC_TYPE put(final KEY_GENERIC_TYPE k, final VALUE_GENERIC_TYPE v) {
    modified = false;
    if (! in(k)) throw new IllegalArgumentException("Key (" + k + ") out of range [" + (bottom ? "-" :
String.valueOf(from)) + ", " + (top ? "-" : String.valueOf(to)) + ")");
    final VALUE_GENERIC_TYPE oldValue = AVL_TREE_MAP.this.put(k, v);
    return modified ? this.defRetValue : oldValue;
}

```

```

@Override
SUPPRESS_WARNINGS_KEY_UNCHECKED
public VALUE_GENERIC_TYPE REMOVE_VALUE(final KEY_TYPE k) {
    modified = false;
    if (! in(KEY_GENERIC_CAST k)) return this.defRetValue;
    final VALUE_GENERIC_TYPE oldValue = AVL_TREE_MAP.this.REMOVE_VALUE(k);
    return modified ? oldValue : this.defRetValue;
}

```

```

@Override
public int size() {
    final SubmapIterator i = new SubmapIterator();
    int n = 0;

    while(i.hasNext()) {
        n++;
        i.nextEntry();
    }

    return n;
}

```

```

@Override
public boolean isEmpty() { return ! new SubmapIterator().hasNext(); }

```

```

@Override
public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() { return actualComparator; }

```

```

@Override
public SORTED_MAP KEY_VALUE_GENERIC headMap(final KEY_GENERIC_TYPE to) {
    if (top) return new Submap(from, bottom, to, false);
}

```

```

return compare(to, this.to) < 0 ? new Submap(from, bottom, to, false) : this;
}

@Override
public SORTED_MAP KEY_VALUE_GENERIC tailMap(final KEY_GENERIC_TYPE from) {
    if (bottom) return new Submap(from, false, to, top);
    return compare(from, this.from) > 0 ? new Submap(from, false, to, top) : this;
}

@Override
public SORTED_MAP KEY_VALUE_GENERIC subMap(KEY_GENERIC_TYPE from, KEY_GENERIC_TYPE
to) {
    if (top && bottom) return new Submap(from, false, to, false);
    if (! top) to = compare(to, this.to) < 0 ? to : this.to;
    if (! bottom) from = compare(from, this.from) > 0 ? from : this.from;
    if (! top && ! bottom && from == this.from && to == this.to) return this;
    return new Submap(from, false, to, false);
}

/** Locates the first entry.
 *
 * @return the first entry of this submap, or { @code null } if the submap is empty.
 */
public AVL_TREE_MAP.Entry KEY_VALUE_GENERIC firstEntry() {
    if (tree == null) return null;
    // If this submap goes to -infinity, we return the main map first entry; otherwise, we locate the start of the map.
    AVL_TREE_MAP.Entry KEY_VALUE_GENERIC e;
    if (bottom) e = firstEntry();
    else {
        e = locateKey(from);
        // If we find either the start or something greater we're OK.
        if (compare(e.key, from) < 0) e = e.next();
    }
    // Finally, if this subset doesn't go to infinity, we check that the resulting key isn't greater than the end.
    if (e == null || ! top && compare(e.key, to) >= 0) return null;
    return e;
}

/** Locates the last entry.
 *
 * @return the last entry of this submap, or { @code null } if the submap is empty.
 */
public AVL_TREE_MAP.Entry KEY_VALUE_GENERIC lastEntry() {
    if (tree == null) return null;
    // If this submap goes to infinity, we return the main map last entry; otherwise, we locate the end of the map.
    AVL_TREE_MAP.Entry KEY_VALUE_GENERIC e;
    if (top) e = lastEntry();
    else {

```

```

    e = locateKey(to);
    // If we find something smaller than the end we're OK.
    if (compare(e.key, to) >= 0) e = e.prev();
}
// Finally, if this subset doesn't go to -infinity, we check that the resulting key isn't smaller than the start.
if (e == null || ! bottom && compare(e.key, from) < 0) return null;
return e;
}

@Override
public KEY_GENERIC_TYPE FIRST_KEY() {
    AVL_TREE_MAP.Entry KEY_VALUE_GENERIC e = firstEntry();
    if (e == null) throw new NoSuchElementException();
    return e.key;
}

@Override
public KEY_GENERIC_TYPE LAST_KEY() {
    AVL_TREE_MAP.Entry KEY_VALUE_GENERIC e = lastEntry();
    if (e == null) throw new NoSuchElementException();
    return e.key;
}

/** An iterator for subranges.
 *
 * <p>This class inherits from { @link TreeIterator}, but overrides the methods that
 * update the pointer after a { @link java.util.ListIterator#next()} or { @link java.util.ListIterator#previous()}. If we
 would
 * move out of the range of the submap we just overwrite the next or previous
 * entry with { @code null}.
 */
private class SubmapIterator extends TreeIterator {
    SubmapIterator() { next = firstEntry(); }
    SubmapIterator(final KEY_GENERIC_TYPE k) {
        this();

        if (next != null) {
            if (! bottom && compare(k, next.key) < 0) prev = null;
            else if (! top && compare(k, (prev = lastEntry()).key) >= 0) next = null;
            else {
                next = locateKey(k);

                if (compare(next.key, k) <= 0) {
                    prev = next;
                    next = next.next();
                }
                else prev = next.prev();
            }
        }
    }
}

```

```

    }
}

@Override
void updatePrevious() {
    prev = prev.previous();
    if (! bottom && prev != null && AVL_TREE_MAP.this.compare(prev.key, from) < 0) prev = null;
}

@Override
void updateNext() {
    next = next.next();
    if (! top && next != null && AVL_TREE_MAP.this.compare(next.key, to) >= 0) next = null;
}
}

private class SubmapEntryIterator extends SubmapIterator implements ObjectListIterator<MAP.Entry
KEY_VALUE_GENERIC> {
    SubmapEntryIterator() {}
    SubmapEntryIterator(final KEY_GENERIC_TYPE k) { super(k); }

    @Override
    public MAP.Entry KEY_VALUE_GENERIC next() { return nextEntry(); }
    @Override
    public MAP.Entry KEY_VALUE_GENERIC previous() { return previousEntry(); }
}

/** An iterator on a subrange of keys.
 *
 * <p>This class can iterate in both directions on a subrange of the
 * keys of a threaded tree. We simply override the {@link
 * java.util.ListIterator#next()}/{@link java.util.ListIterator#previous()} methods (and possibly their
 * type-specific counterparts) so that they return keys instead of
 * entries.
 */
private final class SubmapKeyIterator extends SubmapIterator implements KEY_LIST_ITERATOR
KEY_GENERIC {
    public SubmapKeyIterator() { super(); }
    public SubmapKeyIterator(KEY_GENERIC_TYPE from) { super(from); }

    @Override
    public KEY_GENERIC_TYPE NEXT_KEY() { return nextEntry().key; }
    @Override
    public KEY_GENERIC_TYPE PREV_KEY() { return previousEntry().key; }
};

/** An iterator on a subrange of values.

```

```

*
* <p>This class can iterate in both directions on the values of a
* subrange of the keys of a threaded tree. We simply override the
* {@link java.util.ListIterator#next()}/{@link java.util.ListIterator#previous()} methods (and possibly their
* type-specific counterparts) so that they return values instead of
* entries.
*/
private final class SubmapValueIterator extends SubmapIterator implements VALUE_LIST_ITERATOR
VALUE_GENERIC {
    @Override
    public VALUE_GENERIC_TYPE NEXT_VALUE() { return nextEntry().value; }
    @Override
    public VALUE_GENERIC_TYPE PREV_VALUE() { return previousEntry().value; }
};
}

/** Returns a deep copy of this tree map.
*
* <p>This method performs a deep copy of this tree map; the data stored in the
* set, however, is not cloned. Note that this makes a difference only for object keys.
*
* @return a deep copy of this tree map.
*/

@Override
SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
public AVL_TREE_MAP KEY_VALUE_GENERIC clone() {
    AVL_TREE_MAP KEY_VALUE_GENERIC c;
    try {
        c = (AVL_TREE_MAP KEY_VALUE_GENERIC)super.clone();
    }
    catch(CloneNotSupportedException cantHappen) {
        throw new InternalError();
    }

    c.keys = null;
    c.values = null;
    c.entries = null;
    c.allocatePaths();

    if (count != 0) {
        // Also this apparently unfathomable code is derived from GNU libavl.
        Entry KEY_VALUE_GENERIC e, p, q, rp = new Entry KEY_VALUE_GENERIC_DIAMOND(), rq = new Entry
KEY_VALUE_GENERIC_DIAMOND();

        p = rp;
        rp.left(tree);

```

```

q = rq;
rq.pred(null);

while(true) {
if (! p.pred()) {
e = p.left.clone();
e.pred(q.left);
e.succ(q);
q.left(e);

p = p.left;
q = q.left;
}
else {
while(p.succ()) {
p = p.right;

if (p == null) {
q.right = null;
c.tree = rq.left;

c.firstEntry = c.tree;
while(c.firstEntry.left != null) c.firstEntry = c.firstEntry.left;
c.lastEntry = c.tree;
while(c.lastEntry.right != null) c.lastEntry = c.lastEntry.right;

return c;
}
q = q.right;
}

p = p.right;
q = q.right;
}

if (! p.succ()) {
e = p.right.clone();
e.succ(q.right);
e.pred(q);
q.right(e);
}
}
}

return c;
}

```

```

private void writeObject(java.io.ObjectOutputStream s) throws java.io.IOException {
    int n = count;
    EntryIterator i = new EntryIterator();
    Entry KEY_VALUE_GENERIC e;

    s.defaultWriteObject();

    while(n-- != 0) {
        e = i.nextEntry();
        s.WRITE_KEY(e.key);
        s.WRITE_VALUE(e.value);
    }
}

/** Reads the given number of entries from the input stream, returning the corresponding tree.
 *
 * @param s the input stream.
 * @param n the (positive) number of entries to read.
 * @param pred the entry containing the key that precedes the first key in the tree.
 * @param succ the entry containing the key that follows the last key in the tree.
 */
SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
private Entry KEY_VALUE_GENERIC readTree(final java.io.ObjectInputStream s, final int n, final Entry
KEY_VALUE_GENERIC pred, final Entry KEY_VALUE_GENERIC succ) throws java.io.IOException,
ClassNotFoundException {
    if (n == 1) {
        final Entry KEY_VALUE_GENERIC top = new Entry
KEY_VALUE_GENERIC_DIAMOND(KEY_GENERIC_CAST s.READ_KEY(), VALUE_GENERIC_CAST
s.READ_VALUE());
        top.pred(pred);
        top.succ(succ);

        return top;
    }

    if (n == 2) {
        /* We handle separately this case so that recursion will
        *always* be on nonempty subtrees. */
        final Entry KEY_VALUE_GENERIC top = new Entry
KEY_VALUE_GENERIC_DIAMOND(KEY_GENERIC_CAST s.READ_KEY(), VALUE_GENERIC_CAST
s.READ_VALUE());
        top.right(new Entry KEY_VALUE_GENERIC_DIAMOND(KEY_GENERIC_CAST s.READ_KEY(),
VALUE_GENERIC_CAST s.READ_VALUE()));
        top.right.pred(top);
        top.balance(1);
        top.pred(pred);
        top.right.succ(succ);
    }
}

```

```

return top;
}

// The right subtree is the largest one.
final int rightN = n / 2, leftN = n - rightN - 1;

final Entry KEY_VALUE_GENERIC top = new Entry KEY_VALUE_GENERIC_DIAMOND();

top.left(readTree(s, leftN, pred, top));

top.key = KEY_GENERIC_CAST s.READ_KEY();
top.value = VALUE_GENERIC_CAST s.READ_VALUE();

top.right(readTree(s, rightN, top, succ));

if (n == (n & -n)) top.balance(1); // Quick test for determining whether n is a power of 2.

return top;
}

private void readObject(java.io.ObjectInputStream s) throws java.io.IOException, ClassNotFoundException {
s.defaultReadObject();
/* The storedComparator is now correctly set, but we must restore
on-the-fly the actualComparator. */
setActualComparator();
allocatePaths();

if (count != 0) {
tree = readTree(s, count, null, null);
Entry KEY_VALUE_GENERIC e;

e = tree;
while(e.left() != null) e = e.left();
firstEntry = e;

e = tree;
while(e.right() != null) e = e.right();
lastEntry = e;
}
}

#ifdef ASSERTS_CODE
private static KEY_VALUE_GENERIC int checkTree(Entry KEY_VALUE_GENERIC e) {
if (e == null) return 0;

```

```

final int leftN = checkTree(e.left()), rightN = checkTree(e.right());
if (leftN + e.balance() != rightN)
    throw new AssertionError("Mismatch between left tree size (" + leftN + "), right tree size (" + rightN + ") and
balance (" + e.balance() + ")");

return Math.max(leftN, rightN) + 1;
}
#endif

#ifdef TEST

private static long seed = System.currentTimeMillis();
private static java.util.Random r = new java.util.Random(seed);

private static KEY_TYPE genKey() {
#if KEY_CLASS_Byte || KEY_CLASS_Short || KEY_CLASS_Character
return (KEY_TYPE)(r.nextInt());
#elif KEYS_PRIMITIVE
return r.NEXT_KEY();
#else
return Integer.toBinaryString(r.nextInt());
#endif
}

private static VALUE_TYPE genValue() {
#if VALUE_CLASS_Byte || VALUE_CLASS_Short || VALUE_CLASS_Character
return (VALUE_TYPE)(r.nextInt());
#elif VALUES_PRIMITIVE
return r.NEXT_VALUE();
#elif !VALUE_CLASS_Reference || KEY_CLASS_Reference
return Integer.toBinaryString(r.nextInt());
#else
return new java.io.Serializable() {};
#endif
}

private static java.text.NumberFormat format = new java.text.DecimalFormat("#,###.00");
private static java.text.FieldPosition p = new java.text.FieldPosition(0);

private static String format(double d) {
StringBuffer s = new StringBuffer();
return format.format(d, s, p).toString();
}

private static void speedTest(int n, boolean comp) {
int i, j;

```

```

AVL_TREE_MAP m;
java.util.TreeMap t;
KEY_TYPE k[] = new KEY_TYPE[n];
KEY_TYPE nk[] = new KEY_TYPE[n];
VALUE_TYPE v[] = new VALUE_TYPE[n];
long ms;

for(i = 0; i < n; i++) {
    k[i] = genKey();
    nk[i] = genKey();
    v[i] = genValue();
}

double totPut = 0, totYes = 0, totNo = 0, totAddTo = 0, totIterFor = 0, totIterBack = 0, totRemYes = 0, d, dd, ddd;

if (comp) { for(j = 0; j < 20; j++) {

    t = new java.util.TreeMap();

    /* We first add all pairs to t. */
    for(i = 0; i < n; i++) t.put(KEY2OBJ(k[i]), VALUE2OBJ(v[i]));

    /* Then we remove the first half and put it back. */
    for(i = 0; i < n/2; i++) t.remove(KEY2OBJ(k[i]));

    ms = System.currentTimeMillis();
    for(i = 0; i < n/2; i++) t.put(KEY2OBJ(k[i]), VALUE2OBJ(v[i]));
    d = System.currentTimeMillis() - ms;

    /* Then we remove the other half and put it back again. */
    ms = System.currentTimeMillis();
    for(i = n/2; i < n; i++) t.remove(KEY2OBJ(k[i]));
    dd = System.currentTimeMillis() - ms ;

    ms = System.currentTimeMillis();
    for(i = n/2; i < n; i++) t.put(KEY2OBJ(k[i]), VALUE2OBJ(v[i]));
    d += System.currentTimeMillis() - ms;
    if (j > 2) totPut += n/d;
    System.out.print("Add: " + format(n/d) + " K/s ");

    /* Then we remove again the first half. */
    ms = System.currentTimeMillis();
    for(i = 0; i < n/2; i++) t.remove(KEY2OBJ(k[i]));
    dd += System.currentTimeMillis() - ms ;
    if (j > 2) totRemYes += n/dd;
    System.out.print("RemYes: " + format(n/dd) + " K/s ");

    /* And then we put it back. */

```

```

for(i = 0; i < n/2; i++) t.put(KEY2OBJ(k[i]), VALUE2OBJ(v[i]));

#if VALUES_PRIMITIVE && !VALUE_CLASS_Boolean
/* we perform n/2 addTo() operations with get then put */
ms = System.currentTimeMillis();
for(i = 0; i < n/2; i++) t.put(KEY2OBJ(k[i]), (VALUE_TYPE)((VALUE_CLASS) t.get(KEY2OBJ(k[i])) + i));
ddd = System.currentTimeMillis() - ms;
if (j > 2) totAddTo += n/ddd;
System.out.print("AddTo: " + format(n/ddd) + " K/s ");
#endif

/* We check for pairs in t. */
ms = System.currentTimeMillis();
for(i = 0; i < n; i++) t.containsKey(KEY2OBJ(k[i]));
d = 1.0 * n / (System.currentTimeMillis() - ms);
if (j > 2) totYes += d;
System.out.print("Yes: " + format(d) + " K/s ");

/* We check for pairs not in t. */
ms = System.currentTimeMillis();
for(i = 0; i < n; i++) t.containsKey(KEY2OBJ(nk[i]));
d = 1.0 * n / (System.currentTimeMillis() - ms);
if (j > 2) totNo += d;
System.out.print("No: " + format(d) + " K/s ");

/* We iterate on t. */
ms = System.currentTimeMillis();
for(Iterator it = t.entrySet().iterator(); it.hasNext(); it.next());
d = 1.0 * n / (System.currentTimeMillis() - ms);
if (j > 2) totIterFor += d;
System.out.print("IterFor: " + format(d) + " K/s ");

System.out.println();
}

System.out.println();
System.out.println("java.util Put: " + format(totPut/(j-3)) + " K/s RemYes: " + format(totRemYes/(j-3)) + " K/s Yes:
" + format(totYes/(j-3)) + " K/s No: " + format(totNo/(j-3)) + " K/s AddTo: " + format(totAddTo/(j-3)) + " K/s
IterFor: " + format(totIterFor/(j-3)) + " K/s");

System.out.println();

t = null;
totPut = totYes = totNo = totIterFor = totIterBack = totRemYes = totAddTo = 0;

}

for(j = 0; j < 20; j++) {

```

```

m = new AVL_TREE_MAP();

/* We first add all pairs to m. */
for(i = 0; i < n; i++) m.put(k[i], v[i]);

/* Then we remove the first half and put it back. */
for(i = 0; i < n/2; i++) m.remove(k[i]);

ms = System.currentTimeMillis();
for(i = 0; i < n/2; i++) m.put(k[i], v[i]);
d = System.currentTimeMillis() - ms;

/* Then we remove the other half and put it back again. */
ms = System.currentTimeMillis();
for(i = n/2; i < n; i++) m.remove(k[i]);
dd = System.currentTimeMillis() - ms ;

ms = System.currentTimeMillis();
for(i = n/2; i < n; i++) m.put(k[i], v[i]);
d += System.currentTimeMillis() - ms;
if (j > 2) totPut += n/d;
System.out.print("Add: " + format(n/d) + " K/s ");

/* Then we remove again the first half. */
ms = System.currentTimeMillis();
for(i = 0; i < n/2; i++) m.remove(k[i]);
dd += System.currentTimeMillis() - ms ;
if (j > 2) totRemYes += n/dd;
System.out.print("RemYes: " + format(n/dd) + " K/s ");

/* And then we put it back. */
for(i = 0; i < n/2; i++) m.put(k[i], v[i]);

#if VALUES_PRIMITIVE && !VALUE_CLASS_Boolean
/* we perform n/2 addTo() operations with get then put */
ms = System.currentTimeMillis();
for(i = 0; i < n/2; i++) m.addTo(k[i], (VALUE_TYPE) i);
ddd = System.currentTimeMillis() - ms;
if (j > 2) totAddTo += n/ddd;
System.out.print("AddTo: " + format(n/ddd) + " K/s ");
#endif

/* We check for pairs in m. */
ms = System.currentTimeMillis();
for(i = 0; i < n; i++) m.containsKey(k[i]);
d = 1.0 * n / (System.currentTimeMillis() - ms);
if (j > 2) totYes += d;

```

```

System.out.print("Yes: " + format(d) + " K/s ");

/* We check for pairs not in m. */
ms = System.currentTimeMillis();
for(i = 0; i < n; i++) m.containsKey(nk[i]);
d = 1.0 * n / (System.currentTimeMillis() - ms);
if (j > 2) totNo += d;
System.out.print("No: " + format(d) + " K/s ");

/* We iterate on m. */
java.util.ListIterator it = (java.util.ListIterator)m.entrySet().iterator();
ms = System.currentTimeMillis();
for(it = (java.util.ListIterator)m.entrySet().iterator(); it.hasNext(); it.next());
d = 1.0 * n / (System.currentTimeMillis() - ms);
if (j > 2) totIterFor += d;
System.out.print("IterFor: " + format(d) + " K/s ");

/* We iterate back on m. */
ms = System.currentTimeMillis();
for(; it.hasPrevious(); it.previous());
d = 1.0 * n / (System.currentTimeMillis() - ms);
if (j > 2) totIterBack += d;
System.out.print("IterBack: " + format(d) + " K/s ");

System.out.println();
}

System.out.println();
System.out.println("fastutil Put: " + format(totPut/(j-3)) + " K/s RemYes: " + format(totRemYes/(j-3)) + " K/s Yes:
" + format(totYes/(j-3)) + " K/s No: " + format(totNo/(j-3))+ "K/s AddTo: " + format(totAddTo/(j-3)) + " K/s
IterFor: " + format(totIterFor/(j-3)) + " K/s");

System.out.println();

}

private static boolean valEquals(Object o1, Object o2) {
return o1 == null ? o2 == null : o1.equals(o2);
}

private static void fatal(String msg) {
System.out.println(msg);
System.exit(1);
}

```

```

private static void ensure(boolean cond, String msg) {
    if (cond) return;
    fatal(msg);
}

private static void compareMT(SORTED_MAP m, SortedMap t, int level, long seed) {
    /* Now we check that m and t are equal. */
    if (!m.equals(t) || !t.equals(m)) System.err.println("m: " + m + " t: " + t);

    ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) at start");
    ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) at start");

    /* Now we check that m actually holds that data. */
    for(Iterator i=t.entrySet().iterator(); i.hasNext();) {
        java.util.Map.Entry e = (java.util.Map.Entry)i.next();
        ensure(valEquals(e.getValue(), m.get(e.getKey())), "Error (" + level + ", " + seed + "): m and t differ on an entry
("+e+") after insertion (iterating on t)");
    }

    /* Now we check that m actually holds that data, but iterating on m. */
    for(Iterator i=m.entrySet().iterator(); i.hasNext();) {
        Entry e = (Entry)i.next();
        ensure(valEquals(e.getValue(), t.get(e.getKey())), "Error (" + level + ", " + seed + "): m and t differ on an entry
("+e+") after insertion (iterating on m)");
    }

    /* Now we check that m actually holds the same keys. */
    for(Iterator i=t.keySet().iterator(); i.hasNext();) {
        Object o = i.next();
        ensure(m.containsKey(o), "Error (" + level + ", " + seed + "): m and t differ on a key (" + o + ") after insertion
(iterating on t)");
        ensure(m.keySet().contains(o), "Error (" + level + ", " + seed + "): m and t differ on a key (" + o + ", in keySet()) after
insertion (iterating on t)");
    }

    /* Now we check that m actually holds the same keys, but iterating on m. */
    for(Iterator i=m.keySet().iterator(); i.hasNext();) {
        Object o = i.next();
        ensure(t.containsKey(o), "Error (" + level + ", " + seed + "): m and t differ on a key after insertion (iterating on m)");
        ensure(t.keySet().contains(o), "Error (" + level + ", " + seed + "): m and t differ on a key (in keySet()) after insertion
(iterating on m)");
    }

    /* Now we check that m actually hold the same values. */
    for(Iterator i=t.values().iterator(); i.hasNext();) {

```

```

    Object o = i.next();
    ensure(m.containsValue(o), "Error (" + level + ", " + seed + "): m and t differ on a value after insertion (iterating on
t)");
    ensure(m.values().contains(o), "Error (" + level + ", " + seed + "): m and t differ on a value (in values()) after
insertion (iterating on t)");
}

/* Now we check that m actually hold the same values, but iterating on m. */
for(Iterator i=m.values().iterator(); i.hasNext(); ) {
    Object o = i.next();
    ensure(t.containsValue(o), "Error (" + level + ", " + seed + "): m and t differ on a value after insertion (iterating on
m)");
    ensure(t.values().contains(o), "Error (" + level + ", " + seed + "): m and t differ on a value (in values()) after
insertion (iterating on m)");
}

}

private static Object[] k, v, nk;
private static KEY_TYPE kt[];
private static KEY_TYPE nkt[];
private static VALUE_TYPE vt[];
private static AVL_TREE_MAP topMap;

protected static void testMaps(SORTED_MAP m, SortedMap t, int n, int level) {
    long ms;
    boolean mThrowsIllegal, tThrowsIllegal, mThrowsNoElement, tThrowsNoElement;
    Object rt = null, rm = null;

    if (level > 4) return;

    /* Now we check that both maps agree on first/last keys. */

    mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

    try {
        m.firstKey();
    }
    catch (NoSuchElementException e) { mThrowsNoElement = true; }
    try {
        t.firstKey();
    }
    catch (NoSuchElementException e) { tThrowsNoElement = true; }

    ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): firstKey() divergence at
start in NoSuchElementException (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");
    if (! mThrowsNoElement) ensure(t.firstKey().equals(m.firstKey()), "Error (" + level + ", " + seed + "): m and t differ

```

```

at start on their first key (" + m.firstKey() + ", " + t.firstKey() +");

mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

try {
    m.lastKey();
}
catch (NoSuchElementException e) { mThrowsNoElement = true; }
try {
    t.lastKey();
}
catch (NoSuchElementException e) { tThrowsNoElement = true; }

ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): lastKey() divergence at start
in NoSuchElementException (" + mThrowsNoElement + ", " + tThrowsNoElement + "));

if (! mThrowsNoElement) ensure(t.lastKey().equals(m.lastKey()), "Error (" + level + ", " + seed + "): m and t differ
at start on their last key (" + m.lastKey() + ", " + t.lastKey() + "));

compareMT(m, t, level, seed);

/* Now we check that inquiries about random data give the same answer in m and t. For
m we use the polymorphic method. */

for(int i=0; i<n; i++) {
    KEY_TYPE T = genKey();

    mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

    try {
        m.containsKey(KEY2OBJ(T));
    }
    catch (NoSuchElementException e) { mThrowsNoElement = true; }
    catch (IllegalArgumentException e) { mThrowsIllegal = true; }

    try {
        t.containsKey(KEY2OBJ(T));
    }
    catch (NoSuchElementException e) { tThrowsNoElement = true; }
    catch (IllegalArgumentException e) { tThrowsIllegal = true; }

    ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): containsKey() divergence in
NoSuchElementException (" + mThrowsNoElement + ", " + tThrowsNoElement + "));
    ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): containsKey() divergence in
IllegalArgumentException (" + mThrowsIllegal + ", " + tThrowsIllegal + "));
    if (!mThrowsNoElement && !mThrowsIllegal) {
        ensure(m.containsKey(KEY2OBJ(T)) == t.containsKey(KEY2OBJ(T)), "Error (" + level + ", " + seed + "):

```

```

divergence in keys between t and m (polymorphic method)");

#if KEY_CLASS_Object && ! (VALUES_REFERENCE)
    if ((m.GET_VALUE(T) != VALUE_NULL) != ((t.get(KEY2OBJ(T)) == null ? VALUE_NULL :
VALUE_OBJ2TYPE(t.get(KEY2OBJ(T)))) != VALUE_NULL) ||
        t.get(KEY2OBJ(T)) != null &&
        ! VALUE2OBJ(m.GET_VALUE(T)).equals(t.get(KEY2OBJ(T))))
#else
    if ((m.get(T) != VALUE_NULL) != ((t.get(KEY2OBJ(T)) == null ? VALUE_NULL :
VALUE_OBJ2TYPE(t.get(KEY2OBJ(T)))) != VALUE_NULL) ||
        t.get(KEY2OBJ(T)) != null &&
        ! m.get(KEY2OBJ(T)).equals(t.get(KEY2OBJ(T))))
#endif
    {
        System.out.println("Error (" + level + ", " + seed + "): divergence between t and m (polymorphic method)");
        System.exit(1);
    }
}

/* Again, we check that inquiries about random data give the same answer in m and t, but
for m we use the standard method. */

for(int i=0; i<n; i++) {
    KEY_TYPE T = genKey();
    mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

    try {
        m.get(KEY2OBJ(T));
    }
    catch (NoSuchElementException e) { mThrowsNoElement = true; }
    catch (IllegalArgumentException e) { mThrowsIllegal = true; }

    try {
        t.get(KEY2OBJ(T));
    }
    catch (NoSuchElementException e) { tThrowsNoElement = true; }
    catch (IllegalArgumentException e) { tThrowsIllegal = true; }

    ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): get() divergence in
NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");
    ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): get() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ")");
    if (!mThrowsNoElement && !mThrowsIllegal) ensure(valEquals(m.get(KEY2OBJ(T)), t.get(KEY2OBJ(T))),
"Error (" + level + ", " + seed + "): divergence between t and m (standard method)");
}

/* Now we put and remove random data in m and t, checking that the result is the same. */

```

```
for(int i=0; i<20*n; i++) {  
    KEY_TYPE T = genKey();  
    VALUE_TYPE U = genValue();
```

```
mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;
```

```
try {  
    rm = m.put(KEY2OBJ(T), VALUE2OBJ(U));  
}  
catch (NoSuchElementException e) { mThrowsNoElement = true; }  
catch (IllegalArgumentException e) { mThrowsIllegal = true; }
```

```
try {  
    rt = t.put(KEY2OBJ(T), VALUE2OBJ(U));  
}  
catch (NoSuchElementException e) { tThrowsNoElement = true; }  
catch (IllegalArgumentException e) { tThrowsIllegal = true; }
```

```
ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): put() divergence in  
NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + "));  
ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): put() divergence in  
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + "));  
if (!mThrowsNoElement && !mThrowsIllegal) ensure(valEquals(rm, rt), "Error (" + level + ", " + seed + "):  
divergence in put() between t and m (" + rt + ", " + rm + "));
```

```
T = genKey();
```

```
mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;
```

```
try {  
    rm = m.remove(KEY2OBJ(T));  
}  
catch (NoSuchElementException e) { mThrowsNoElement = true; }  
catch (IllegalArgumentException e) { mThrowsIllegal = true; }
```

```
try {  
    rt = t.remove(KEY2OBJ(T));  
}  
catch (NoSuchElementException e) { tThrowsNoElement = true; }  
catch (IllegalArgumentException e) { tThrowsIllegal = true; }
```

```
ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): remove() divergence in  
NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + "));  
ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): remove() divergence in
```

```

IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + "));
    if (!mThrowsNoElement && !mThrowsIllegal) ensure(valEquals(rm, rt), "Error (" + level + ", " + seed + "):
divergence in remove() between t and m (" + rt + ", " + rm + "));
}

compareMT(m, t, level, seed);

#if VALUES_PRIMITIVE && !VALUE_CLASS_Boolean
/* Now we check that addTo results in the same values as get/put */
if (m instanceof AVL_TREE_MAP) {
    for (Iterator i=m.entrySet().iterator(); i.hasNext(); ) {
        Entry e = (Entry)i.next();
        VALUE_TYPE incr = genValue();
        ((AVL_TREE_MAP) m).addTo(e.key, incr);
        t.put(e.key, (VALUE_TYPE) ((VALUE_CLASS) t.get(e.key) + incr));
    }
    compareMT(m, t, level, seed);

    /* Now we make sure that addTo works with a key that did not previously exist (with special default return value)*/
    KEY_TYPE newKey;

    do {
        newKey = genKey();
    } while (m.containsKey(newKey));

    VALUE_TYPE newValue, drv;
    drv = m.defaultReturnValue();
    newValue = genValue();
    m.defaultReturnValue(genValue());

    ((AVL_TREE_MAP) m).addTo(newKey, newValue);
    t.put(newKey, (VALUE_TYPE) (newValue + m.defaultReturnValue()));

    compareMT(m, t, level, seed);
    m.defaultReturnValue(drv); // set the default value back to the old drv
}
#endif

/* Now we check that both maps agree on first/last keys. */

mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

try {
    m.firstKey();
}
catch (NoSuchElementException e) { mThrowsNoElement = true; }
try {
    t.firstKey();
}

```

```

    }
    catch (NoSuchElementException e) { tThrowsNoElement = true; }

    ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): firstKey() divergence in
NoSuchElementException (" + mThrowsNoElement + ", " + tThrowsNoElement + "));
    if (! mThrowsNoElement) ensure(t.firstKey().equals(m.firstKey()), "Error (" + level + ", " + seed + "): m and t differ
on their first key (" + m.firstKey() + ", " + t.firstKey() + "));

    mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

    try {
        m.lastKey();
    }
    catch (NoSuchElementException e) { mThrowsNoElement = true; }
    try {
        t.lastKey();
    }
    catch (NoSuchElementException e) { tThrowsNoElement = true; }

    ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): lastKey() divergence in
NoSuchElementException (" + mThrowsNoElement + ", " + tThrowsNoElement + "));

    if (! mThrowsNoElement) ensure(t.lastKey().equals(m.lastKey()), "Error (" + level + ", " + seed + "): m and t differ
on their last key (" + m.lastKey() + ", " + t.lastKey() + "));

    /* Now we check cloning. */

    if (level == 0) {
        ensure(m.equals(((AVL_TREE_MAP)m).clone()), "Error (" + level + ", " + seed + "): m does not equal m.clone()");
        ensure(((AVL_TREE_MAP)m).clone().equals(m), "Error (" + level + ", " + seed + "): m.clone() does not equal m");
    }

    int h = m.hashCode();

    /* Now we save and read m. */

    SORTED_MAP m2 = null;

    try {
        java.io.File ff = new java.io.File("it.unimi.dsi.fastutil.test");
        java.io.OutputStream os = new java.io.FileOutputStream(ff);
        java.io.ObjectOutputStream oos = new java.io.ObjectOutputStream(os);

        oos.writeObject(m);
        oos.close();

        java.io.InputStream is = new java.io.FileInputStream(ff);

```

```

java.io.ObjectInputStream ois = new java.io.ObjectInputStream(is);

m2 = (SORTED_MAP)ois.readObject();
ois.close();
ff.delete();
}
catch(Exception e) {
    e.printStackTrace();
    System.exit(1);
}

#if !VALUE_CLASS_Reference
ensure(m2.hashCode() == h, "Error (" + level + ", " + seed + "): hashCode() changed after save/read");

/* Now we check that m2 actually holds that data. */

ensure(m2.equals(t), "Error (" + level + ", " + seed + "): ! m2.equals(t) after save/read");
ensure(t.equals(m2), "Error (" + level + ", " + seed + "): ! t.equals(m2) after save/read");
#else
m2.clear();
m2.putAll(m);
#endif

/* Now we take out of m everything, and check that it is empty. */

for(Iterator i=t.keySet().iterator(); i.hasNext();) m2.remove(i.next());

ensure(m2.isEmpty(), "Error (" + level + ", " + seed + "): m2 is not empty (as it should be)");

/* Now we play with iterators. */

{
    java.util.ListIterator i, j;
    Map.Entry E, F;
    Object J;
    i = (java.util.ListIterator)m.entrySet().iterator();
    j = new java.util.LinkedList(t.entrySet()).listIterator();

    for(int k = 0; k < 2*n; k++) {
        ensure(i.hasNext() == j.hasNext(), "Error (" + level + ", " + seed + "): divergence in hasNext()");
        ensure(i.hasPrevious() == j.hasPrevious(), "Error (" + level + ", " + seed + "): divergence in hasPrevious()");

        if (r.nextFloat() < .8 && i.hasNext()) {
            ensure((E=(Entry)i.next()).getKey().equals(J = (F=(Map.Entry)j.next()).getKey()), "Error (" + level + ", " + seed +
"): divergence in next()");
        }

        if (r.nextFloat() < 0.3) {
            i.remove();
        }
    }
}

```

```

j.remove();
t.remove(J);
}
else if (r.nextFloat() < 0.3) {
    Object U = VALUE2OBJ(genValue());
    E.setValue(U);
    t.put(F.getKey(), U);
}
}
else if (r.nextFloat() < .2 && i.hasPrevious()) {
    ensure((E=(Entry)i.previous()).getKey().equals(J = (F=(Map.Entry)j.previous()).getKey()), "Error (" + level + ", " +
seed + "): divergence in previous()");

    if (r.nextFloat() < 0.3) {
        i.remove();
        j.remove();
        t.remove(J);
    }
    else if (r.nextFloat() < 0.3) {
        Object U = VALUE2OBJ(genValue());
        E.setValue(U);
        t.put(F.getKey(), U);
    }
}

ensure(i.nextIndex() == j.nextIndex(), "Error (" + level + ", " + seed + "): divergence in nextIndex()");
ensure(i.previousIndex() == j.previousIndex(), "Error (" + level + ", " + seed + "): divergence in previousIndex()");

}

}

{
    boolean badPrevious = false;
    Object previous = null;
    it.unimi.dsi.fastutil.BidirectionalIterator i;
    java.util.ListIterator j;
    Object I, J;
    KEY_TYPE from = genKey();
    j = new java.util.LinkedList(t.keySet()).listIterator();
    while(j.hasNext()) {
        Object k = j.next();
        if (((Comparable)k).compareTo(KEY2OBJ(from)) > 0) {
            badPrevious = true;
            j.previous();
            break;
        }
    }
}

```

```

    previous = k;
}

i = (it.unimi.dsi.fastutil.BidirectionalIterator)((SORTED_SET)m.keySet()).iterator(from);

for(int k = 0; k < 2*n; k++) {
    ensure(i.hasNext() == j.hasNext(), "Error (" + level + ", " + seed + "): divergence in hasNext() (iterator with starting
point " + from + ")");
    ensure(i.hasPrevious() == j.hasPrevious() || badPrevious && (i.hasPrevious() == (previous != null)), "Error (" +
level + ", " + seed + "): divergence in hasPrevious() (iterator with starting point " + from + ")" + badPrevious);

    if (r.nextFloat() < .8 && i.hasNext()) {
        ensure((I = i.next()).equals(J = j.next()), "Error (" + level + ", " + seed + "): divergence in next() (" + I + ", " + J + ",
iterator with starting point " + from + ")");
        //System.err.println("Done next " + I + " " + J + " " + badPrevious);

        badPrevious = false;

        if (r.nextFloat() < 0.5) {
            //System.err.println("Removing in next");
            i.remove();
            j.remove();
            t.remove(J);
        }
    }
    else if (!badPrevious && r.nextFloat() < .2 && i.hasPrevious()) {
        ensure((I = i.previous()).equals(J = j.previous()), "Error (" + level + ", " + seed + "): divergence in previous() (" + I +
", " + J + ", iterator with starting point " + from + ")");

        if (r.nextFloat() < 0.5) {
            //System.err.println("Removing in prev");
            i.remove();
            j.remove();
            t.remove(J);
        }
    }
}

/* Now we check that m actually holds that data. */

ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after iteration");
ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after iteration");

/* Now we select a pair of keys and create a submap. */

if (! m.isEmpty()) {

```

```

java.util.ListIterator i;
Object start = m.firstKey(), end = m.firstKey();
for(i = (java.util.ListIterator)m.keySet().iterator(); i.hasNext() && r.nextFloat() < .3; start = end = i.next());
for(; i.hasNext() && r.nextFloat() < .95; end = i.next());

//System.err.println("Checking subMap from " + start + " to " + end + " (level=" + (level+1) + ")...");
testMaps((SORTED_MAP)m.subMap((KEY_CLASS)start, (KEY_CLASS)end), t.subMap(start, end), n, level + 1);

ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after subMap");
ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after subMap");

//System.err.println("Checking headMap to " + end + " (level=" + (level+1) + ")...");
testMaps((SORTED_MAP)m.headMap((KEY_CLASS)end), t.headMap(end), n, level + 1);

ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after headMap");
ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after headMap");

//System.err.println("Checking tailMap from " + start + " (level=" + (level+1) + ")...");
testMaps((SORTED_MAP)m.tailMap((KEY_CLASS)start), t.tailMap(start), n, level + 1);

ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after tailMap");
ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after tailMap");
}

}

private static void runTest(int n) {
    AVL_TREE_MAP m = new AVL_TREE_MAP();
    SortedMap t = new java.util.TreeMap();
    topMap = m;
    k = new Object[n];
    v = new Object[n];
    nk = new Object[n];
    kt = new KEY_TYPE[n];
    nkt = new KEY_TYPE[n];
    vt = new VALUE_TYPE[n];

    for(int i = 0; i < n; i++) {
    #if KEY_CLASS_Object
        k[i] = kt[i] = genKey();
        nk[i] = nkt[i] = genKey();
    #else
        k[i] = new KEY_CLASS(kt[i] = genKey());
        nk[i] = new KEY_CLASS(nkt[i] = genKey());
    #endif
    #if VALUES_REFERENCE

```

```

    v[i] = vt[i] = genValue();
#else
    v[i] = new VALUE_CLASS(vt[i] = genValue());
#endif
}

/* We add pairs to t. */
for(int i = 0; i < n; i++) t.put(k[i], v[i]);

/* We add to m the same data */
m.putAll(t);

testMaps(m, t, n, 0);

System.out.println("Test OK");
return;
}

public static void main(String args[]) {
    int n = Integer.parseInt(args[1]);
    if (args.length > 2) r = new java.util.Random(seed = Long.parseLong(args[2]));

    try {
        if ("speedTest".equals(args[0]) || "speedComp".equals(args[0])) speedTest(n, "speedComp".equals(args[0]));
        else if ("test".equals(args[0])) runTest(n);
    } catch(Throwable e) {
        e.printStackTrace(System.err);
        System.err.println("seed: " + seed);
    }
}

#endif

}

```

Found in path(s):

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/drv/AVLTreeMap.drv

No license file was found, but licenses were detected in source scan.

distributed under the Apache License 2.0.

Found in path(s):

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/src/overview.html

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2004-2017 Sebastiano Vigna
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

#if KEY_CLASS_Byte

// HORRIBLE kluges to work around bug #6478546

private static final int MAX_IO_LENGTH = 1024 * 1024;

private static int read(final InputStream is, final byte a[], final int offset, final int length) throws IOException {
    if (length == 0) return 0;

    int read = 0, result;
    do {
        result = is.read(a, offset + read, Math.min(length - read, MAX_IO_LENGTH));
        if (result < 0) return read;
        read += result;
    } while(read < length);

    return read;
}

private static void write(final OutputStream outputStream, final byte a[], final int offset, final int length) throws
IOException {
    int written = 0;
    while(written < length) {
        outputStream.write(a, offset + written, Math.min(length - written, MAX_IO_LENGTH));
        written += Math.min(length - written, MAX_IO_LENGTH);
    }
}
}
```

```

private static void write(final DataOutput dataOutput, final byte a[], final int offset, final int length) throws
IOException {
    int written = 0;
    while(written < length) {
        dataOutput.write(a, offset + written, Math.min(length - written, MAX_IO_LENGTH));
        written += Math.min(length - written, MAX_IO_LENGTH);
    }
}

// Additional read/write methods to work around the DataInput/DataOutput schizophrenia.

/** Loads bytes from a given input stream, storing them in a given array fragment.
 *
 * <p>Note that this method is going to be significantly faster than { @link #loadBytes(DataInput,byte[],int,int) }
 * as it uses { @link InputStream}'s bulk-read methods.
 *
 * @param inputStream an input stream.
 * @param array an array which will be filled with data from { @code inputStream }.
 * @param offset the index of the first element of { @code array } to be filled.
 * @param length the number of elements of { @code array } to be filled.
 * @return the number of elements actually read from { @code inputStream } (it might be less than { @code length } if
 * { @code inputStream } ends).
 */
public static int LOAD_KEYS(final InputStream inputStream, final KEY_TYPE[] array, final int offset, final int
length) throws IOException {
    return read(inputStream, array, offset, length);
}

/** Loads bytes from a given input stream, storing them in a given array.
 *
 * <p>Note that this method is going to be significantly faster than { @link #loadBytes(DataInput,byte[]) }
 * as it uses { @link InputStream}'s bulk-read methods.
 *
 * @param inputStream an input stream.
 * @param array an array which will be filled with data from { @code inputStream }.
 * @return the number of elements actually read from { @code inputStream } (it might be less than the array length if
 * { @code inputStream } ends).
 */
public static int LOAD_KEYS(final InputStream inputStream, final KEY_TYPE[] array) throws IOException {
    return read(inputStream, array, 0, array.length);
}

/** Stores an array fragment to a given output stream.
 *
 * <p>Note that this method is going to be significantly faster than { @link #storeBytes(byte[],int,int,DataOutput) }
 * as it uses { @link OutputStream}'s bulk-read methods.
 *
 * @param array an array whose elements will be written to { @code outputStream }.

```

```

* @param offset the index of the first element of { @code array } to be written.
* @param length the number of elements of { @code array } to be written.
* @param outputStream an output stream.
*/
public static void STORE_KEYS(final KEY_TYPE array[], final int offset, final int length, final OutputStream
outputStream) throws IOException {
    write(outputStream, array, offset, length);
}

/** Stores an array to a given output stream.
*
* <p>Note that this method is going to be significantly faster than { @link #storeBytes(byte[],DataOutput)}
* as it uses { @link OutputStream}'s bulk-read methods.
*
* @param array an array whose elements will be written to { @code outputStream}.
* @param outputStream an output stream.
*/
public static void STORE_KEYS(final KEY_TYPE array[], final OutputStream outputStream) throws IOException
{
    write(outputStream, array, 0, array.length);
}

private static long read(final InputStream is, final byte a[][], final long offset, final long length) throws IOException
{
    if (length == 0) return 0;

    long read = 0;
    int segment = segment(offset);
    int displacement = displacement(offset);
    int result;
    do {
        result = is.read(a[segment], displacement, (int)Math.min(a[segment].length - displacement, Math.min(length - read,
MAX_IO_LENGTH)));
        if (result < 0) return read;
        read += result;
        displacement += result;
        if (displacement == a[segment].length) {
            segment++;
            displacement = 0;
        }
    } while (read < length);

    return read;
}

private static void write(final OutputStream outputStream, final byte a[][], final long offset, final long length) throws
IOException {

```

```

if (length == 0) return;
long written = 0;
int toWrite;
int segment = segment(offset);
int displacement = displacement(offset);
do {
    toWrite = (int)Math.min(a[segment].length - displacement, Math.min(length - written, MAX_IO_LENGTH));
    outputStream.write(a[segment], displacement, toWrite);
    written += toWrite;
    displacement += toWrite;
    if (displacement == a[segment].length) {
        segment++;
        displacement = 0;
    }
} while(written < length);
}

```

```

private static void write(final DataOutput dataOutput, final byte a[], final long offset, final long length) throws
IOException {
    if (length == 0) return;
    long written = 0;
    int toWrite;
    int segment = segment(offset);
    int displacement = displacement(offset);
    do {
        toWrite = (int)Math.min(a[segment].length - displacement, Math.min(length - written, MAX_IO_LENGTH));
        dataOutput.write(a[segment], displacement, toWrite);
        written += toWrite;
        displacement += toWrite;
        if (displacement == a[segment].length) {
            segment++;
            displacement = 0;
        }
    } while(written < length);
}

```

// Additional read/write methods to work around the DataInput/DataOutput schizophrenia.

/** Loads bytes from a given input stream, storing them in a given big-array fragment.

*

* <p>Note that this method is going to be significantly faster than { @link

#loadBytes(DataInput,byte[],long,long)}

* as it uses { @link InputStream}'s bulk-read methods.

*

* @param inputStream an input stream.

* @param array a big array which will be filled with data from { @code inputStream}.

* @param offset the index of the first element of { @code array} to be filled.

```

* @param length the number of elements of { @code array } to be filled.
* @return the number of elements actually read from { @code inputStream } (it might be less than { @code length } if
{ @code inputStream } ends).
*/
public static long LOAD_KEYS(final InputStream inputStream, final KEY_TYPE[][] array, final long offset, final
long length) throws IOException {
    return read(inputStream, array, offset, length);
}

/** Loads bytes from a given input stream, storing them in a given big array.
*
* <p>Note that this method is going to be significantly faster than { @link #loadBytes(DataInput,byte[][])}
* as it uses { @link InputStream}'s bulk-read methods.
*
* @param inputStream an input stream.
* @param array a big array which will be filled with data from { @code inputStream }.
* @return the number of elements actually read from { @code inputStream } (it might be less than the array length if
{ @code inputStream } ends).
*/
public static long LOAD_KEYS(final InputStream inputStream, final KEY_TYPE[][] array) throws IOException {
    return read(inputStream, array, 0, BIG_ARRAYS.length(array));
}

/** Stores a big-array fragment to a given output stream.
*
* <p>Note that this method is going to be significantly faster than { @link
#storeBytes(byte[][],long,long,DataOutput)}
* as it uses { @link OutputStream}'s bulk-read methods.
*
* @param array a big array whose elements will be written to { @code outputStream }.
* @param offset the index of the first element of { @code array } to be written.
* @param length the number of elements of { @code array } to be written.
* @param outputStream an output stream.
*/
public static void STORE_KEYS(final KEY_TYPE array[][], final long offset, final long length, final OutputStream
outputStream) throws IOException {
    write(outputStream, array, offset, length);
}

/** Stores a big array to a given output stream.
*
* <p>Note that this method is going to be significantly faster than { @link #storeBytes(byte[][],DataOutput)}
* as it uses { @link OutputStream}'s bulk-read methods.
*
* @param array a big array whose elements will be written to { @code outputStream }.
* @param outputStream an output stream.
*/
public static void STORE_KEYS(final KEY_TYPE array[][], final OutputStream outputStream) throws

```

```

IOException {
    write(outputStream, array, 0, BIG_ARRAYS.length(array));
}

#endif

/** Loads elements from a given data input, storing them in a given array fragment.
 *
 * @param dataInput a data input.
 * @param array an array which will be filled with data from { @code dataInput }.
 * @param offset the index of the first element of { @code array } to be filled.
 * @param length the number of elements of { @code array } to be filled.
 * @return the number of elements actually read from { @code dataInput } (it might be less than { @code length } if
 { @code dataInput } ends).
 */
public static int LOAD_KEYS(final DataInput dataInput, final KEY_TYPE[] array, final int offset, final int length)
throws IOException {
    PACKAGE.ARRAYS.ensureOffsetLength(array, offset, length);
    int i = 0;
    try {
        for(i = 0; i < length; i++) array[i + offset] = dataInput.READ_KEY();
    }
    catch(EOFException itsOk) {}
    return i;
}

/** Loads elements from a given data input, storing them in a given array.
 *
 * @param dataInput a data input.
 * @param array an array which will be filled with data from { @code dataInput }.
 * @return the number of elements actually read from { @code dataInput } (it might be less than the array length if
 { @code dataInput } ends).
 */
public static int LOAD_KEYS(final DataInput dataInput, final KEY_TYPE[] array) throws IOException {
    int i = 0;
    try {
        final int length = array.length;
        for(i = 0; i < length; i++) array[i] = dataInput.READ_KEY();
    }
    catch(EOFException itsOk) {}
    return i;
}

/** Loads elements from a file given by a { @link File } object, storing them in a given array fragment.
 *
 * @param file a file.
 * @param array an array which will be filled with data from the specified file.

```

```

* @param offset the index of the first element of { @code array } to be filled.
* @param length the number of elements of { @code array } to be filled.
* @return the number of elements actually read from the given file (it might be less than { @code length } if the file
is too short).
*/

```

```

public static int LOAD_KEYS(final File file, final KEY_TYPE[] array, final int offset, final int length) throws
IOException {

```

```

    PACKAGE.ARRAYS.ensureOffsetLength(array, offset, length);

```

```

    final FileInputStream fis = new FileInputStream(file);

```

```

    #if KEY_CLASS_Byte

```

```

        final int result = read(fis, array, offset, length);

```

```

        fis.close();

```

```

        return result;

```

```

    #else

```

```

        final DataInputStream dis = new DataInputStream(new FastBufferedInputStream(fis));

```

```

        int i = 0;

```

```

        try {

```

```

            for(i = 0; i < length; i++) array[i + offset] = dis.READ_KEY();

```

```

        }

```

```

        catch(EOFException itsOk) {}

```

```

        dis.close();

```

```

        return i;

```

```

    #endif

```

```

}

```

```

/** Loads elements from a file given by a pathname, storing them in a given array fragment.

```

```

*

```

```

* @param filename a filename.

```

```

* @param array an array which will be filled with data from the specified file.

```

```

* @param offset the index of the first element of { @code array } to be filled.

```

```

* @param length the number of elements of { @code array } to be filled.

```

```

* @return the number of elements actually read from the given file (it might be less than { @code length } if the file
is too short).

```

```

*/

```

```

public static int LOAD_KEYS(final CharSequence filename, final KEY_TYPE[] array, final int offset, final int
length) throws IOException {

```

```

    return LOAD_KEYS(new File(filename.toString()), array, offset, length);

```

```

}

```

```

/** Loads elements from a file given by a { @link File } object, storing them in a given array.

```

```

*

```

```

* @param file a file.

```

```

* @param array an array which will be filled with data from the specified file.

```

```

* @return the number of elements actually read from the given file (it might be less than the array length if the file
is too short).

```

```

*/
public static int LOAD_KEYS(final File file, final KEY_TYPE[] array) throws IOException {
    final FileInputStream fis = new FileInputStream(file);
    #if KEY_CLASS_Byte
        final int result = read(fis, array, 0, array.length);
        fis.close();
        return result;
    #else
        final DataInputStream dis = new DataInputStream(new FastBufferedInputStream(fis));

        int i = 0;
        try {
            final int length = array.length;
            for(i = 0; i < length; i++) array[i] = dis.READ_KEY();
        }
        catch(EOFException itsOk) {}

        dis.close();

        return i;
    #endif
}

/** Loads elements from a file given by a pathname, storing them in a given array.
 *
 * @param filename a filename.
 * @param array an array which will be filled with data from the specified file.
 * @return the number of elements actually read from the given file (it might be less than the array length if the file
is too short).
 */
public static int LOAD_KEYS(final CharSequence filename, final KEY_TYPE[] array) throws IOException {
    return LOAD_KEYS(new File(filename.toString()), array);
}

/** Loads elements from a file given by a {@link File} object, storing them in a new array.
 *
 * <p>Note that the length of the returned array will be computed
 * dividing the specified file size by the number of bytes used to
 * represent each element.
 *
 * @param file a file.
 * @return an array filled with the content of the specified file.
 */
public static KEY_TYPE[] LOAD_KEYS(final File file) throws IOException {
    final FileInputStream fis = new FileInputStream(file);

    #if KEY_CLASS_Boolean
        final long length = fis.getChannel().size();

```

```

#else
    final long length = fis.getChannel().size() / (KEY_CLASS.SIZE / 8);
#endif

    if (length > Integer.MAX_VALUE) {
        fis.close();
        throw new IllegalArgumentException("File too long: " + fis.getChannel().size()+ " bytes (" + length + " elements)");
    }

    final KEY_TYPE[] array = new KEY_TYPE[(int)length];

    #if KEY_CLASS_Byte
    if (read(fis, array, 0, (int)length) < length) throw new EOFException();
    fis.close();
    #else
    final DataInputStream dis = new DataInputStream(new FastBufferedInputStream(fis));
    for(int i = 0; i < length; i++) array[i] = dis.READ_KEY();
    dis.close();
    #endif
    return array;
}

/** Loads elements from a file given by a filename, storing them in a new array.
 *
 * <p>Note that the length of the returned array will be computed
 * dividing the specified file size by the number of bytes used to
 * represent each element.
 *
 * @param filename a filename.
 * @return an array filled with the content of the specified file.
 */
public static KEY_TYPE[] LOAD_KEYS(final CharSequence filename) throws IOException {
    return LOAD_KEYS(new File(filename.toString()));
}

/** Stores an array fragment to a given data output.
 *
 * @param array an array whose elements will be written to { @code dataOutput }.
 * @param offset the index of the first element of { @code array } to be written.
 * @param length the number of elements of { @code array } to be written.
 * @param dataOutput a data output.
 */
public static void STORE_KEYS(final KEY_TYPE array[], final int offset, final int length, final DataOutput
dataOutput) throws IOException {
    PACKAGE.ARRAYS.ensureOffsetLength(array, offset, length);
    #if KEY_CLASS_Byte
    write(dataOutput, array, offset, length);
    #else

```

```

    for(int i = 0; i < length; i++) dataOutput.WRITE_KEY(array[offset + i]);
#endif
}

/** Stores an array to a given data output.
 *
 * @param array an array whose elements will be written to { @code dataOutput}.
 * @param dataOutput a data output.
 */
public static void STORE_KEYS(final KEY_TYPE array[], final DataOutput dataOutput) throws IOException {
    #if KEY_CLASS_Byte
        write(dataOutput, array, 0, array.length);
    #else
        final int length = array.length;
        for(int i = 0; i < length; i++) dataOutput.WRITE_KEY(array[i]);
    #endif
}

/** Stores an array fragment to a file given by a { @link File} object.
 *
 * @param array an array whose elements will be written to { @code filename}.
 * @param offset the index of the first element of { @code array} to be written.
 * @param length the number of elements of { @code array} to be written.
 * @param file a file.
 */
public static void STORE_KEYS(final KEY_TYPE array[], final int offset, final int length, final File file) throws
IOException {
    PACKAGE.ARRAYS.ensureOffsetLength(array, offset, length);
    #if KEY_CLASS_Byte
        final OutputStream os = new FastBufferedOutputStream(new FileOutputStream(file));
        write(os, array, offset, length);
        os.close();
    #else
        final DataOutputStream dos = new DataOutputStream(new FastBufferedOutputStream(new
FileOutputStream(file)));
        for(int i = 0; i < length; i++) dos.WRITE_KEY(array[offset + i]);
        dos.close();
    #endif
}

/** Stores an array fragment to a file given by a pathname.
 *
 * @param array an array whose elements will be written to { @code filename}.
 * @param offset the index of the first element of { @code array} to be written.
 * @param length the number of elements of { @code array} to be written.
 * @param filename a filename.
 */
public static void STORE_KEYS(final KEY_TYPE array[], final int offset, final int length, final CharSequence

```

```

filename) throws IOException {
    STORE_KEYS(array, offset, length, new File(filename.toString()));
}

/** Stores an array to a file given by a {@link File} object.
 *
 * @param array an array whose elements will be written to {@code filename}.
 * @param file a file.
 */
public static void STORE_KEYS(final KEY_TYPE array[], final File file) throws IOException {
    #if KEY_CLASS_Byte
    final OutputStream os = new FastBufferedOutputStream(new FileOutputStream(file));
    write(os, array, 0, array.length);
    os.close();
    #else
    final int length = array.length;
    final DataOutputStream dos = new DataOutputStream(new FastBufferedOutputStream(new
    FileOutputStream(file)));
    for(int i = 0; i < length; i++) dos.WRITE_KEY(array[i]);
    dos.close();
    #endif
}

/** Stores an array to a file given by a pathname.
 *
 * @param array an array whose elements will be written to {@code filename}.
 * @param filename a filename.
 */
public static void STORE_KEYS(final KEY_TYPE array[], final CharSequence filename) throws IOException {
    STORE_KEYS(array, new File(filename.toString()));
}

/** Loads elements from a given data input, storing them in a given big-array fragment.
 *
 * @param dataInput a data input.
 * @param array a big array which will be filled with data from {@code dataInput}.
 * @param offset the index of the first element of {@code bigArray} to be filled.
 * @param length the number of elements of {@code bigArray} to be filled.
 * @return the number of elements actually read from {@code dataInput} (it might be less than {@code length} if
 * {@code dataInput} ends).
 */
public static long LOAD_KEYS(final DataInput dataInput, final KEY_TYPE[][] array, final long offset, final long
length) throws IOException {
    PACKAGE.BIG_ARRAYS.ensureOffsetLength(array, offset, length);
}

```

```

long c = 0;
try {
    for(int i = segment(offset); i < segment(offset + length + SEGMENT_MASK); i++) {
        final KEY_TYPE[] t = array[i];
        final int l = (int)Math.min(t.length, offset + length - start(i));
        for(int d = (int)Math.max(0, offset - start(i)); d < l; d++) {
            t[d] = dataInput.READ_KEY();
            c++;
        }
    }
}
catch(EOFException itsOk) {}
return c;
}

/** Loads elements from a given data input, storing them in a given big array.
 *
 * @param dataInput a data input.
 * @param array a big array which will be filled with data from {@code dataInput}.
 * @return the number of elements actually read from {@code dataInput} (it might be less than the array length if
 * {@code dataInput} ends).
 */
public static long LOAD_KEYS(final DataInput dataInput, final KEY_TYPE[][] array) throws IOException {
    long c = 0;
    try {
        for(int i = 0; i < array.length; i++) {
            final KEY_TYPE[] t = array[i];
            final int l = t.length;
            for(int d = 0; d < l; d++) {
                t[d] = dataInput.READ_KEY();
                c++;
            }
        }
    }
    catch(EOFException itsOk) {}
    return c;
}

/** Loads elements from a file given by a {@link File} object, storing them in a given big-array fragment.
 *
 * @param file a file.
 * @param array a big array which will be filled with data from the specified file.
 * @param offset the index of the first element of {@code array} to be filled.
 * @param length the number of elements of {@code array} to be filled.
 * @return the number of elements actually read from the given file (it might be less than {@code length} if the file
 * is too short).
 */
public static long LOAD_KEYS(final File file, final KEY_TYPE[][] array, final long offset, final long length)

```

```

throws IOException {
    PACKAGE.BIG_ARRAYS.ensureOffsetLength(array, offset, length);

    final FileInputStream fis = new FileInputStream(file);
    #if KEY_CLASS_Byte
    final long result = read(fis, array, offset, length);
    fis.close();
    return result;
    #else
    final DataInputStream dis = new DataInputStream(new FastBufferedInputStream(fis));

    long c = 0;
    try {
        for(int i = segment(offset); i < segment(offset + length + SEGMENT_MASK); i++) {
            final KEY_TYPE[] t = array[i];
            final int l = (int)Math.min(t.length, offset + length - start(i));
            for(int d = (int)Math.max(0, offset - start(i)); d < l; d++) {
                t[d] = dis.READ_KEY();
                c++;
            }
        }
    } catch(EOFException itsOk) {}
    dis.close();
    return c;
    #endif
}

/** Loads elements from a file given by a pathname, storing them in a given big-array fragment.
 *
 * @param filename a filename.
 * @param array an array which will be filled with data from the specified file.
 * @param offset the index of the first element of { @code array } to be filled.
 * @param length the number of elements of { @code array } to be filled.
 * @return the number of elements actually read from the given file (it might be less than { @code length } if the file
 is too short).
 */
public static long LOAD_KEYS(final CharSequence filename, final KEY_TYPE[][] array, final long offset, final
long length) throws IOException {
    return LOAD_KEYS(new File(filename.toString()), array, offset, length);
}

/** Loads elements from a file given by a { @link File } object, storing them in a given big array.
 *
 * @param file a file.
 * @param array a big array which will be filled with data from the specified file.
 * @return the number of elements actually read from the given file (it might be less than the array length if the file
 is too short).

```

```

*/
public static long LOAD_KEYS(final File file, final KEY_TYPE[][] array) throws IOException {
    final FileInputStream fis = new FileInputStream(file);
    #if KEY_CLASS_Byte
    final long result = read(fis, array, 0, BIG_ARRAYS.length(array));
    fis.close();
    return result;
    #else
    final DataInputStream dis = new DataInputStream(new FastBufferedInputStream(fis));

    long c = 0;
    try {
        for(int i = 0; i < array.length; i++) {
            final KEY_TYPE[] t = array[i];
            final int l = t.length;
            for(int d = 0; d < l; d++) {
                t[d] = dis.READ_KEY();
                c++;
            }
        }
    } catch(EOFException itsOk) {}

    dis.close();

    return c;
    #endif
}

/** Loads elements from a file given by a pathname, storing them in a given big array.
 *
 * @param filename a filename.
 * @param array a big array which will be filled with data from the specified file.
 * @return the number of elements actually read from the given file (it might be less than the array length if the file
 is too short).
 */
public static long LOAD_KEYS(final CharSequence filename, final KEY_TYPE[][] array) throws IOException {
    return LOAD_KEYS(new File(filename.toString()), array);
}

/** Loads elements from a file given by a {@link File} object, storing them in a new big array.
 *
 * <p>Note that the length of the returned big array will be computed
 * dividing the specified file size by the number of bytes used to
 * represent each element.
 *
 * @param file a file.
 * @return a big array filled with the content of the specified file.

```

```

*/
public static KEY_TYPE[][] LOAD_KEYS_BIG(final File file) throws IOException {
    final FileInputStream fis = new FileInputStream(file);

    #if KEY_CLASS_Boolean
        final long length = fis.getChannel().size();
    #else
        final long length = fis.getChannel().size() / (KEY_CLASS.SIZE / 8);
    #endif

    final KEY_TYPE[][] array = BIG_ARRAYS.newBigArray(length);

    #if KEY_CLASS_Byte
        if (read(fis, array, 0, length) < length) throw new EOFException();
        fis.close();
    #else
        final DataInputStream dis = new DataInputStream(new FastBufferedInputStream(fis));

        for(int i = 0; i < array.length; i++) {
            final KEY_TYPE[] t = array[i];
            final int l = t.length;
            for(int d = 0; d < l; d++) t[d] = dis.READ_KEY();
        }

        dis.close();
    #endif
    return array;
}

/** Loads elements from a file given by a filename, storing them in a new big array.
 *
 * <p>Note that the length of the returned big array will be computed
 * dividing the specified file size by the number of bytes used to
 * represent each element.
 *
 * @param filename a filename.
 * @return a big array filled with the content of the specified file.
 */
public static KEY_TYPE[][] LOAD_KEYS_BIG(final CharSequence filename) throws IOException {
    return LOAD_KEYS_BIG(new File(filename.toString()));
}

/** Stores an array fragment to a given data output.
 *
 * @param array an array whose elements will be written to { @code dataOutput }.
 * @param offset the index of the first element of { @code array } to be written.
 * @param length the number of elements of { @code array } to be written.
 * @param dataOutput a data output.

```

```

*/
public static void STORE_KEYS(final KEY_TYPE array[][], final long offset, final long length, final DataOutput
dataOutput) throws IOException {
    PACKAGE.BIG_ARRAYS.ensureOffsetLength(array, offset, length);
    #if KEY_CLASS_Byte
        write(dataOutput, array, offset, length);
    #else
        for(int i = segment(offset); i < segment(offset + length + SEGMENT_MASK); i++) {
            final KEY_TYPE[] t = array[i];
            final int l = (int)Math.min(t.length, offset + length - start(i));
            for(int d = (int)Math.max(0, offset - start(i)); d < l; d++) dataOutput.WRITE_KEY(t[d]);
        }
    #endif
}

/** Stores a big array to a given data output.
 *
 * @param array a big array whose elements will be written to { @code dataOutput }.
 * @param dataOutput a data output.
 */
public static void STORE_KEYS(final KEY_TYPE array[][], final DataOutput dataOutput) throws IOException {
    #if KEY_CLASS_Byte
        write(dataOutput, array, 0, BIG_ARRAYS.length(array));
    #else
        for(int i = 0; i < array.length; i++) {
            final KEY_TYPE[] t = array[i];
            final int l = t.length;
            for(int d = 0; d < l; d++) dataOutput.WRITE_KEY(t[d]);
        }
    #endif
}

/** Stores a big-array fragment to a file given by a { @link File } object.
 *
 * @param array a big array whose elements will be written to { @code filename }.
 * @param offset the index of the first element of { @code array } to be written.
 * @param length the number of elements of { @code array } to be written.
 * @param file a file.
 */
public static void STORE_KEYS(final KEY_TYPE array[][], final long offset, final long length, final File file)
throws IOException {
    PACKAGE.BIG_ARRAYS.ensureOffsetLength(array, offset, length);
    #if KEY_CLASS_Byte
        final OutputStream os = new FastBufferedOutputStream(new FileOutputStream(file));
        write(os, array, offset, length);
        os.close();
    #else
        final DataOutputStream dos = new DataOutputStream(new FastBufferedOutputStream(new

```

```

FileOutputStream(file));
for(int i = segment(offset); i < segment(offset + length + SEGMENT_MASK); i++) {
    final KEY_TYPE[] t = array[i];
    final int l = (int)Math.min(t.length, offset + length - start(i));
    for(int d = (int)Math.max(0, offset - start(i)); d < l; d++) dos.WRITE_KEY(t[d]);
}
dos.close();

#endif
}

/** Stores a big-array fragment to a file given by a pathname.
 *
 * @param array a big array whose elements will be written to { @code filename }.
 * @param offset the index of the first element of { @code array } to be written.
 * @param length the number of elements of { @code array } to be written.
 * @param filename a filename.
 */
public static void STORE_KEYS(final KEY_TYPE array[][], final long offset, final long length, final
CharSequence filename) throws IOException {
    STORE_KEYS(array, offset, length, new File(filename.toString()));
}

/** Stores an array to a file given by a { @link File } object.
 *
 * @param array an array whose elements will be written to { @code filename }.
 * @param file a file.
 */
public static void STORE_KEYS(final KEY_TYPE array[][], final File file) throws IOException {
    #if KEY_CLASS_Byte
    final OutputStream os = new FastBufferedOutputStream(new FileOutputStream(file));
    write(os, array, 0, BIG_ARRAYS.length(array));
    os.close();
    #else
    final DataOutputStream dos = new DataOutputStream(new FastBufferedOutputStream(new
FileOutputStream(file)));
    for(int i = 0; i < array.length; i++) {
        final KEY_TYPE[] t = array[i];
        final int l = t.length;
        for(int d = 0; d < l; d++) dos.WRITE_KEY(t[d]);
    }
    dos.close();
    #endif
}

/** Stores a big array to a file given by a pathname.
 *
 * @param array a big array whose elements will be written to { @code filename }.

```

```

* @param filename a filename.
*/
public static void STORE_KEYS(final KEY_TYPE array[], final CharSequence filename) throws IOException {
    STORE_KEYS(array, new File(filename.toString()));
}

/** Stores the element returned by an iterator to a given data output.
*
* @param i an iterator whose output will be written to { @code dataOutput}.
* @param dataOutput a filename.
*/
public static void STORE_KEYS(final KEY_ITERATOR i, final DataOutput dataOutput) throws IOException {
    while(i.hasNext()) dataOutput.WRITE_KEY(i.NEXT_KEY());
}

/** Stores the element returned by an iterator to a file given by a { @link File} object.
*
* @param i an iterator whose output will be written to { @code filename}.
* @param file a file.
*/
public static void STORE_KEYS(final KEY_ITERATOR i, final File file) throws IOException {
    final DataOutputStream dos = new DataOutputStream(new FastBufferedOutputStream(new
    FileOutputStream(file)));
    while(i.hasNext()) dos.WRITE_KEY(i.NEXT_KEY());
    dos.close();
}

/** Stores the element returned by an iterator to a file given by a pathname.
*
* @param i an iterator whose output will be written to { @code filename}.
* @param filename a filename.
*/
public static void STORE_KEYS(final KEY_ITERATOR i, final CharSequence filename) throws IOException {
    STORE_KEYS(i, new File(filename.toString()));
}

/** A wrapper that exhibits the content of a data input stream as a type-specific iterator. */

private static final class KEY_DATA_INPUT_WRAPPER implements KEY_ITERATOR {
    private final DataInput dataInput;
    private boolean toAdvance = true;
    private boolean endOfProcess = false;
    private KEY_TYPE next;

    public KEY_DATA_INPUT_WRAPPER(final DataInput dataInput) {
        this.dataInput = dataInput;
    }
}

```

```

@Override
public boolean hasNext() {
    if (!toAdvance) return !endOfProcess;

    toAdvance = false;

    try { next = dataInput.READ_KEY(); }
    catch(EOFException eof) { endOfProcess = true; }
    catch(IOException rethrow) { throw new RuntimeException(rethrow); }

    return !endOfProcess;
}

```

```

@Override
public KEY_TYPE NEXT_KEY() {
    if (!hasNext()) throw new NoSuchElementException();
    toAdvance = true;
    return next;
}
}

```

/** Wraps the given data input stream into an iterator.

*

* @param dataInput a data input.

*/

```

public static KEY_ITERATOR AS_KEY_ITERATOR(final DataInput dataInput) {
    return new KEY_DATA_INPUT_WRAPPER(dataInput);
}

```

/** Wraps a file given by a {@link File} object into an iterator.

*

* @param file a file.

*/

```

public static KEY_ITERATOR AS_KEY_ITERATOR(final File file) throws IOException {
    return new KEY_DATA_INPUT_WRAPPER(new DataInputStream(new FastBufferedInputStream(new
    FileInputStream(file))));
}

```

/** Wraps a file given by a pathname into an iterator.

*

* @param filename a filename.

*/

```

public static KEY_ITERATOR AS_KEY_ITERATOR(final CharSequence filename) throws IOException {
    return AS_KEY_ITERATOR(new File(filename.toString()));
}

```

/** Wraps a file given by a {@link File} object into an iterable object.

*

```

* @param file a file.
*/
public static KEY_ITERABLE AS_KEY_ITERABLE(final File file) {
    return () -> {
        try { return AS_KEY_ITERATOR(file); }
        catch(IOException e) { throw new RuntimeException(e); }
    };
}

/** Wraps a file given by a pathname into an iterable object.
*
* @param filename a filename.
*/
public static KEY_ITERABLE AS_KEY_ITERABLE(final CharSequence filename) {
    return () -> {
        try { return AS_KEY_ITERATOR(filename); }
        catch(IOException e) { throw new RuntimeException(e); }
    };
}

```

Found in path(s):

```

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-
a775574/drv/BinIOFragment.drv

```

No license file was found, but licenses were detected in source scan.

```

/*
* Copyright (C) 2010-2017 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

```

```

package PACKAGE;

```

```

import java.util.List;
import it.unimi.dsi.fastutil.BigList;
import it.unimi.dsi.fastutil.Size64;

```

```
#if ! KEY_CLASS_Reference
```

```
/** A type-specific {@link BigList}; provides some additional methods that use polymorphism to avoid (un)boxing.  
*  
* <p>Additionally, this interface strengthens {@link #iterator()}, {@link #listIterator()},  
* {@link #listIterator(long)} and {@link #subList(long,long)}.  
*  
* <p>Besides polymorphic methods, this interfaces specifies methods to copy into an array or remove contiguous  
* sublists. Although the abstract implementation of this interface provides simple, one-by-one implementations  
* of these methods, it is expected that concrete implementation override them with optimized versions.  
*  
* @see List  
*/
```

```
public interface BIG_LIST KEY_GENERIC extends BigList<KEY_GENERIC_CLASS>, COLLECTION  
KEY_GENERIC, Size64, Comparable<BigList<? extends KEY_GENERIC_CLASS>> {  
#else
```

```
/** A type-specific {@link BigList}; provides some additional methods that use polymorphism to avoid (un)boxing.  
*  
* <p>Additionally, this interface strengthens {@link #listIterator()},  
* {@link #listIterator(long)} and {@link #subList(long,long)}.  
*  
* <p>Besides polymorphic methods, this interfaces specifies methods to copy into an array or remove contiguous  
* sublists. Although the abstract implementation of this interface provides simple, one-by-one implementations  
* of these methods, it is expected that concrete implementation override them with optimized versions.  
*  
* @see List  
*/
```

```
public interface BIG_LIST KEY_GENERIC extends BigList<KEY_GENERIC_CLASS>, COLLECTION  
KEY_GENERIC, Size64 {  
#endif
```

```
/** Returns a type-specific iterator on the elements of this list.  
*  
* <p>Note that this specification strengthens the one given in {@link java.util.Collection#iterator()}.  
* @see java.util.Collection#iterator()  
*/
```

```
@Override  
KEY_BIG_LIST_ITERATOR KEY_GENERIC iterator();
```

```
/** Returns a type-specific big-list iterator on this type-specific big list.  
*  
* <p>Note that this specification strengthens the one given in {@link BigList#listIterator()}.  
* @see BigList#listIterator()  
*/
```

```

@Override
KEY_BIG_LIST_ITERATOR KEY_GENERIC listIterator();

/** Returns a type-specific list iterator on this type-specific big list starting at a given index.
 *
 * <p>Note that this specification strengthens the one given in { @link BigList#listIterator(long)}.
 * @see BigList#listIterator(long)
 */
@Override
KEY_BIG_LIST_ITERATOR KEY_GENERIC listIterator(long index);

/** Returns a type-specific view of the portion of this type-specific big list from the index { @code from}, inclusive,
to the index { @code to}, exclusive.
 *
 * <p>Note that this specification strengthens the one given in { @link BigList#subList(long,long)}.
 *
 * @see BigList#subList(long,long)
 */
@Override
BIG_LIST KEY_GENERIC subList(long from, long to);

/** Copies (hopefully quickly) elements of this type-specific big list into the given big array.
 *
 * @param from the start index (inclusive).
 * @param a the destination big array.
 * @param offset the offset into the destination big array where to store the first element copied.
 * @param length the number of elements to be copied.
 */
void getElements(long from, KEY_TYPE a[], long offset, long length);

/** Removes (hopefully quickly) elements of this type-specific big list.
 *
 * @param from the start index (inclusive).
 * @param to the end index (exclusive).
 */
void removeElements(long from, long to);

/** Add (hopefully quickly) elements to this type-specific big list.
 *
 * @param index the index at which to add elements.
 * @param a the big array containing the elements.
 */
void addElements(long index, KEY_GENERIC_TYPE a[][]);

/** Add (hopefully quickly) elements to this type-specific big list.
 *
 * @param index the index at which to add elements.
 * @param a the big array containing the elements.
 */

```

```

* @param offset the offset of the first element to add.
* @param length the number of elements to add.
*/
void addElements(long index, KEY_GENERIC_TYPE a[], long offset, long length);

#if KEYS_PRIMITIVE

/** Inserts the specified element at the specified position in this type-specific big list (optional operation).
* @see BigList#add(long,Object)
*/
void add(long index, KEY_TYPE key);

/** Inserts all of the elements in the specified type-specific collection into this type-specific big list at the specified
position (optional operation).
* @see List#addAll(int,java.util.Collection)
*/
boolean addAll(long index, COLLECTION c);

/** Inserts all of the elements in the specified type-specific big list into this type-specific big list at the specified
position (optional operation).
* @see List#addAll(int,java.util.Collection)
*/
boolean addAll(long index, BIG_LIST c);

/** Appends all of the elements in the specified type-specific big list to the end of this type-specific big list (optional
operation).
* @see List#addAll(int,java.util.Collection)
*/
boolean addAll(BIG_LIST c);

/** Returns the element at the specified position.
* @see BigList#get(long)
*/
KEY_TYPE GET_KEY(long index);

/** Removes the element at the specified position.
* @see BigList#remove(long) */
KEY_TYPE REMOVE_KEY(long index);

/** Replaces the element at the specified position in this big list with the specified element (optional operation).
* @see BigList#set(long,Object)
*/
KEY_TYPE set(long index, KEY_TYPE k);

/** Returns the index of the first occurrence of the specified element in this type-specific big list, or -1 if this big list
does not contain the element.
* @see BigList#indexOf(Object)
*/

```

```

long indexOf(KEY_TYPE k);

/** Returns the index of the last occurrence of the specified element in this type-specific big list, or -1 if this big list
does not contain the element.
 * @see BigList#lastIndexOf(Object)
 */
long lastIndexOf(KEY_TYPE k);

/** { @inheritDoc }
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
void add(long index, KEY_CLASS key);

/** { @inheritDoc }
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
KEY_GENERIC_CLASS get(long index);

/** { @inheritDoc }
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
long indexOf(Object o);

/** { @inheritDoc }
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
long lastIndexOf(Object o);

/** { @inheritDoc }
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
KEY_GENERIC_CLASS remove(long index);

/** { @inheritDoc }
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
KEY_GENERIC_CLASS set(long index, KEY_GENERIC_CLASS k);
#endif

}

```

Found in path(s):

```
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-
a775574/drv/BigList.drv
```

No license file was found, but licenses were detected in source scan.

```
/*
```

```
* Copyright (C) 2010-2017 Sebastiano Vigna
```

```
*
```

```
* Licensed under the Apache License, Version 2.0 (the "License");
```

```
* you may not use this file except in compliance with the License.
```

```
* You may obtain a copy of the License at
```

```
*
```

```
* http://www.apache.org/licenses/LICENSE-2.0
```

```
*
```

```
* Unless required by applicable law or agreed to in writing, software
```

```
* distributed under the License is distributed on an "AS IS" BASIS,
```

```
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
```

```
* See the License for the specific language governing permissions and
```

```
* limitations under the License.
```

```
*/
```

```
package PACKAGE;
```

```
import it.unimi.dsi.fastutil.BigListIterator;
```

```
import it.unimi.dsi.fastutil.SafeMath;
```

```
/** A type-specific { @link BigListIterator}.
```

```
*
```

```
* @see BigListIterator
```

```
*/
```

```
public interface KEY_BIG_LIST_ITERATOR KEY_GENERIC extends KEY_BIDI_ITERATOR
KEY_GENERIC, BigListIterator<KEY_GENERIC_CLASS> {
```

```
/**
```

```
* Replaces the last element returned by { @link BigListIterator#next() next()} or
```

```
* { @link BigListIterator#previous() previous()} with the specified element (optional operation).
```

```
* @see java.util.ListIterator#set(Object)
```

```
*/
```

```
#if KEYS_REFERENCE
```

```
  @Override
```

```
#endif
```

```
  default void set(@SuppressWarnings("unused") final KEY_GENERIC_TYPE k) { throw new
  UnsupportedOperationException(); }
```

```
/**
```

```
* Inserts the specified element into the list (optional operation).
```

```

* @see java.util.ListIterator#add(Object)
*/

#if KEYS_REFERENCE
@Override
#endif
default void add(@SuppressWarnings("unused") final KEY_GENERIC_TYPE k) { throw new
UnsupportedOperationException(); }

#if KEYS_PRIMITIVE

/** Replaces the last element returned by {@link #next()} or {@link #previous()} with the specified element
(optional operation).
* @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default void set(final KEY_GENERIC_CLASS k) { set(k.KEY_VALUE()); }

/** Inserts the specified element into the list (optional operation).
* @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default void add(final KEY_GENERIC_CLASS k) { add(k.KEY_VALUE()); }

#endif

/** Skips the given number of elements.
*
* <p>The effect of this call is exactly the same as that of
* calling {@link BigListIterator#next() next()} for {@code n} times (possibly stopping
* if {@link #hasNext()} becomes false).
*
* @param n the number of elements to skip.
* @return the number of elements actually skipped.
* @see BigListIterator#next()
*/
default long skip(final long n) {
    long i = n;
    while(i-- != 0 && hasNext()) NEXT_KEY();
    return n - i - 1;
}

/** Moves back for the given number of elements.
*
* <p>The effect of this call is exactly the same as that of
* calling {@link BigListIterator#previous() previous()} for {@code n} times (possibly stopping
* if {@link #hasPrevious()} becomes false).

```

```

*
* @param n the number of elements to skip back.
* @return the number of elements actually skipped.
* @see BigListIterator#previous()
*/
default long back(final long n) {
    long i = n;
    while(i-- != 0 && hasPrevious()) PREV_KEY();
    return n - i - 1;
}

/** {@inheritDoc}
*/
@Override
default int skip(int n) {
    return SafeMath.safeLongToInt(skip((long) n));
}
}

```

Found in path(s):

```
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-
a775574/drv/BigListIterator.drv
```

No license file was found, but licenses were detected in source scan.

```

/*
* Copyright (C) 2002-2017 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

```

```
package PACKAGE;
```

```

/** An abstract class facilitating the creation of type-specific { @linkplain it.unimi.dsi.fastutil.BigListIterator big-
list iterators }.
*
* @deprecated As of fastutil 8 this class is no longer necessary, as its previous abstract
* methods are now default methods of the type-specific interface.

```

*/

@Deprecated

```
public abstract class KEY_ABSTRACT_BIG_LIST_ITERATOR KEY_GENERIC extends
KEY_ABSTRACT_BIDI_ITERATOR KEY_GENERIC implements KEY_BIG_LIST_ITERATOR
KEY_GENERIC {
    protected KEY_ABSTRACT_BIG_LIST_ITERATOR() {}
}
```

Found in path(s):

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/drv/AbstractBigListIterator.drv

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2002-2017 Sebastiano Vigna

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

package PACKAGE;

#if KEYS_REFERENCE

import it.unimi.dsi.fastutil.Stack;

#endif

import java.util.List;

import java.util.Iterator;

import java.util.ListIterator;

import java.util.Collection;

import java.util.NoSuchElementException;

/** An abstract class providing basic methods for lists implementing a type-specific list interface.

*

* <p>As an additional bonus, this class implements on top of the list operations a type-specific stack.

*/

```

public abstract class ABSTRACT_LIST KEY_GENERIC extends ABSTRACT_COLLECTION KEY_GENERIC
implements LIST KEY_GENERIC, STACK KEY_GENERIC {

protected ABSTRACT_LIST() {}

/** Ensures that the given index is nonnegative and not greater than the list size.
 *
 * @param index an index.
 * @throws IndexOutOfBoundsException if the given index is negative or greater than the list size.
 */
protected void ensureIndex(final int index) {
if (index < 0) throw new IndexOutOfBoundsException("Index (" + index + ") is negative");
if (index > size()) throw new IndexOutOfBoundsException("Index (" + index + ") is greater than list size (" +
(size()) + ")");
}

/** Ensures that the given index is nonnegative and smaller than the list size.
 *
 * @param index an index.
 * @throws IndexOutOfBoundsException if the given index is negative or not smaller than the list size.
 */
protected void ensureRestrictedIndex(final int index) {
if (index < 0) throw new IndexOutOfBoundsException("Index (" + index + ") is negative");
if (index >= size()) throw new IndexOutOfBoundsException("Index (" + index + ") is greater than or equal to list
size (" + (size()) + ")");
}

/** {@inheritDoc}
 *
 * <p>This implementation always throws an {@link UnsupportedOperationException}.
 */
@Override
public void add(final int index, final KEY_GENERIC_TYPE k) {
throw new UnsupportedOperationException();
}

/** {@inheritDoc}
 *
 * <p>This implementation delegates to the type-specific version of {@link List#add(int, Object)}.
 */
@Override
public boolean add(final KEY_GENERIC_TYPE k) {
add(size(), k);
return true;
}

/** {@inheritDoc}
 *

```

```

* <p>This implementation always throws an {@link UnsupportedOperationException}.
*/
@Override
public KEY_GENERIC_TYPE REMOVE_KEY(final int i) {
    throw new UnsupportedOperationException();
}

/** {@inheritDoc}
 *
 * <p>This implementation always throws an {@link UnsupportedOperationException}.
 */
@Override
public KEY_GENERIC_TYPE set(final int index, final KEY_GENERIC_TYPE k) {
    throw new UnsupportedOperationException();
}

/** Adds all of the elements in the specified collection to this list (optional operation). */
@Override
public boolean addAll(int index, final Collection<? extends KEY_GENERIC_CLASS> c) {
    ensureIndex(index);
    final Iterator<? extends KEY_GENERIC_CLASS> i = c.iterator();
    final boolean retVal = i.hasNext();
    while(i.hasNext()) add(index++, KEY_CLASS2TYPE(i.next()));
    return retVal;
}

/** {@inheritDoc}
 *
 * <p>This implementation delegates to the type-specific version of {@link List#addAll(int, Collection)}.
 */
@Override
public boolean addAll(final Collection<? extends KEY_GENERIC_CLASS> c) {
    return addAll(size(), c);
}

/** {@inheritDoc}
 *
 * <p>This implementation delegates to {@link #listIterator()}.
 */
@Override
public KEY_LIST_ITERATOR KEY_GENERIC iterator() {
    return listIterator();
}

/** {@inheritDoc}
 *
 * <p>This implementation delegates to {@link #listIterator(int) listIterator(0)}.
 */

```

```

@Override
public KEY_LIST_ITERATOR KEY_GENERIC listIterator() {
    return listIterator(0);
}

/** {@inheritDoc}
 * <p>This implementation is based on the random-access methods. */
@Override
public KEY_LIST_ITERATOR KEY_GENERIC listIterator(final int index) {
    ensureIndex(index);

    return new KEY_LIST_ITERATOR KEY_GENERIC() {
        int pos = index, last = -1;

        @Override
        public boolean hasNext() { return pos < ABSTRACT_LIST.this.size(); }
        @Override
        public boolean hasPrevious() { return pos > 0; }
        @Override
        public KEY_GENERIC_TYPE NEXT_KEY() { if (! hasNext()) throw new NoSuchElementException(); return
ABSTRACT_LIST.this.GET_KEY(last = pos++); }
        @Override
        public KEY_GENERIC_TYPE PREV_KEY() { if (! hasPrevious()) throw new NoSuchElementException(); return
ABSTRACT_LIST.this.GET_KEY(last = --pos); }
        @Override
        public int nextIndex() { return pos; }
        @Override
        public int previousIndex() { return pos - 1; }
        @Override
        public void add(final KEY_GENERIC_TYPE k) {
            ABSTRACT_LIST.this.add(pos++, k);
            last = -1;
        }
        @Override
        public void set(final KEY_GENERIC_TYPE k) {
            if (last == -1) throw new IllegalStateException();
            ABSTRACT_LIST.this.set(last, k);
        }
        @Override
        public void remove() {
            if (last == -1) throw new IllegalStateException();
            ABSTRACT_LIST.this.REMOVE_KEY(last);
            /* If the last operation was a next(), we are removing an element *before* us, and we must decrease pos
            correspondingly. */
            if (last < pos) pos--;
            last = -1;
        }
    };
}

```

```

}

/** Returns true if this list contains the specified element.
 * <p>This implementation delegates to {@code indexOf()}.
 * @see List#contains(Object)
 */

@Override
public boolean contains(final KEY_TYPE k) {
    return indexOf(k) >= 0;
}

@Override
public int indexOf(final KEY_TYPE k) {
    final KEY_LIST_ITERATOR KEY_GENERIC i = listIterator();
    KEY_GENERIC_TYPE e;
    while(i.hasNext()) {
        e = i.NEXT_KEY();
        if (KEY_EQUALS(k, e)) return i.previousIndex();
    }
    return -1;
}

@Override
public int lastIndexOf(final KEY_TYPE k) {
    KEY_LIST_ITERATOR KEY_GENERIC i = listIterator(size());
    KEY_GENERIC_TYPE e;
    while(i.hasPrevious()) {
        e = i.PREV_KEY();
        if (KEY_EQUALS(k, e)) return i.nextIndex();
    }
    return -1;
}

@Override
public void size(final int size) {
    int i = size();
    if (size > i) while(i++ < size) add(KEY_NULL);
    else while(i-- != size) REMOVE_KEY(i);
}

@Override
public LIST KEY_GENERIC subList(final int from, final int to) {
    ensureIndex(from);
    ensureIndex(to);
    if (from > to) throw new IndexOutOfBoundsException("Start index (" + from + ") is greater than end index (" + to +
    ")");
}

```

```

return new SUBLIST KEY_GENERIC_DIAMOND(this, from, to);
}

/** {@inheritDoc}
 *
 * <p>This is a trivial iterator-based implementation. It is expected that
 * implementations will override this method with a more optimized version.
 */
@Override
public void removeElements(final int from, final int to) {
    ensureIndex(to);
    KEY_LIST_ITERATOR KEY_GENERIC i = listIterator(from);
    int n = to - from;
    if (n < 0) throw new IllegalArgumentException("Start index (" + from + ") is greater than end index (" + to + ")");
    while(n-- != 0) {
        i.NEXT_KEY();
        i.remove();
    }
}

/** {@inheritDoc}
 *
 * <p>This is a trivial iterator-based implementation. It is expected that
 * implementations will override this method with a more optimized version.
 */
@Override
public void addElements(int index, final KEY_GENERIC_TYPE a[], int offset, int length) {
    ensureIndex(index);
    if (offset < 0) throw new ArrayIndexOutOfBoundsException("Offset (" + offset + ") is negative");
    if (offset + length > a.length) throw new ArrayIndexOutOfBoundsException("End index (" + (offset + length) + ") is greater than array length (" + a.length + ")");
    while(length-- != 0) add(index++, a[offset++]);
}

/** {@inheritDoc}
 *
 * <p>This implementation delegates to the analogous method for array fragments.
 */
@Override
public void addElements(final int index, final KEY_GENERIC_TYPE a[]) {
    addElements(index, a, 0, a.length);
}

/** {@inheritDoc}
 *
 * <p>This is a trivial iterator-based implementation. It is expected that
 * implementations will override this method with a more optimized version.

```

```

*/
@Override
public void getElements(final int from, final KEY_TYPE a[], int offset, int length) {
    KEY_LIST_ITERATOR KEY_GENERIC i = listIterator(from);
    if (offset < 0) throw new ArrayIndexOutOfBoundsException("Offset (" + offset + ") is negative");
    if (offset + length > a.length) throw new ArrayIndexOutOfBoundsException("End index (" + (offset + length) + ") is
greater than array length (" + a.length + ")");
    if (from + length > size()) throw new IndexOutOfBoundsException("End index (" + (from + length) + ") is greater
than list size (" + size() + ")");
    while(length-- != 0) a[offset++] = i.NEXT_KEY();
}

/** {@inheritDoc}
 * <p>This implementation delegates to {@link #removeElements(int, int)}.*/
@Override
public void clear() {
    removeElements(0, size());
}

#if ! KEY_CLASS_Reference
private boolean valEquals(final Object a, final Object b) {
    return a == null ? b == null : a.equals(b);
}
#endif

/** Returns the hash code for this list, which is identical to {@link java.util.List#hashCode()}.
 *
 * @return the hash code for this list.
 */
@Override
public int hashCode() {
    KEY_ITERATOR KEY_GENERIC i = iterator();
    int h = 1, s = size();
    while (s-- != 0) {
        KEY_GENERIC_TYPE k = i.NEXT_KEY();
        h = 31 * h + KEY2JAVAHASH(k);
    }
    return h;
}

@Override
public boolean equals(final Object o) {
    if (o == this) return true;
    if (!(o instanceof List)) return false;

    final List<?> l = (List<?>)o;

```

```

int s = size();
if (s != l.size()) return false;

#if KEYS_PRIMITIVE
if (l instanceof LIST) {
    final KEY_LIST_ITERATOR KEY_GENERIC i1 = listIterator(), i2 = ((LIST KEY_GENERIC)l).listIterator();
    while(s-- != 0) if (i1.NEXT_KEY() != i2.NEXT_KEY()) return false;
    return true;
}
#endif

final ListIterator<?> i1 = listIterator(), i2 = l.listIterator();

#if KEY_CLASS_Reference
while(s-- != 0) if (i1.next() != i2.next()) return false;
#else
while(s-- != 0) if (! valEquals(i1.next(), i2.next())) return false;
#endif
return true;
}

#if ! KEY_CLASS_Reference
/** Compares this list to another object. If the
 * argument is a {@link java.util.List}, this method performs a lexicographical comparison; otherwise,
 * it throws a {@code ClassCastException}.
 *
 * @param l a list.
 * @return if the argument is a {@link java.util.List}, a negative integer,
 * zero, or a positive integer as this list is lexicographically less than, equal
 * to, or greater than the argument.
 * @throws ClassCastException if the argument is not a list.
 */

```

SUPPRESS_WARNINGS_KEY_UNCHECKED

```

@Override
public int compareTo(final List<? extends KEY_GENERIC_CLASS> l) {
    if (l == this) return 0;

    if (l instanceof LIST) {

        final KEY_LIST_ITERATOR KEY_GENERIC i1 = listIterator(), i2 = ((LIST KEY_GENERIC)l).listIterator();
        int r;
        KEY_GENERIC_TYPE e1, e2;

        while(i1.hasNext() && i2.hasNext()) {
            e1 = i1.NEXT_KEY();
            e2 = i2.NEXT_KEY();
            if ((r = KEY_CMP(e1, e2)) != 0) return r;
        }
    }
}

```

```

    }
    return i2.hasNext() ? -1 : (i1.hasNext() ? 1 : 0);
}

ListIterator<? extends KEY_GENERIC_CLASS> i1 = listIterator(), i2 = l.listIterator();
int r;

while(i1.hasNext() && i2.hasNext()) {
    if ((r = ((Comparable<? super KEY_GENERIC_CLASS>)i1.next()).compareTo(i2.next())) != 0) return r;
}
return i2.hasNext() ? -1 : (i1.hasNext() ? 1 : 0);
}
#endif

@Override
public void push(final KEY_GENERIC_TYPE o) {
    add(o);
}

@Override
public KEY_GENERIC_TYPE POP() {
    if (isEmpty()) throw new NoSuchElementException();
    return REMOVE_KEY(size() - 1);
}

@Override
public KEY_GENERIC_TYPE TOP() {
    if (isEmpty()) throw new NoSuchElementException();
    return GET_KEY(size() - 1);
}

@Override
public KEY_GENERIC_TYPE PEEK(final int i) {
    return GET_KEY(size() - 1 - i);
}

#if KEYS_PRIMITIVE

/** Removes a single instance of the specified element from this collection, if it is present (optional operation).
 * <p>This implementation delegates to {@code indexOf()}.
 * @see List#remove(Object)
 */
@Override
public boolean rem(final KEY_TYPE k) {
    int index = indexOf(k);
    if (index == -1) return false;
    REMOVE_KEY(index);
    return true;
}

```

```

}

@Override
public boolean addAll(int index, final COLLECTION c) {
    ensureIndex(index);
    final KEY_ITERATOR i = c.iterator();
    final boolean retVal = i.hasNext();
    while(i.hasNext()) add(index++, i.NEXT_KEY());
    return retVal;
}

/** {@inheritDoc}
 *
 * <p>This implementation delegates to the type-specific version of {@link List#addAll(int, Collection)}.
 */
@Override
public boolean addAll(final int index, final LIST l) {
    return addAll(index, (COLLECTION)l);
}

/** {@inheritDoc}
 *
 * <p>This implementation delegates to the type-specific version of {@link List#addAll(int, Collection)}.
 */
@Override
public boolean addAll(final COLLECTION c) {
    return addAll(size(), c);
}

/** {@inheritDoc}
 *
 * <p>This implementation delegates to the type-specific list version of {@link List#addAll(int, Collection)}.
 */
@Override
public boolean addAll(final LIST l) {
    return addAll(size(), l);
}

#endif

@Override
public String toString() {
    final StringBuilder s = new StringBuilder();
    final KEY_ITERATOR KEY_GENERIC i = iterator();
    int n = size();
    KEY_GENERIC_TYPE k;
    boolean first = true;

```

```

s.append("[");

while(n-- != 0) {
    if (first) first = false;
    else s.append(", ");
    k = i.NEXT_KEY();
#if KEYS_REFERENCE
    if (this == k) s.append("(this list)"); else
#endif
    s.append(String.valueOf(k));
}

s.append("]");
return s.toString();
}

/** A class implementing a sublist view. */
public static class SUBLIST KEY_GENERIC extends ABSTRACT_LIST KEY_GENERIC implements
java.io.Serializable {
    private static final long serialVersionUID = -7046029254386353129L;
    /** The list this sublist restricts. */
    protected final LIST KEY_GENERIC l;
    /** Initial (inclusive) index of this sublist. */
    protected final int from;
    /** Final (exclusive) index of this sublist. */
    protected int to;

    public SUBLIST(final LIST KEY_GENERIC l, final int from, final int to) {
        this.l = l;
        this.from = from;
        this.to = to;
    }

    private boolean assertRange() {
        assert from <= l.size();
        assert to <= l.size();
        assert to >= from;
        return true;
    }

    @Override
    public boolean add(final KEY_GENERIC_TYPE k) {
        l.add(to, k);
        to++;
        assert assertRange();
        return true;
    }
}

```

```
@Override
public void add(final int index, final KEY_GENERIC_TYPE k) {
    ensureIndex(index);
    l.add(from + index, k);
    to++;
    assert assertRange();
}
```

```
@Override
public boolean addAll(final int index, final Collection<? extends KEY_GENERIC_CLASS> c) {
    ensureIndex(index);
    to += c.size();
    return l.addAll(from + index, c);
}
```

```
@Override
public KEY_GENERIC_TYPE GET_KEY(final int index) {
    ensureRestrictedIndex(index);
    return l.GET_KEY(from + index);
}
```

```
@Override
public KEY_GENERIC_TYPE REMOVE_KEY(final int index) {
    ensureRestrictedIndex(index);
    to--;
    return l.REMOVE_KEY(from + index);
}
```

```
@Override
public KEY_GENERIC_TYPE set(final int index, final KEY_GENERIC_TYPE k) {
    ensureRestrictedIndex(index);
    return l.set(from + index, k);
}
```

```
@Override
public int size() {
    return to - from;
}
```

```
@Override
public void getElements(final int from, final KEY_TYPE[] a, final int offset, final int length) {
    ensureIndex(from);
    if (from + length > size()) throw new IndexOutOfBoundsException("End index (" + from + length + ") is greater than list size (" + size() + ")");
    l.getElements(this.from + from, a, offset, length);
}
```

```
@Override
```

```

public void removeElements(final int from, final int to) {
    ensureIndex(from);
    ensureIndex(to);
    l.removeElements(this.from + from, this.from + to);
    this.to -= (to - from);
    assert assertRange();
}

@Override
public void addElements(int index, final KEY_GENERIC_TYPE a[], int offset, int length) {
    ensureIndex(index);
    l.addElements(this.from + index, a, offset, length);
    this.to += length;
    assert assertRange();
}

@Override
public KEY_LIST_ITERATOR KEY_GENERIC listIterator(final int index) {
    ensureIndex(index);

    return new KEY_LIST_ITERATOR KEY_GENERIC() {
        int pos = index, last = -1;

        @Override
        public boolean hasNext() { return pos < size(); }
        @Override
        public boolean hasPrevious() { return pos > 0; }
        @Override
        public KEY_GENERIC_TYPE NEXT_KEY() { if (! hasNext()) throw new NoSuchElementException(); return
l.GET_KEY(from + (last = pos++)); }
        @Override
        public KEY_GENERIC_TYPE PREV_KEY() { if (! hasPrevious()) throw new NoSuchElementException(); return
l.GET_KEY(from + (last = --pos)); }
        @Override
        public int nextIndex() { return pos; }
        @Override
        public int previousIndex() { return pos - 1; }
        @Override
        public void add(KEY_GENERIC_TYPE k) {
            if (last == -1) throw new IllegalStateException();
            SUBLIST.this.add(pos++, k);
            last = -1;
            assert assertRange();
        }
        @Override
        public void set(KEY_GENERIC_TYPE k) {
            if (last == -1) throw new IllegalStateException();
            SUBLIST.this.set(last, k);
        }
    };
}

```

```

    }
    @Override
    public void remove() {
        if (last == -1) throw new IllegalStateException();
        SUBLIST.this.REMOVE_KEY(last);
        /* If the last operation was a next(), we are removing an element *before* us, and we must decrease pos
correspondingly. */
        if (last < pos) pos--;
        last = -1;
        assert assertRange();
    }
};
}

@Override
public LIST KEY_GENERIC subList(final int from, final int to) {
    ensureIndex(from);
    ensureIndex(to);
    if (from > to) throw new IllegalArgumentException("Start index (" + from + ") is greater than end index (" + to +
"");

    return new SUBLIST KEY_GENERIC_DIAMOND(this, from, to);
}

#if KEYS_PRIMITIVE

@Override
public boolean rem(final KEY_TYPE k) {
    int index = indexOf(k);
    if (index == -1) return false;
    to--;
    l.REMOVE_KEY(from + index);
    assert assertRange();
    return true;
}

@Override
public boolean addAll(final int index, final COLLECTION c) {
    ensureIndex(index);
    return super.addAll(index, c);
}

@Override
public boolean addAll(final int index, final LIST l) {
    ensureIndex(index);
    return super.addAll(index, l);
}
#endif

```

```
}
```

```
}
```

Found in path(s):

```
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-  
a775574/drv/AbstractList.drv
```

No license file was found, but licenses were detected in source scan.

```
/*
```

```
* Copyright (C) 2002-2017 Sebastiano Vigna
```

```
*
```

```
* Licensed under the Apache License, Version 2.0 (the "License");
```

```
* you may not use this file except in compliance with the License.
```

```
* You may obtain a copy of the License at
```

```
*
```

```
* http://www.apache.org/licenses/LICENSE-2.0
```

```
*
```

```
* Unless required by applicable law or agreed to in writing, software
```

```
* distributed under the License is distributed on an "AS IS" BASIS,
```

```
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
```

```
* See the License for the specific language governing permissions and
```

```
* limitations under the License.
```

```
*/
```

```
package PACKAGE;
```

```
import java.util.Collection;
```

```
import java.util.Set;
```

```
/** A class providing static methods and objects that do useful things with type-specific sets.
```

```
*
```

```
* @see java.util.Collections
```

```
*/
```

```
public final class SETS {
```

```
    private SETS() {}
```

```
    /** An immutable class representing the empty set and implementing a type-specific set interface.
```

```
    *
```

```
    * <p>This class may be useful to implement your own in case you subclass
```

```
    * a type-specific set.
```

```
    */
```

```
    public static class EmptySet KEY_GENERIC extends COLLECTIONS.EmptyCollection KEY_GENERIC
```

```

implements SET KEY_GENERIC, java.io.Serializable, Cloneable {
    private static final long serialVersionUID = -7046029254386353129L;

    protected EmptySet() {}

    @Override
    public boolean remove(KEY_TYPE ok) { throw new UnsupportedOperationException(); }

    @Override
    public Object clone() { return EMPTY_SET; }

    @Override
    @SuppressWarnings("rawtypes")
    public boolean equals(final Object o) { return o instanceof Set && ((Set)o).isEmpty(); }

#if KEYS_PRIMITIVE
    @Deprecated
    @Override
    public boolean rem(final KEY_TYPE k) { return super.rem(k); }
#endif

    private Object readResolve() { return EMPTY_SET; }
}

/** An empty set (immutable). It is serializable and cloneable.
 */
SUPPRESS_WARNINGS_KEY_RAWTYPES
public static final EmptySet EMPTY_SET = new EmptySet();

#if KEYS_REFERENCE
/** Returns an empty set (immutable). It is serializable and cloneable.
 *
 * <p>This method provides a typesafe access to { @link #EMPTY_SET}.
 * @return an empty set (immutable).
 */
@Override
public static KEY_GENERIC SET KEY_GENERIC emptySet() {
    return EMPTY_SET;
}
#endif

/** An immutable class representing a type-specific singleton set.
 *
 * <p>This class may be useful to implement your own in case you subclass
 * a type-specific set.
 */

```

```

public static class Singleton KEY_GENERIC extends ABSTRACT_SET KEY_GENERIC implements
java.io.Serializable, Cloneable {

    private static final long serialVersionUID = -7046029254386353129L;

    protected final KEY_GENERIC_TYPE element;

    protected Singleton(final KEY_GENERIC_TYPE element) {
        this.element = element;
    }

    @Override
    public boolean contains(final KEY_TYPE k) { return KEY_EQUALS(k, element); }

    @Override
    public boolean remove(final KEY_TYPE k) { throw new UnsupportedOperationException(); }

    @Override
    public KEY_LIST_ITERATOR KEY_GENERIC iterator() { return ITERATORS.singleton(element); }

    @Override
    public int size() { return 1; }

    @Override
    public boolean addAll(final Collection<? extends KEY_GENERIC_CLASS> c) { throw new
UnsupportedOperationException(); }

    @Override
    public boolean removeAll(final Collection<?> c) { throw new UnsupportedOperationException(); }

    @Override
    public boolean retainAll(final Collection<?> c) { throw new UnsupportedOperationException(); }

#if KEYS_PRIMITIVE

    @Override
    public boolean addAll(final COLLECTION c) { throw new UnsupportedOperationException(); }

    @Override
    public boolean removeAll(final COLLECTION c) { throw new UnsupportedOperationException(); }

    @Override
    public boolean retainAll(final COLLECTION c) { throw new UnsupportedOperationException(); }

#endif

    @Override
    public Object clone() { return this; }

```

```

}

/** Returns a type-specific immutable set containing only the specified element. The returned set is serializable and
cloneable.
 *
 * @param element the only element of the returned set.
 * @return a type-specific immutable set containing just { @code element}.
 */

public static KEY_GENERIC SET KEY_GENERIC singleton(final KEY_GENERIC_TYPE element) {
    return new Singleton KEY_GENERIC_DIAMOND(element);
}

#if KEYS_PRIMITIVE

/** Returns a type-specific immutable set containing only the specified element. The returned set is serializable and
cloneable.
 *
 * @param element the only element of the returned set.
 * @return a type-specific immutable set containing just { @code element}.
 */

public static KEY_GENERIC SET KEY_GENERIC singleton(final KEY_GENERIC_CLASS element) {
    return new Singleton KEY_GENERIC_DIAMOND(KEY_CLASS2TYPE(element));
}

#endif

/** A synchronized wrapper class for sets. */

public static class SynchronizedSet KEY_GENERIC extends COLLECTIONS.SynchronizedCollection
KEY_GENERIC implements SET KEY_GENERIC, java.io.Serializable {

    private static final long serialVersionUID = -7046029254386353129L;

    protected SynchronizedSet(final SET KEY_GENERIC s, final Object sync) {
        super(s, sync);
    }

    protected SynchronizedSet(final SET KEY_GENERIC s) {
        super(s);
    }

    @Override
    public boolean remove(final KEY_TYPE k) { synchronized(sync) { return collection.REMOVE(k); } }

#if KEYS_PRIMITIVE
    @Deprecated

```

```

@Override
public boolean rem(final KEY_TYPE k) { return super.rem(k); }
#endif
}

/** Returns a synchronized type-specific set backed by the given type-specific set.
 *
 * @param s the set to be wrapped in a synchronized set.
 * @return a synchronized view of the specified set.
 * @see java.util.Collections#synchronizedSet(Set)
 */
public static KEY_GENERIC SET KEY_GENERIC synchronize(final SET KEY_GENERIC s) { return new
SynchronizedSet KEY_GENERIC_DIAMOND(s); }

/** Returns a synchronized type-specific set backed by the given type-specific set, using an assigned object to
synchronize.
 *
 * @param s the set to be wrapped in a synchronized set.
 * @param sync an object that will be used to synchronize the access to the set.
 * @return a synchronized view of the specified set.
 * @see java.util.Collections#synchronizedSet(Set)
 */
public static KEY_GENERIC SET KEY_GENERIC synchronize(final SET KEY_GENERIC s, final Object sync) {
return new SynchronizedSet KEY_GENERIC_DIAMOND(s, sync); }

/** An unmodifiable wrapper class for sets. */

public static class UnmodifiableSet KEY_GENERIC extends COLLECTIONS.UnmodifiableCollection
KEY_GENERIC implements SET KEY_GENERIC, java.io.Serializable {

private static final long serialVersionUID = -7046029254386353129L;

protected UnmodifiableSet(final SET KEY_GENERIC s) {
super(s);
}

@Override
public boolean remove(final KEY_TYPE k) { throw new UnsupportedOperationException(); }

@Override
public boolean equals(final Object o) { if (o == this) return true; return collection.equals(o); }

@Override
public int hashCode() { return collection.hashCode(); }

```

```

#if KEYS_PRIMITIVE
    @Deprecated
    @Override
    public boolean rem(final KEY_TYPE k) { return super.rem(k); }
#endif
}

/** Returns an unmodifiable type-specific set backed by the given type-specific set.
 *
 * @param s the set to be wrapped in an unmodifiable set.
 * @return an unmodifiable view of the specified set.
 * @see java.util.Collections#unmodifiableSet(Set)
 */
public static KEY_GENERIC SET KEY_GENERIC unmodifiable(final SET KEY_GENERIC s) { return new
UnmodifiableSet KEY_GENERIC_DIAMOND(s); }

#ifdef TEST

    private static KEY_TYPE genKey() {
#if KEY_CLASS_Byte || KEY_CLASS_Short || KEY_CLASS_Character
        return (KEY_TYPE)(r.nextInt());
#elif KEYS_PRIMITIVE
        return r.NEXT_KEY();
#elif KEY_CLASS_Object
        return Integer.toBinaryString(r.nextInt());
#else
        return new java.io.Serializable() {};
#endif
    }

    private static void test() {
        int n = 100;
        int c;
        KEY_TYPE k = genKey();
        Singleton m = new Singleton(k);
        Set t = java.util.Collections.singleton(KEY2OBJ(k));

        long ms;
        boolean mThrowsIllegal, tThrowsIllegal, mThrowsNoElement, tThrowsNoElement, mThrowsIndex, tThrowsIndex,
        mThrowsUnsupp, tThrowsUnsupp;
        boolean rt = false, rm = false;

```

```

/* Now we check that m and t are equal. */
if (!m.equals(t) || !t.equals(m)) System.err.println("m: " + m + " t: " + t);

ensure(m.equals(t), "Error (" + seed + "): ! m.equals(t) at start");
ensure(t.equals(m), "Error (" + seed + "): ! t.equals(m) at start");

/* Now we check that m actually holds that data. */
for(java.util.Iterator i=t.iterator(); i.hasNext(); ) {
    ensure(m.contains(i.next()), "Error (" + seed + "): m and t differ on an entry after insertion (iterating on t)");
}

/* Now we check that m actually holds that data, but iterating on m. */
for(java.util.Iterator i=m.iterator(); i.hasNext(); ) {
    ensure(t.contains(i.next()), "Error (" + seed + "): m and t differ on an entry after insertion (iterating on m)");
}

/* Now we check that inquiries about random data give the same answer in m and t. For
m we use the polymorphic method. */

for(int i=0; i<n; i++) {
    KEY_TYPE T = genKey();

    mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;

    try {
        m.contains(T);
    }
    catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
    catch (IllegalArgumentException e) { mThrowsIllegal = true; }
    catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }

    try {
        t.contains(KEY2OBJ(T));
    }
    catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
    catch (IllegalArgumentException e) { tThrowsIllegal = true; }
    catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }

    ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + seed + "): contains() divergence in
java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ") " + m);
    ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + seed + "): contains() divergence in IllegalArgumentException
for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
    ensure(mThrowsIndex == tThrowsIndex, "Error (" + seed + "): contains() divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
    if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex) ensure(m.contains(KEY2OBJ(T)) ==
t.contains(KEY2OBJ(T)), "Error (" + seed + "): divergence in keys between t and m (polymorphic method) " + m);
}

```

```

/* Again, we check that inquiries about random data give the same answer in m and t, but
   for m we use the standard method. */

for(int i=0; i<n; i++) {
    KEY_TYPE T = genKey();

    mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;

    try {
        m.contains(KEY2OBJ(T));
    }
    catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
    catch (IllegalArgumentException e) { mThrowsIllegal = true; }
    catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
    catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

    try {
        t.contains(KEY2OBJ(T));
    }
    catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
    catch (IllegalArgumentException e) { tThrowsIllegal = true; }
    catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
    catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }

    ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + seed + "): contains() divergence in
java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ") " + m);
    ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + seed + "): contains() divergence in IllegalArgumentException
for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
    ensure(mThrowsIndex == tThrowsIndex, "Error (" + seed + "): contains() divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
    ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + seed + "): contains() divergence in
UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);
    if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex && !mThrowsUnsupp)
    ensure(m.contains(KEY2OBJ(T)) == t.contains(KEY2OBJ(T)), "Error (" + seed + "): divergence between t and m
(standard method) " + m);
}

/* Now we add and remove random data in m and t, checking that the result is the same. */

for(int i=0; i<20*n; i++) {
    KEY_TYPE T = genKey();

    mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;

    try {

```

```

    rm = m.add(KEY2OBJ(T));
}
catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
catch (IllegalArgumentException e) { mThrowsIllegal = true; }
catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

try {
    rt = t.add(KEY2OBJ(T));
}
catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
catch (IllegalArgumentException e) { tThrowsIllegal = true; }
catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }

ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + seed + "): add() divergence in
java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ") " + m);
ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + seed + "): add() divergence in IllegalArgumentException for "
+ T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
ensure(mThrowsIndex == tThrowsIndex, "Error (" + seed + "): add() divergence in IndexOutOfBoundsException
for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + seed + "): add() divergence in
UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);
if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex && !mThrowsUnsupp) ensure(rm == rt, "Error
(" + seed + "): divergence in add() between t and m " + m);

T = genKey();

mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;

try {
    rm = m.remove(KEY2OBJ(T));
}
catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
catch (IllegalArgumentException e) { mThrowsIllegal = true; }
catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

try {
    rt = t.remove(KEY2OBJ(T));
}
catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
catch (IllegalArgumentException e) { tThrowsIllegal = true; }
catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }

if (!KEY_EQUALS(T, k) && mThrowsUnsupp && !tThrowsUnsupp) mThrowsUnsupp = false; // Stupid bug in

```

Collections.singleton()

```
    ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + seed + "): remove() divergence in
java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ") " + m);
    ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + seed + "): remove() divergence in IllegalArgumentException
for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
    ensure(mThrowsIndex == tThrowsIndex, "Error (" + seed + "): remove() divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
    ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + seed + "): remove() divergence in
UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);
    if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex && !mThrowsUnsupp) ensure(rm == rt, "Error
(" + seed + "): divergence in remove() between t and m " + m);
}
```

```
ensure(m.equals(t), "Error (" + seed + "): ! m.equals(t) after removal " + m);
```

```
ensure(t.equals(m), "Error (" + seed + "): ! t.equals(m) after removal " + m);
```

```
/* Now we add and remove random collections in m and t, checking that the result is the same. */
```

```
for(int i=0; i<20*n; i++) {
    KEY_TYPE T = genKey();
```

```
    mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;
```

```
    try {
        rm = m.addAll(java.util.Collections.singleton(KEY2OBJ(T)));
    }
    catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
    catch (IllegalArgumentException e) { mThrowsIllegal = true; }
    catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
    catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }
```

```
    try {
        rt = t.addAll(java.util.Collections.singleton(KEY2OBJ(T)));
    }
    catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
    catch (IllegalArgumentException e) { tThrowsIllegal = true; }
    catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
    catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }
```

```
    ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + seed + "): addAll() divergence in
java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ") " + m);
    ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + seed + "): addAll() divergence in IllegalArgumentException
for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
    ensure(mThrowsIndex == tThrowsIndex, "Error (" + seed + "): addAll() divergence in IndexOutOfBoundsException
for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
    ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + seed + "): addAll() divergence in
```

```

UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);
    if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex && !mThrowsUnsupp) ensure(rm == rt, "Error
(" + seed + "): divergence in addAll() between t and m " + m);

    T = genKey();

    mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;

    try {
        rm = m.removeAll(java.util.Collections.singleton(KEY2OBJ(T)));
    }
    catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
    catch (IllegalArgumentException e) { mThrowsIllegal = true; }
    catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
    catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

    try {
        rt = t.removeAll(java.util.Collections.singleton(KEY2OBJ(T)));
    }
    catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
    catch (IllegalArgumentException e) { tThrowsIllegal = true; }
    catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
    catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }

    if (!KEY_EQUALS(T, k) && mThrowsUnsupp && !tThrowsUnsupp) mThrowsUnsupp = false; // Stupid bug in
Collections.singleton()

    ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + seed + "): removeAll() divergence in
java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ") " + m);
    ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + seed + "): removeAll() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
    ensure(mThrowsIndex == tThrowsIndex, "Error (" + seed + "): removeAll() divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
    ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + seed + "): removeAll() divergence in
UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);
    if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex && !mThrowsUnsupp) ensure(rm == rt, "Error
(" + seed + "): divergence in removeAll() between t and m " + m);
    }

    ensure(m.equals(t), "Error (" + seed + "): ! m.equals(t) after set removal " + m);
    ensure(t.equals(m), "Error (" + seed + "): ! t.equals(m) after set removal " + m);

    /* Now we check that m actually holds the same data. */

    for(java.util.Iterator i=t.iterator(); i.hasNext();) {
        ensure(m.contains(i.next()), "Error (" + seed + "): m and t differ on an entry after removal (iterating on t)");
    }

```

```

/* Now we check that m actually holds that data, but iterating on m. */

for(java.util.Iterator i=m.iterator(); i.hasNext(); ) {
    ensure(t.contains(i.next()), "Error (" + seed + "): m and t differ on an entry after removal (iterating on m)");
}

if (m instanceof Singleton) {
    ensure(m.equals(((Singleton)m).clone()), "Error (" + seed + "): m does not equal m.clone()");
    ensure(((Singleton)m).clone().equals(m), "Error (" + seed + "): m.clone() does not equal m");
}

int h = m.hashCode();

/* Now we save and read m. */

SET m2 = null;

try {
    java.io.File ff = new java.io.File("it.unimi.dsi.fastutil.test");
    java.io.OutputStream os = new java.io.FileOutputStream(ff);
    java.io.ObjectOutputStream oos = new java.io.ObjectOutputStream(os);

    oos.writeObject(m);
    oos.close();

    java.io.InputStream is = new java.io.FileInputStream(ff);
    java.io.ObjectInputStream ois = new java.io.ObjectInputStream(is);

    m2 = (SET)ois.readObject();
    ois.close();
    ff.delete();
}
catch(Exception e) {
    e.printStackTrace();
    System.exit(1);
}

#ifdef !KEY_CLASS_Reference

ensure(m2.hashCode() == h, "Error (" + seed + "): hashCode() changed after save/read");

/* Now we check that m2 actually holds that data. */

ensure(m2.equals(t), "Error (" + seed + "): ! m2.equals(t) after save/read");
ensure(t.equals(m2), "Error (" + seed + "): ! t.equals(m2) after save/read");
#endif

```

```

System.out.println("Test OK");
return;
}

private static long seed = System.currentTimeMillis();
private static java.util.Random r = new java.util.Random(seed);

private static java.text.NumberFormat format = new java.text.DecimalFormat("#,###.00");
private static java.text.FieldPosition fp = new java.text.FieldPosition(0);

private static String format(double d) {
    StringBuffer s = new StringBuffer();
    return format.format(d, s, fp).toString();
}

private static void fatal(String msg) {
    System.out.println(msg);
    System.exit(1);
}

private static void ensure(boolean cond, String msg) {
    if (cond) return;
    fatal(msg);
}

/** This method expects as first argument a lower-cased type (e.g., "int"),
 * and as second optional argument a seed. */

public static void main(String arg[]) throws Exception {
    if (arg.length > 1) r = new java.util.Random(seed = Long.parseLong(arg[1]));

    try {
        test();
    } catch(Throwable e) {
        e.printStackTrace(System.err);
        System.err.println("seed: " + seed);
    }
}

#endif

}

```

Found in path(s):

```

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-
a775574/drv/Sets.drv

```

No license file was found, but licenses were detected in source scan.

```

/*
 * Copyright (C) 2004-2017 Sebastiano Vigna
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

/** Loads elements from a given fast buffered reader, storing them in a given array fragment.
 *
 * @param reader a buffered reader.
 * @param array an array which will be filled with data from { @code reader}.
 * @param offset the index of the first element of { @code array} to be filled.
 * @param length the number of elements of { @code array} to be filled.
 * @return the number of elements actually read from { @code reader} (it might be less than { @code length} if
 { @code reader} ends).
 */
public static int LOAD_KEYS(final BufferedReader reader, final KEY_TYPE[] array, final int offset, final int
length) throws IOException {
    PACKAGE.ARRAYS.ensureOffsetLength(array, offset, length);
    int i = 0;
    String s;
    try {
        for(i = 0; i < length; i++)
            if ((s = reader.readLine()) != null) array[i + offset] = KEY_CLASS.PARSE_KEY(s.trim());
            else break;
    }
    catch(EOFException itsOk) {}
    return i;
}

/** Loads elements from a given buffered reader, storing them in a given array.
 *
 * @param reader a buffered reader.
 * @param array an array which will be filled with data from { @code reader}.
 * @return the number of elements actually read from { @code reader} (it might be less than the array length if
 { @code reader} ends).
 */

```

```

public static int LOAD_KEYS(final BufferedReader reader, final KEY_TYPE[] array) throws IOException {
    return LOAD_KEYS(reader, array, 0, array.length);
}

/** Loads elements from a file given by a {@link File} object, storing them in a given array fragment.
 *
 * @param file a file.
 * @param array an array which will be filled with data from the specified file.
 * @param offset the index of the first element of {@code array} to be filled.
 * @param length the number of elements of {@code array} to be filled.
 * @return the number of elements actually read from the given file (it might be less than {@code length} if the file
is too short).
 */
public static int LOAD_KEYS(final File file, final KEY_TYPE[] array, final int offset, final int length) throws
IOException {
    final BufferedReader reader = new BufferedReader(new FileReader(file));
    final int result = LOAD_KEYS(reader, array, offset, length);
    reader.close();

    return result;
}

/** Loads elements from a file given by a filename, storing them in a given array fragment.
 *
 * @param filename a filename.
 * @param array an array which will be filled with data from the specified file.
 * @param offset the index of the first element of {@code array} to be filled.
 * @param length the number of elements of {@code array} to be filled.
 * @return the number of elements actually read from the given file (it might be less than {@code length} if the file
is too short).
 */
public static int LOAD_KEYS(final CharSequence filename, final KEY_TYPE[] array, final int offset, final int
length) throws IOException {
    return LOAD_KEYS(new File(filename.toString()), array, offset, length);
}

/** Loads elements from a file given by a {@link File} object, storing them in a given array.
 *
 * @param file a file.
 * @param array an array which will be filled with data from the specified file.
 * @return the number of elements actually read from the given file (it might be less than the array length if the file
is too short).
 */
public static int LOAD_KEYS(final File file, final KEY_TYPE[] array) throws IOException {
    return LOAD_KEYS(file, array, 0, array.length);
}

/** Loads elements from a file given by a filename, storing them in a given array.

```

```

*
* @param filename a filename.
* @param array an array which will be filled with data from the specified file.
* @return the number of elements actually read from the given file (it might be less than the array length if the file
is too short).
*/
public static int LOAD_KEYS(final CharSequence filename, final KEY_TYPE[] array) throws IOException {
    return LOAD_KEYS(filename, array, 0, array.length);
}

/** Stores an array fragment to a given print stream.
*
* @param array an array whose elements will be written to { @code stream}.
* @param offset the index of the first element of { @code array} to be written.
* @param length the number of elements of { @code array} to be written.
* @param stream a print stream.
*/
public static void STORE_KEYS(final KEY_TYPE array[], final int offset, final int length, final PrintStream
stream) {
    PACKAGE.ARRAYS.ensureOffsetLength(array, offset, length);
    for(int i = 0; i < length; i++) stream.println(array[offset + i]);
}

/** Stores an array to a given print stream.
*
* @param array an array whose elements will be written to { @code stream}.
* @param stream a print stream.
*/
public static void STORE_KEYS(final KEY_TYPE array[], final PrintStream stream) {
    STORE_KEYS(array, 0, array.length, stream);
}

/** Stores an array fragment to a file given by a { @link File} object.
*
* @param array an array whose elements will be written to { @code filename}.
* @param offset the index of the first element of { @code array} to be written.
* @param length the number of elements of { @code array} to be written.
* @param file a file.
*/
public static void STORE_KEYS(final KEY_TYPE array[], final int offset, final int length, final File file) throws
IOException {
    final PrintStream stream = new PrintStream(new FastBufferedOutputStream(new FileOutputStream(file)));
    STORE_KEYS(array, offset, length, stream);
    stream.close();
}

/** Stores an array fragment to a file given by a pathname.
*

```

```

* @param array an array whose elements will be written to { @code filename}.
* @param offset the index of the first element of { @code array} to be written.
* @param length the number of elements of { @code array} to be written.
* @param filename a filename.
*/
public static void STORE_KEYS(final KEY_TYPE array[], final int offset, final int length, final CharSequence
filename) throws IOException {
    STORE_KEYS(array, offset, length, new File(filename.toString()));
}

/** Stores an array to a file given by a { @link File} object.
*
* @param array an array whose elements will be written to { @code filename}.
* @param file a file.
*/
public static void STORE_KEYS(final KEY_TYPE array[], final File file) throws IOException {
    STORE_KEYS(array, 0, array.length, file);
}

/** Stores an array to a file given by a pathname.
*
* @param array an array whose elements will be written to { @code filename}.
* @param filename a filename.
*/
public static void STORE_KEYS(final KEY_TYPE array[], final CharSequence filename) throws IOException {
    STORE_KEYS(array, 0, array.length, filename);
}

/** Stores the element returned by an iterator to a given print stream.
*
* @param i an iterator whose output will be written to { @code stream}.
* @param stream a print stream.
*/
public static void STORE_KEYS(final KEY_ITERATOR i, final PrintStream stream) {
    while(i.hasNext()) stream.println(i.NEXT_KEY());
}

/** Stores the element returned by an iterator to a file given by a { @link File} object.
*
* @param i an iterator whose output will be written to { @code filename}.
* @param file a file.
*/
public static void STORE_KEYS(final KEY_ITERATOR i, final File file) throws IOException {
    final PrintStream stream = new PrintStream(new FastBufferedOutputStream(new FileOutputStream(file)));
    STORE_KEYS(i, stream);
    stream.close();
}

```

```

/** Stores the element returned by an iterator to a file given by a pathname.
 *
 * @param i an iterator whose output will be written to { @code filename }.
 * @param filename a filename.
 */
public static void STORE_KEYS(final KEY_ITERATOR i, final CharSequence filename) throws IOException {
    STORE_KEYS(i, new File(filename.toString()));
}

```

```

/** Loads elements from a given fast buffered reader, storing them in a given big-array fragment.
 *
 * @param reader a buffered reader.
 * @param array a big array which will be filled with data from { @code reader }.
 * @param offset the index of the first element of { @code array } to be filled.
 * @param length the number of elements of { @code array } to be filled.
 * @return the number of elements actually read from { @code reader } (it might be less than { @code length } if
 { @code reader } ends).
 */
public static long LOAD_KEYS(final BufferedReader reader, final KEY_TYPE[][] array, final long offset, final
long length) throws IOException {
    PACKAGE.BIG_ARRAYS.ensureOffsetLength(array, offset, length);
    long c = 0;
    String s;
    try {
        for(int i = segment(offset); i < segment(offset + length + SEGMENT_MASK); i++) {
            final KEY_TYPE[] t = array[i];
            final int l = (int)Math.min(t.length, offset + length - start(i));
            for(int d = (int)Math.max(0, offset - start(i)); d < l; d++) {
                if ((s = reader.readLine()) != null) t[d] = KEY_CLASS.PARSE_KEY(s.trim());
                else return c;
                c++;
            }
        }
    }
    catch EOFException itsOk) {}
    return c;
}

```

```

/** Loads elements from a given buffered reader, storing them in a given array.
 *
 * @param reader a buffered reader.
 * @param array a big array which will be filled with data from { @code reader }.
 * @return the number of elements actually read from { @code reader } (it might be less than the array length if
 { @code reader } ends).
 */
public static long LOAD_KEYS(final BufferedReader reader, final KEY_TYPE[][] array) throws IOException {
    return LOAD_KEYS(reader, array, 0, PACKAGE.BIG_ARRAYS.length(array));
}

```

```

}

/** Loads elements from a file given by a {@link File} object, storing them in a given big-array fragment.
 *
 * @param file a file.
 * @param array a big array which will be filled with data from the specified file.
 * @param offset the index of the first element of {@code array} to be filled.
 * @param length the number of elements of {@code array} to be filled.
 * @return the number of elements actually read from the given file (it might be less than {@code length} if the file
 is too short).
 */
public static long LOAD_KEYS(final File file, final KEY_TYPE[][] array, final long offset, final long length)
throws IOException {
    final BufferedReader reader = new BufferedReader(new FileReader(file));
    final long result = LOAD_KEYS(reader, array, offset, length);
    reader.close();

    return result;
}

/** Loads elements from a file given by a filename, storing them in a given big-array fragment.
 *
 * @param filename a filename.
 * @param array a big array which will be filled with data from the specified file.
 * @param offset the index of the first element of {@code array} to be filled.
 * @param length the number of elements of {@code array} to be filled.
 * @return the number of elements actually read from the given file (it might be less than {@code length} if the file
 is too short).
 */
public static long LOAD_KEYS(final CharSequence filename, final KEY_TYPE[][] array, final long offset, final
long length) throws IOException {
    return LOAD_KEYS(new File(filename.toString()), array, offset, length);
}

/** Loads elements from a file given by a {@link File} object, storing them in a given array.
 *
 * @param file a file.
 * @param array a big array which will be filled with data from the specified file.
 * @return the number of elements actually read from the given file (it might be less than the array length if the file
 is too short).
 */
public static long LOAD_KEYS(final File file, final KEY_TYPE[][] array) throws IOException {
    return LOAD_KEYS(file, array, 0, PACKAGE.BIG_ARRAYS.length(array));
}

/** Loads elements from a file given by a filename, storing them in a given array.
 *
 * @param filename a filename.

```

```

* @param array a big array which will be filled with data from the specified file.
* @return the number of elements actually read from the given file (it might be less than the array length if the file
is too short).
*/
public static long LOAD_KEYS(final CharSequence filename, final KEY_TYPE[][] array) throws IOException {
    return LOAD_KEYS(filename, array, 0, PACKAGE.BIG_ARRAYS.length(array));
}

/** Stores a big-array fragment to a given print stream.
*
* @param array a big array whose elements will be written to { @code stream}.
* @param offset the index of the first element of { @code array} to be written.
* @param length the number of elements of { @code array} to be written.
* @param stream a print stream.
*/
public static void STORE_KEYS(final KEY_TYPE array[][], final long offset, final long length, final PrintStream
stream) {
    PACKAGE.BIG_ARRAYS.ensureOffsetLength(array, offset, length);

    for(int i = segment(offset); i < segment(offset + length + SEGMENT_MASK); i++) {
        final KEY_TYPE[] t = array[i];
        final int l = (int)Math.min(t.length, offset + length - start(i));
        for(int d = (int)Math.max(0, offset - start(i)); d < l; d++) stream.println(t[d]);
    }
}

/** Stores a big array to a given print stream.
*
* @param array a big array whose elements will be written to { @code stream}.
* @param stream a print stream.
*/
public static void STORE_KEYS(final KEY_TYPE array[][], final PrintStream stream) {
    STORE_KEYS(array, 0, PACKAGE.BIG_ARRAYS.length(array), stream);
}

/** Stores a big-array fragment to a file given by a { @link File} object.
*
* @param array a big array whose elements will be written to { @code filename}.
* @param offset the index of the first element of { @code array} to be written.
* @param length the number of elements of { @code array} to be written.
* @param file a file.
*/
public static void STORE_KEYS(final KEY_TYPE array[][], final long offset, final long length, final File file)
throws IOException {
    final PrintStream stream = new PrintStream(new FastBufferedOutputStream(new FileOutputStream(file)));
    STORE_KEYS(array, offset, length, stream);
    stream.close();
}

```

```

/** Stores a big-array fragment to a file given by a pathname.
 *
 * @param array a big array whose elements will be written to { @code filename }.
 * @param offset the index of the first element of { @code array } to be written.
 * @param length the number of elements of { @code array } to be written.
 * @param filename a filename.
 */
public static void STORE_KEYS(final KEY_TYPE array[], final long offset, final long length, final
CharSequence filename) throws IOException {
    STORE_KEYS(array, offset, length, new File(filename.toString()));
}

/** Stores a big array to a file given by a { @link File } object.
 *
 * @param array a big array whose elements will be written to { @code filename }.
 * @param file a file.
 */
public static void STORE_KEYS(final KEY_TYPE array[], final File file) throws IOException {
    STORE_KEYS(array, 0, PACKAGE.BIG_ARRAYS.length(array), file);
}

/** Stores a big array to a file given by a pathname.
 *
 * @param array a big array whose elements will be written to { @code filename }.
 * @param filename a filename.
 */
public static void STORE_KEYS(final KEY_TYPE array[], final CharSequence filename) throws IOException {
    STORE_KEYS(array, 0, PACKAGE.BIG_ARRAYS.length(array), filename);
}

/** A wrapper that exhibits the content of a reader as a type-specific iterator. */

private static final class KEY_READER_WRAPPER implements KEY_ITERATOR {
    private final BufferedReader reader;
    private boolean toAdvance = true;
    private String s;
    private KEY_TYPE next;

    public KEY_READER_WRAPPER(final BufferedReader reader) {
        this.reader = reader;
    }

    @Override
    public boolean hasNext() {
        if (!toAdvance) return s != null;

```

```

toAdvance = false;

try {
    s = reader.readLine();
}
catch(EOFException itsOk) {}
catch(IOException rethrow) { throw new RuntimeException(rethrow); }

if (s == null) return false;

next = KEY_CLASS.PARSE_KEY(s.trim());
return true;
}

@Override
public KEY_TYPE NEXT_KEY() {
    if (! hasNext()) throw new NoSuchElementException();
    toAdvance = true;
    return next;
}
}

/** Wraps the given buffered reader into an iterator.
 *
 * @param reader a buffered reader.
 */
public static KEY_ITERATOR AS_KEY_ITERATOR(final BufferedReader reader) {
    return new KEY_READER_WRAPPER(reader);
}

/** Wraps a file given by a {@link File} object into an iterator.
 *
 * @param file a file.
 */
public static KEY_ITERATOR AS_KEY_ITERATOR(final File file) throws IOException {
    return new KEY_READER_WRAPPER(new BufferedReader(new FileReader(file)));
}

/** Wraps a file given by a pathname into an iterator.
 *
 * @param filename a filename.
 */
public static KEY_ITERATOR AS_KEY_ITERATOR(final CharSequence filename) throws IOException {
    return AS_KEY_ITERATOR(new File(filename.toString()));
}
}

```

```
/** Wraps a file given by a {@link File} object into an iterable object.
```

```
*
```

```
* @param file a file.
```

```
*/
```

```
public static KEY_ITERABLE AS_KEY_ITERABLE(final File file) {  
    return () -> {  
        try { return AS_KEY_ITERATOR(file); }  
        catch(IOException e) { throw new RuntimeException(e); }  
    };  
}
```

```
/** Wraps a file given by a pathname into an iterable object.
```

```
*
```

```
* @param filename a filename.
```

```
*/
```

```
public static KEY_ITERABLE AS_KEY_ITERABLE(final CharSequence filename) {  
    return () -> {  
        try { return AS_KEY_ITERATOR(filename); }  
        catch(IOException e) { throw new RuntimeException(e); }  
    };  
}
```

Found in path(s):

```
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-  
a775574/drv/TextIOFragment.drv
```

No license file was found, but licenses were detected in source scan.

```
/*
```

```
* Copyright (C) 2017 Sebastiano Vigna
```

```
*
```

```
* Licensed under the Apache License, Version 2.0 (the "License");
```

```
* you may not use this file except in compliance with the License.
```

```
* You may obtain a copy of the License at
```

```
*
```

```
* http://www.apache.org/licenses/LICENSE-2.0
```

```
*
```

```
* Unless required by applicable law or agreed to in writing, software
```

```
* distributed under the License is distributed on an "AS IS" BASIS,
```

```
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
```

```
* See the License for the specific language governing permissions and
```

```
* limitations under the License.
```

```
*/
```

Found in path(s):

```
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-  
a775574/test/it/unimi/dsi/fastutil/objects/ObjectOpenCustomHashSetTest.java
```

```
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-  
a775574/test/it/unimi/dsi/fastutil/ints/StripedInt2IntOpenHashMapTest.java
```

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/src/it/unimi/dsi/fastutil/SafeMath.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/ints/IntOpenHashSetTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/ints/Int2IntMapGenericAVLTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/ints/IntArrayFIFOQueueTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/doubles/DoubleOpenHashSetTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/ints/Int2ObjectMapGenericTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/objects/Object2IntOpenHashMapTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/doubles/DoubleArraysTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/objects/ObjectHeapPriorityQueueTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/ints/Int2IntMapGenericLinkedOpenHashTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/objects/ObjectBigArrayBigListTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/objects/AbstractObject2IntFunctionTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/io/FastBufferedOutputStreamTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/ints/Int2IntLinkedOpenHashMapTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/objects/Reference2ReferenceArrayMapTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/ints/Int2ObjectMapTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/objects/ObjectArraysTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/objects/Object2ObjectArrayMapTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/objects/ObjectArrayListTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/ints/Int2ObjectMapGenericArrayTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/objects/ObjectOpenHashSetTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/ints/IntListsTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/bytes/ByteArrayFrontCodedListTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/ints/IntArrayPriorityQueueTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/ints/IntArrayFrontCodedListTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/io/BinIOTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/ints/AbstractIntCollectionTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/ints/IntSetsTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/ints/Int2IntMapGenericOpenHashTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/longs/LongArrayFrontCodedListTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/bytes/ByteArraysTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/objects/ObjectAVLTreeSetTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/objects/ObjectArraySetTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/ints/Int2ObjectMapGenericOpenHashTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/BigArraysTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/shorts/ShortArraysTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/ints/IntOpenCustomHashSetTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/objects/ObjectRBTreeSetTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/doubles/DoubleBigArraysTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/ints/Int2ObjectMapGenericLinkedOpenHashTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/objects/ReferenceArraySetTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/objects/ObjectArrayPriorityQueueTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/chars/CharArrayFrontCodedListTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/ints/IntArraySetTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/ints/Int2IntMapGenericDefaultTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/ints/IntCollectionsTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/io/FastByteArrayOutputStreamTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/objects/ObjectBigArraysTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/ints/IntHeapSemiIndirectPriorityQueueTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/io/FastMultiByteArrayInputStreamTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/io/InspectableFileCachedInputStreamTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/ints/IntLinkedOpenHashSetTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/ints/IntSemiIndirectHeapsTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/ints/Int2IntMapsTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/ints/IntHeapPriorityQueueTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/objects/ObjectOpenHashBigSetTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/longs/LongArraysTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/io/TextIOTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/floats/FloatArraysTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/ints/IntBigArrayBigListTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/ints/AbstractInt2IntMapTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/ints/Int2ObjectFunctionTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/ints/Int2ObjectMapGenericAVLTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/ints/Int2IntOpenHashMapTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/ArraysTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/ints/Int2ObjectMapGenericRBTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/longs/LongOpenHashBigSetTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/floats/FloatOpenHashSetTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/ints/Int2IntMapGenericArrayTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/ints/IntLinkedOpenCustomHashSetTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/ints/Int2IntMapGenericRBTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/ints/Int2IntOpenCustomHashMapTest.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/ints/IntArraysTest.java
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/ints/IntOpenHashBigSetTest.java
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/chars/CharArraysTest.java
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/shorts/ShortArrayFrontCodedListTest.java
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/ints/Int2IntMapGenericTest.java
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/ints/IntArrayIndirectPriorityQueueTest.java
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/ints/IntBigArraysTest.java
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/HashCommonTest.java
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/io/FastBufferedInputStreamTest.java
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/objects/ObjectSetsTest.java
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/ints/IntArrayListTest.java
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/test/it/unimi/dsi/fastutil/ints/Int2ObjectMapGenericDefaultTest.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2002-2017 Sebastiano Vigna

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

package PACKAGE;

import java.util.Collection;

import java.util.Comparator;

import java.util.Iterator;

```

import java.util.SortedSet;
import java.util.NoSuchElementException;

/** A type-specific red-black tree set with a fast, small-footprint implementation.
 *
 * <p>The iterators provided by this class are type-specific { @link
 * it.unimi.dsi.fastutil.BidirectionalIterator bidirectional iterators}.
 * Moreover, the iterator returned by { @code iterator()} can be safely cast
 * to a type-specific { @linkplain java.util.ListIterator list iterator}.
 */

public class RB_TREE_SET KEY_GENERIC extends ABSTRACT_SORTED_SET KEY_GENERIC implements
java.io.Serializable, Cloneable, SORTED_SET KEY_GENERIC {

    /** A reference to the root entry. */
    protected transient Entry KEY_GENERIC tree;

    /** Number of elements in this set. */
    protected int count;

    /** The entry of the first element of this set. */
    protected transient Entry KEY_GENERIC firstEntry;

    /** The entry of the last element of this set. */
    protected transient Entry KEY_GENERIC lastEntry;

    /** This set's comparator, as provided in the constructor. */
    protected Comparator<? super KEY_GENERIC_CLASS> storedComparator;

    /** This set's actual comparator; it may differ from { @link #storedComparator} because it is
    always a type-specific comparator, so it could be derived from the former by wrapping. */
    protected transient KEY_COMPARATOR KEY_SUPER_GENERIC actualComparator;

    private static final long serialVersionUID = -7046029254386353130L;

    {
        allocatePaths();
    }

    /** Creates a new empty tree set.
    */

    public RB_TREE_SET() {
        tree = null;
        count = 0;
    }

    /** Generates the comparator that will be actually used.

```

```

*
* <p>When a given {@link Comparator} is specified and stored in {@link
* #storedComparator}, we must check whether it is type-specific. If it is
* so, we can use it directly, and we store it in {@link #actualComparator}. Otherwise,
* we adapt it using a helper static method.
*/
private void setActualComparator() {
#if KEY_CLASS_Object
    actualComparator = storedComparator;
#else
    actualComparator = COMPARATORS.AS_KEY_COMPARATOR(storedComparator);
#endif
}

/** Creates a new empty tree set with the given comparator.
*
* @param c a {@link Comparator} (even better, a type-specific comparator).
*/

public RB_TREE_SET(final Comparator<? super KEY_GENERIC_CLASS> c) {
    this();
    storedComparator = c;
    setActualComparator();
}

/** Creates a new tree set copying a given collection.
*
* @param c a collection to be copied into the new tree set.
*/

public RB_TREE_SET(final Collection<? extends KEY_GENERIC_CLASS> c) {
    this();
    addAll(c);
}

/** Creates a new tree set copying a given sorted set (and its {@link Comparator}).
*
* @param s a {@link SortedSet} to be copied into the new tree set.
*/

public RB_TREE_SET(final SortedSet<KEY_GENERIC_CLASS> s) {
    this(s.comparator());
    addAll(s);
}

/** Creates a new tree set copying a given type-specific collection.

```

```

*
* @param c a type-specific collection to be copied into the new tree set.
*/

public RB_TREE_SET(final COLLECTION KEY_EXTENDS_GENERIC c) {
    this();
    addAll(c);
}

/** Creates a new tree set copying a given type-specific sorted set (and its {@link Comparator}).
*
* @param s a type-specific sorted set to be copied into the new tree set.
*/

public RB_TREE_SET(final SORTED_SET KEY_GENERIC s) {
    this(s.comparator());
    addAll(s);
}

/** Creates a new tree set using elements provided by a type-specific iterator.
*
* @param i a type-specific iterator whose elements will fill the set.
*/

public RB_TREE_SET(final STD_KEY_ITERATOR KEY_EXTENDS_GENERIC i) {
    while(i.hasNext()) add(i.NEXT_KEY());
}

#if KEYS_PRIMITIVE

/** Creates a new tree set using elements provided by an iterator.
*
* @param i an iterator whose elements will fill the set.
*/

SUPPRESS_WARNINGS_KEY_UNCHECKED
public RB_TREE_SET(final Iterator<?> i) {
    this(ITERATORS.AS_KEY_ITERATOR(i));
}

#endif

/** Creates a new tree set and fills it with the elements of a given array using a given {@link Comparator}.
*
* @param a an array whose elements will be used to fill the set.
* @param offset the first element to use.

```

```

* @param length the number of elements to use.
* @param c a { @link Comparator } (even better, a type-specific comparator).
*/

public RB_TREE_SET(final KEY_GENERIC_TYPE[] a, final int offset, final int length, final Comparator<? super
KEY_GENERIC_CLASS> c) {
    this(c);
    ARRAYS.ensureOffsetLength(a, offset, length);
    for(int i = 0; i < length; i++) add(a[offset + i]);
}

/** Creates a new tree set and fills it with the elements of a given array.
*
* @param a an array whose elements will be used to fill the set.
* @param offset the first element to use.
* @param length the number of elements to use.
*/

public RB_TREE_SET(final KEY_GENERIC_TYPE[] a, final int offset, final int length) {
    this(a, offset, length, null);
}

/** Creates a new tree set copying the elements of an array.
*
* @param a an array to be copied into the new tree set.
*/

public RB_TREE_SET(final KEY_GENERIC_TYPE[] a) {
    this();
    int i = a.length;
    while(i-- != 0) add(a[i]);
}

/** Creates a new tree set copying the elements of an array using a given { @link Comparator }.
*
* @param a an array to be copied into the new tree set.
* @param c a { @link Comparator } (even better, a type-specific comparator).
*/

public RB_TREE_SET(final KEY_GENERIC_TYPE[] a, final Comparator<? super KEY_GENERIC_CLASS> c)
{
    this(c);
    int i = a.length;
    while(i-- != 0) add(a[i]);
}

```

```

/*
 * The following methods implements some basic building blocks used by
 * all accessors. They are (and should be maintained) identical to those used in RBTreeMap.drv.
 *
 * The add()/remove() code is derived from Ben Pfaff's GNU libavl
 * (http://www.msu.edu/~pfaffben/avl/). If you want to understand what's
 * going on, you should have a look at the literate code contained therein
 * first.
 */

/** Compares two keys in the right way.
 *
 * <p>This method uses the { @link #actualComparator } if it is non-{@code null}.
 * Otherwise, it resorts to primitive type comparisons or to { @link Comparable#compareTo(Object) compareTo()}.
 *
 * @param k1 the first key.
 * @param k2 the second key.
 * @return a number smaller than, equal to or greater than 0, as usual
 * (i.e., when  $k1 < k2$ ,  $k1 = k2$  or  $k1 > k2$ , respectively).
 */

SUPPRESS_WARNINGS_KEY_UNCHECKED
final int compare(final KEY_GENERIC_TYPE k1, final KEY_GENERIC_TYPE k2) {
    return actualComparator == null ? KEY_CMP(k1, k2) : actualComparator.compare(k1, k2);
}

/** Returns the entry corresponding to the given key, if it is in the tree; {@code null}, otherwise.
 *
 * @param k the key to search for.
 * @return the corresponding entry, or {@code null} if no entry with the given key exists.
 */

private Entry KEY_GENERIC findKey(final KEY_GENERIC_TYPE k) {
    Entry KEY_GENERIC e = tree;
    int cmp;

    while (e != null && (cmp = compare(k, e.key)) != 0)
        e = cmp < 0 ? e.left() : e.right();

    return e;
}

```

```

/** Locates a key.
 *
 * @param k a key.
 * @return the last entry on a search for the given key; this will be
 * the given key, if it present; otherwise, it will be either the smallest greater key or the greatest smaller key.
 */

final Entry KEY_GENERIC locateKey(final KEY_GENERIC_TYPE k) {
    Entry KEY_GENERIC e = tree, last = tree;
    int cmp = 0;

    while (e != null && (cmp = compare(k, e.key)) != 0) {
        last = e;
        e = cmp < 0 ? e.left() : e.right();
    }

    return cmp == 0 ? e : last;
}

/** This vector remembers the path and the direction followed during the
 * current insertion. It suffices for about 232 entries. */
private transient boolean dirPath[];
private transient Entry KEY_GENERIC nodePath[];

SUPPRESS_WARNINGS_KEY_UNCHECKED_RAWTYPES
private void allocatePaths() {
    dirPath = new boolean[64];
    nodePath = new Entry[64];
}

@Override
public boolean add(final KEY_GENERIC_TYPE k) {
    int maxDepth = 0;

    if (tree == null) { // The case of the empty tree is treated separately.
        count++;
        tree = lastEntry = firstEntry = new Entry KEY_GENERIC_DIAMOND(k);
    }
    else {
        Entry KEY_GENERIC p = tree, e;
        int cmp, i = 0;

        while(true) {
            if ((cmp = compare(k, p.key)) == 0) {
                // We clean up the node path, or we could have stale references later.
                while(i-- != 0) nodePath[i] = null;
                return false;
            }

```

```

}

nodePath[i] = p;

if (dirPath[i++] = cmp > 0) {
    if (p.succ()) {
        count++;
        e = new Entry KEY_GENERIC_DIAMOND(k);

        if (p.right == null) lastEntry = e;

        e.left = p;
        e.right = p.right;

        p.right(e);

        break;
    }

    p = p.right;
}
else {
    if (p.pred()) {
        count++;
        e = new Entry KEY_GENERIC_DIAMOND(k);

        if (p.left == null) firstEntry = e;

        e.right = p;
        e.left = p.left;

        p.left(e);

        break;
    }

    p = p.left;
}
}

maxDepth = i--;

while(i > 0 && ! nodePath[i].black()) {
    if (! dirPath[i - 1]) {
        Entry KEY_GENERIC y = nodePath[i - 1].right;

        if (! nodePath[i - 1].succ() && ! y.black()) {
            nodePath[i].black(true);

```



```

}
else {
    Entry KEY_GENERIC x;

    if (dirPath[i] y = nodePath[i];
    else {
        x = nodePath[i];
        y = x.left;
        x.left = y.right;
        y.right = x;
        nodePath[i - 1].right = y;

        if (y.succ()) {
            y.succ(false);
            x.pred(y);
        }

    }

    x = nodePath[i - 1];
    x.black(false);
    y.black(true);

    x.right = y.left;
    y.left = x;
    if (i < 2) tree = y;
    else {
        if (dirPath[i - 2]) nodePath[i - 2].right = y;
        else nodePath[i - 2].left = y;
    }

    if (y.pred()){
        y.pred(false);
        x.succ(y);
    }

    break;
}
}
}
}
tree.black(true);
// We clean up the node path, or we could have stale references later.
while(maxDepth-- != 0) nodePath[maxDepth] = null;
return true;
}

```

```

SUPPRESS_WARNINGS_KEY_UNCHECKED
@Override
public boolean remove(final KEY_TYPE k) {
    if (tree == null) return false;

    Entry KEY_GENERIC p = tree;
    int cmp;
    int i = 0;
    final KEY_GENERIC_TYPE kk = KEY_GENERIC_CAST k;

    while(true) {
        if ((cmp = compare(kk, p.key)) == 0) break;

        dirPath[i] = cmp > 0;
        nodePath[i] = p;

        if (dirPath[i++] {
            if ((p = p.right()) == null) {
                // We clean up the node path, or we could have stale references later.
                while(i-- != 0) nodePath[i] = null;
                return false;
            }
        }
        else {
            if ((p = p.left()) == null) {
                // We clean up the node path, or we could have stale references later.
                while(i-- != 0) nodePath[i] = null;
                return false;
            }
        }
    }

    if (p.left == null) firstEntry = p.next();
    if (p.right == null) lastEntry = p.prev();

    if (p.succ()) {
        if (p.pred()) {
            if (i == 0) tree = p.left;
            else {
                if (dirPath[i - 1]) nodePath[i - 1].succ(p.right);
                else nodePath[i - 1].pred(p.left);
            }
        }
    }
    else {
        p.prev().right = p.right;

        if (i == 0) tree = p.left;
        else {

```

```

    if (dirPath[i - 1]) nodePath[i - 1].right = p.left;
    else nodePath[i - 1].left = p.left;
  }
}
else {
  boolean color;
  Entry KEY_GENERIC r = p.right;

  if (r.pred()) {
    r.left = p.left;
    r.pred(p.pred());
    if (! r.pred()) r.prev().right = r;
    if (i == 0) tree = r;
    else {
      if (dirPath[i - 1]) nodePath[i - 1].right = r;
      else nodePath[i - 1].left = r;
    }

    color = r.black();
    r.black(p.black());
    p.black(color);
    dirPath[i] = true;
    nodePath[i++] = r;
  }
  else {
    Entry KEY_GENERIC s;
    int j = i++;

    while(true) {
      dirPath[i] = false;
      nodePath[i++] = r;
      s = r.left;
      if (s.pred()) break;
      r = s;
    }

    dirPath[j] = true;
    nodePath[j] = s;

    if (s.succ()) r.pred(s);
    else r.left = s.right;

    s.left = p.left;

    if (! p.pred()) {
      p.prev().right = s;
      s.pred(false);

```

```

}

s.right(p.right);

color = s.black();
s.black(p.black());
p.black(color);

if (j == 0) tree = s;
else {
    if (dirPath[j - 1]) nodePath[j - 1].right = s;
    else nodePath[j - 1].left = s;
}
}
}

int maxDepth = i;

if (p.black()) {
    for(; i > 0; i--) {
        if (dirPath[i - 1] && ! nodePath[i - 1].succ() ||
            ! dirPath[i - 1] && ! nodePath[i - 1].pred()) {
            Entry KEY_GENERIC x = dirPath[i - 1] ? nodePath[i - 1].right : nodePath[i - 1].left;

            if (! x.black()) {
                x.black(true);
                break;
            }
        }

        if (! dirPath[i - 1]) {
            Entry KEY_GENERIC w = nodePath[i - 1].right;

            if (! w.black()) {
                w.black(true);
                nodePath[i - 1].black(false);

                nodePath[i - 1].right = w.left;
                w.left = nodePath[i - 1];

                if (i < 2) tree = w;
                else {
                    if (dirPath[i - 2]) nodePath[i - 2].right = w;
                    else nodePath[i - 2].left = w;
                }

                nodePath[i] = nodePath[i - 1];
                dirPath[i] = false;
            }
        }
    }
}

```

```

nodePath[i - 1] = w;
if (maxDepth == i++) maxDepth++;

w = nodePath[i - 1].right;
}

if ((w.pred() || w.left.black()) &&
    (w.succ() || w.right.black())) {
    w.black(false);
}
else {
    if (w.succ() || w.right.black()) {
        Entry KEY_GENERIC y = w.left;

        y.black(true);
        w.black(false);
        w.left = y.right;
        y.right = w;
        w = nodePath[i - 1].right = y;

        if (w.succ()) {
            w.succ(false);
            w.right.pred(w);
        }
    }

    w.black(nodePath[i - 1].black());
    nodePath[i - 1].black(true);
    w.right.black(true);

    nodePath[i - 1].right = w.left;
    w.left = nodePath[i - 1];

    if (i < 2) tree = w;
    else {
        if (dirPath[i - 2]) nodePath[i - 2].right = w;
        else nodePath[i - 2].left = w;
    }

    if (w.pred()) {
        w.pred(false);
        nodePath[i - 1].succ(w);
    }
    break;
}
else {
    Entry KEY_GENERIC w = nodePath[i - 1].left;

```

```

if (! w.black()) {
    w.black (true);
    nodePath[i - 1].black(false);

    nodePath[i - 1].left = w.right;
    w.right = nodePath[i - 1];

    if (i < 2) tree = w;
    else {
        if (dirPath[i - 2]) nodePath[i - 2].right = w;
        else nodePath[i - 2].left = w;
    }

    nodePath[i] = nodePath[i - 1];
    dirPath[i] = true;
    nodePath[i - 1] = w;
    if (maxDepth == i++) maxDepth++;

    w = nodePath[i - 1].left;
}

if ((w.pred() || w.left.black()) &&
    (w.succ() || w.right.black())) {
    w.black(false);
}
else {
    if (w.pred() || w.left.black()) {
        Entry KEY_GENERIC y = w.right;

        y.black(true);
        w.black (false);
        w.right = y.left;
        y.left = w;
        w = nodePath[i - 1].left = y;

        if (w.pred()) {
            w.pred(false);
            w.left.succ(w);
        }
    }

    w.black(nodePath[i - 1].black());
    nodePath[i - 1].black(true);
    w.left.black(true);

    nodePath[i - 1].left = w.right;
    w.right = nodePath[i - 1];
}

```

```

    if (i < 2) tree = w;
    else {
        if (dirPath[i - 2]) nodePath[i - 2].right = w;
        else nodePath[i - 2].left = w;
    }

    if (w.succ()) {
        w.succ(false);
        nodePath[i - 1].pred(w);
    }
    break;
}
}
}

if (tree != null) tree.black(true);
}

count--;
// We clean up the node path, or we could have stale references later.
while(maxDepth-- != 0) nodePath[maxDepth] = null;
return true;
}

SUPPRESS_WARNINGS_KEY_UNCHECKED
@Override
public boolean contains(final KEY_TYPE k) {
    return findKey(KEY_GENERIC_CAST k) != null;
}

#if KEY_CLASS_Object
SUPPRESS_WARNINGS_KEY_UNCHECKED
public K get(final KEY_TYPE k) {
    final Entry KEY_GENERIC entry = findKey(KEY_GENERIC_CAST k);
    return entry == null ? null : entry.key;
}
#endif

@Override
public void clear() {
    count = 0;
    tree = null;
    firstEntry = lastEntry = null;
}

/** This class represent an entry in a tree set.

```

```

*
* <p>We use the only "metadata", i.e., {@link Entry#info}, to store
* information about color, predecessor status and successor status.
*
* <p>Note that since the class is recursive, it can be
* considered equivalently a tree.
*/

private static final class Entry KEY_GENERIC implements Cloneable {
    /** The the bit in this mask is true, the node is black. */
    private static final int BLACK_MASK = 1;
    /** If the bit in this mask is true, {@link #right} points to a successor. */
    private static final int SUCC_MASK = 1 << 31;
    /** If the bit in this mask is true, {@link #left} points to a predecessor. */
    private static final int PRED_MASK = 1 << 30;
    /** The key of this entry. */
    KEY_GENERIC_TYPE key;
    /** The pointers to the left and right subtrees. */
    Entry KEY_GENERIC left, right;
    /** This integers holds different information in different bits (see {@link #SUCC_MASK}, {@link
    #PRED_MASK} and {@link #BLACK_MASK}). */
    int info;

    Entry() {}

    /** Creates a new red entry with the given key.
    *
    * @param k a key.
    */
    Entry(final KEY_GENERIC_TYPE k) {
        this.key = k;
        info = SUCC_MASK | PRED_MASK;
    }

    /** Returns the left subtree.
    *
    * @return the left subtree ({@code null} if the left
    * subtree is empty).
    */
    Entry KEY_GENERIC left() {
        return (info & PRED_MASK) != 0 ? null : left;
    }

    /** Returns the right subtree.
    *
    * @return the right subtree ({@code null} if the right
    * subtree is empty).
    */

```

```

Entry KEY_GENERIC right() {
    return (info & SUCC_MASK) != 0 ? null : right;
}

/** Checks whether the left pointer is really a predecessor.
 * @return true if the left pointer is a predecessor.
 */
boolean pred() {
    return (info & PRED_MASK) != 0;
}

/** Checks whether the right pointer is really a successor.
 * @return true if the right pointer is a successor.
 */
boolean succ() {
    return (info & SUCC_MASK) != 0;
}

/** Sets whether the left pointer is really a predecessor.
 * @param pred if true then the left pointer will be considered a predecessor.
 */
void pred(final boolean pred) {
    if (pred) info |= PRED_MASK;
    else info &= ~PRED_MASK;
}

/** Sets whether the right pointer is really a successor.
 * @param succ if true then the right pointer will be considered a successor.
 */
void succ(final boolean succ) {
    if (succ) info |= SUCC_MASK;
    else info &= ~SUCC_MASK;
}

/** Sets the left pointer to a predecessor.
 * @param pred the predecessor.
 */
void pred(final Entry KEY_GENERIC pred) {
    info |= PRED_MASK;
    left = pred;
}

/** Sets the right pointer to a successor.
 * @param succ the successor.
 */
void succ(final Entry KEY_GENERIC succ) {
    info |= SUCC_MASK;
    right = succ;
}

```

```

}

/** Sets the left pointer to the given subtree.
 * @param left the new left subtree.
 */
void left(final Entry KEY_GENERIC left) {
    info &= ~PRED_MASK;
    this.left = left;
}

/** Sets the right pointer to the given subtree.
 * @param right the new right subtree.
 */
void right(final Entry KEY_GENERIC right) {
    info &= ~SUCC_MASK;
    this.right = right;
}

/** Returns whether this node is black.
 * @return true iff this node is black.
 */
boolean black() {
    return (info & BLACK_MASK) != 0;
}

/** Sets whether this node is black.
 * @param black if true, then this node becomes black; otherwise, it becomes red..
 */
void black(final boolean black) {
    if (black) info |= BLACK_MASK;
    else info &= ~BLACK_MASK;
}

/** Computes the next entry in the set order.
 *
 * @return the next entry ({ @code null}) if this is the last entry).
 */
Entry KEY_GENERIC next() {
    Entry KEY_GENERIC next = this.right;
    if ((info & SUCC_MASK) == 0) while ((next.info & PRED_MASK) == 0) next = next.left;
    return next;
}

/** Computes the previous entry in the set order.
 *
 * @return the previous entry ({ @code null}) if this is the first entry).
 */

```

```

Entry KEY_GENERIC prev() {
    Entry KEY_GENERIC prev = this.left;
    if ((info & PRED_MASK) == 0) while ((prev.info & SUCC_MASK) == 0) prev = prev.right;
    return prev;
}

@Override
SUPPRESS_WARNINGS_KEY_UNCHECKED
public Entry KEY_GENERIC clone() {
    Entry KEY_GENERIC c;
    try {
        c = (Entry KEY_GENERIC)super.clone();
    }
    catch(CloneNotSupportedException cantHappen) {
        throw new InternalError();
    }

    c.key = key;
    c.info = info;

    return c;
}

public boolean equals(final Object o) {
    if (!(o instanceof Entry)) return false;
    Entry KEY_GENERIC_WILDCARD e = (Entry KEY_GENERIC_WILDCARD)o;

    return KEY_EQUALS(key, e.key);
}

public int hashCode() {
    return KEY2JAVAHASH_NOT_NULL(key);
}

public String toString() {
    return String.valueOf(key);
}

/*
    public void prettyPrint() {
        prettyPrint(0);
    }

    public void prettyPrint(int level) {
        if (pred()) {

```

```

    for (int i = 0; i < level; i++)
        System.err.print(" ");
    System.err.println("pred: " + left);
    }
    else if (left != null)
        left.prettyPrint(level + 1);
    for (int i = 0; i < level; i++)
        System.err.print(" ");
    System.err.println(key + " (" + (black() ? "black" : "red") + ")");
    if (succ()) {
        for (int i = 0; i < level; i++)
            System.err.print(" ");
        System.err.println("succ: " + right);
    }
    else if (right != null)
        right.prettyPrint(level + 1);
    }*/
}

/*
    public void prettyPrint() {
        System.err.println("size: " + count);
        if (tree != null) tree.prettyPrint();
    }
    */

@Override
public int size() {
    return count;
}

@Override
public boolean isEmpty() {
    return count == 0;
}

@Override
public KEY_GENERIC_TYPE FIRST() {
    if (tree == null) throw new NoSuchElementException();
    return firstEntry.key;
}

@Override
public KEY_GENERIC_TYPE LAST() {
    if (tree == null) throw new NoSuchElementException();
    return lastEntry.key;
}

```

```

/** An iterator on the whole range.
 *
 * <p>This class can iterate in both directions on a threaded tree.
 */

private class SetIterator implements KEY_LIST_ITERATOR KEY_GENERIC {
    /** The entry that will be returned by the next call to {@link java.util.ListIterator#previous()} (or {@code null} if
    no previous entry exists). */
    Entry KEY_GENERIC prev;
    /** The entry that will be returned by the next call to {@link java.util.ListIterator#next()} (or {@code null} if no
    next entry exists). */
    Entry KEY_GENERIC next;
    /** The last entry that was returned (or {@code null} if we did not iterate or used {@link #remove()}). */
    Entry KEY_GENERIC curr;
    /** The current index (in the sense of a {@link java.util.ListIterator}). Note that this value is not meaningful when
    this iterator has been created using the nonempty constructor.*/
    int index = 0;

    SetIterator() {
        next = firstEntry;
    }

    SetIterator(final KEY_GENERIC_TYPE k) {
        if ((next = locateKey(k)) != null) {
            if (compare(next.key, k) <= 0) {
                prev = next;
                next = next.next();
            }
            else prev = next.prev();
        }
    }

    @Override
    public boolean hasNext() { return next != null; }
    @Override
    public boolean hasPrevious() { return prev != null; }

    void updateNext() { next = next.next(); }
    void updatePrevious() { prev = prev.prev(); }

    @Override
    public KEY_GENERIC_TYPE NEXT_KEY() { return nextEntry().key; }
    @Override
    public KEY_GENERIC_TYPE PREV_KEY() { return previousEntry().key; }

    Entry KEY_GENERIC nextEntry() {
        if (! hasNext()) throw new NoSuchElementException();
    }

```

```

curr = prev = next;
index++;
updateNext();
return curr;
}

```

```

Entry KEY_GENERIC previousEntry() {
if (! hasPrevious()) throw new NoSuchElementException();
curr = next = prev;
index--;
updatePrevious();
return curr;
}

```

```

@Override
public int nextIndex() { return index; }
@Override
public int previousIndex() { return index - 1; }

```

```

@Override
public void remove() {
if (curr == null) throw new IllegalStateException();
/* If the last operation was a next(), we are removing an entry that precedes
the current index, and thus we must decrement it. */
if (curr == prev) index--;
next = prev = curr;
updatePrevious();
updateNext();
RB_TREE_SET.this.remove(curr.key);
curr = null;
}
}

```

```

@Override
public KEY_BIDI_ITERATOR KEY_GENERIC iterator() { return new SetIterator(); }

```

```

@Override
public KEY_BIDI_ITERATOR KEY_GENERIC iterator(final KEY_GENERIC_TYPE from) { return new
SetIterator(from); }

```

```

@Override
public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() { return actualComparator; }

```

```

@Override
public SORTED_SET KEY_GENERIC headSet(final KEY_GENERIC_TYPE to) { return new
Subset(KEY_NULL, true, to, false); }

```

```

@Override

```

```

public SORTED_SET KEY_GENERIC tailSet(final KEY_GENERIC_TYPE from) { return new Subset(from,
false, KEY_NULL, true); }

@Override
public SORTED_SET KEY_GENERIC subSet(final KEY_GENERIC_TYPE from, final KEY_GENERIC_TYPE
to) { return new Subset(from, false, to, false); }

/** A subset with given range.
 *
 * <p>This class represents a subset. One has to specify the left/right
 * limits (which can be set to -&infin; or &infin;). Since the subset is a
 * view on the set, at a given moment it could happen that the limits of
 * the range are not any longer in the main set. Thus, things such as
 * { @link java.util.SortedSet#first()} or { @link java.util.Collection#size()} must be always computed
 * on-the-fly.
 */
private final class Subset extends ABSTRACT_SORTED_SET KEY_GENERIC implements java.io.Serializable,
SORTED_SET KEY_GENERIC {
    private static final long serialVersionUID = -7046029254386353129L;

    /** The start of the subset range, unless { @link #bottom} is true. */
    KEY_GENERIC_TYPE from;
    /** The end of the subset range, unless { @link #top} is true. */
    KEY_GENERIC_TYPE to;
    /** If true, the subset range starts from -&infin;. */
    boolean bottom;
    /** If true, the subset range goes to &infin;. */
    boolean top;

    /** Creates a new subset with given key range.
     *
     * @param from the start of the subset range.
     * @param bottom if true, the first parameter is ignored and the range starts from -&infin;.
     * @param to the end of the subset range.
     * @param top if true, the third parameter is ignored and the range goes to &infin;.
     */
    public Subset(final KEY_GENERIC_TYPE from, final boolean bottom, final KEY_GENERIC_TYPE to, final
boolean top) {
        if (! bottom && ! top && RB_TREE_SET.this.compare(from, to) > 0) throw new IllegalArgumentException("Start
element (" + from + ") is larger than end element (" + to + ")");

        this.from = from;
        this.bottom = bottom;
        this.to = to;
        this.top = top;
    }

    @Override

```

```

public void clear() {
    final SubsetIterator i = new SubsetIterator();
    while(i.hasNext()) {
        i.NEXT_KEY();
        i.remove();
    }
}

/** Checks whether a key is in the subset range.
 * @param k a key.
 * @return true if is the key is in the subset range.
 */
final boolean in(final KEY_GENERIC_TYPE k) {
    return (bottom || RB_TREE_SET.this.compare(k, from) >= 0) &&
        (top || RB_TREE_SET.this.compare(k, to) < 0);
}

@Override
SUPPRESS_WARNINGS_KEY_UNCHECKED
public boolean contains(final KEY_TYPE k) {
    return in(KEY_GENERIC_CAST k) && RB_TREE_SET.this.contains(k);
}

@Override
public boolean add(final KEY_GENERIC_TYPE k) {
    if (! in(k)) throw new IllegalArgumentException("Element (" + k + ") out of range [" + (bottom ? "-" :
String.valueOf(from)) + ", " + (top ? "-" : String.valueOf(to)) + ")");
    return RB_TREE_SET.this.add(k);
}

@Override
SUPPRESS_WARNINGS_KEY_UNCHECKED
public boolean remove(final KEY_TYPE k) {
    if (! in(KEY_GENERIC_CAST k)) return false;
    return RB_TREE_SET.this.remove(k);
}

@Override
public int size() {
    final SubsetIterator i = new SubsetIterator();
    int n = 0;

    while(i.hasNext()) {
        n++;
        i.NEXT_KEY();
    }

    return n;
}

```

```

}

@Override
public boolean isEmpty() {
    return ! new SubsetIterator().hasNext();
}

@Override
public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() {
    return actualComparator;
}

@Override
public KEY_BIDI_ITERATOR KEY_GENERIC iterator() {
    return new SubsetIterator();
}

@Override
public KEY_BIDI_ITERATOR KEY_GENERIC iterator(final KEY_GENERIC_TYPE from) {
    return new SubsetIterator(from);
}

@Override
public SORTED_SET KEY_GENERIC headSet(final KEY_GENERIC_TYPE to) {
    if (top) return new Subset(from, bottom, to, false);
    return compare(to, this.to) < 0 ? new Subset(from, bottom, to, false) : this;
}

@Override
public SORTED_SET KEY_GENERIC tailSet(final KEY_GENERIC_TYPE from) {
    if (bottom) return new Subset(from, false, to, top);
    return compare(from, this.from) > 0 ? new Subset(from, false, to, top) : this;
}

@Override
public SORTED_SET KEY_GENERIC subSet(KEY_GENERIC_TYPE from, KEY_GENERIC_TYPE to) {
    if (top && bottom) return new Subset(from, false, to, false);
    if (! top) to = compare(to, this.to) < 0 ? to : this.to;
    if (! bottom) from = compare(from, this.from) > 0 ? from : this.from;
    if (! top && ! bottom && from == this.from && to == this.to) return this;
    return new Subset(from, false, to, false);
}

/** Locates the first entry.
 *
 * @return the first entry of this subset, or { @code null } if the subset is empty.

```

```

*/
public RB_TREE_SET.Entry KEY_GENERIC firstEntry() {
    if (tree == null) return null;
    // If this subset goes to -infinity, we return the main set first entry; otherwise, we locate the start of the set.
    RB_TREE_SET.Entry KEY_GENERIC e;
    if (bottom) e = firstEntry;
    else {
        e = locateKey(from);
        // If we find either the start or something greater we're OK.
        if (compare(e.key, from) < 0) e = e.next();
    }
    // Finally, if this subset doesn't go to infinity, we check that the resulting key isn't greater than the end.
    if (e == null || ! top && compare(e.key, to) >= 0) return null;
    return e;
}

```

/** Locates the last entry.

*

* @return the last entry of this subset, or { @code null } if the subset is empty.

*/

```

public RB_TREE_SET.Entry KEY_GENERIC lastEntry() {
    if (tree == null) return null;
    // If this subset goes to infinity, we return the main set last entry; otherwise, we locate the end of the set.
    RB_TREE_SET.Entry KEY_GENERIC e;
    if (top) e = lastEntry;
    else {
        e = locateKey(to);
        // If we find something smaller than the end we're OK.
        if (compare(e.key, to) >= 0) e = e.prev();
    }
    // Finally, if this subset doesn't go to -infinity, we check that the resulting key isn't smaller than the start.
    if (e == null || ! bottom && compare(e.key, from) < 0) return null;
    return e;
}

```

@Override

```

public KEY_GENERIC_TYPE FIRST() {
    RB_TREE_SET.Entry KEY_GENERIC e = firstEntry();
    if (e == null) throw new NoSuchElementException();
    return e.key;
}

```

@Override

```

public KEY_GENERIC_TYPE LAST() {
    RB_TREE_SET.Entry KEY_GENERIC e = lastEntry();
    if (e == null) throw new NoSuchElementException();
    return e.key;
}

```

```

}

/** An iterator for subranges.
 *
 * <p>This class inherits from { @link SetIterator}, but overrides the methods that
 * update the pointer after a { @link java.util.ListIterator#next()} or { @link java.util.ListIterator#previous()}. If we
 would
 * move out of the range of the subset we just overwrite the next or previous
 * entry with { @code null}.
 */
private final class SubsetIterator extends SetIterator {
SubsetIterator() {
next = firstEntry();
}

SubsetIterator(final KEY_GENERIC_TYPE k) {
this();

if (next != null) {
if (! bottom && compare(k, next.key) < 0) prev = null;
else if (! top && compare(k, (prev = lastEntry()).key) >= 0) next = null;
else {
next = locateKey(k);

if (compare(next.key, k) <= 0) {
prev = next;
next = next.next();
}
else prev = next.prev();
}
}
}

@Override
void updatePrevious() {
prev = prev.prev();
if (! bottom && prev != null && RB_TREE_SET.this.compare(prev.key, from) < 0) prev = null;
}

@Override
void updateNext() {
next = next.next();
if (! top && next != null && RB_TREE_SET.this.compare(next.key, to) >= 0) next = null;
}
}
}

```

```

/** Returns a deep copy of this tree set.
 *
 * <p>This method performs a deep copy of this tree set; the data stored in the
 * set, however, is not cloned. Note that this makes a difference only for object keys.
 *
 * @return a deep copy of this tree set.
 */
@Override
SUPPRESS_WARNINGS_KEY_UNCHECKED
public Object clone() {
    RB_TREE_SET KEY_GENERIC c;
    try {
        c = (RB_TREE_SET KEY_GENERIC)super.clone();
    }
    catch(CloneNotSupportedException cantHappen) {
        throw new InternalError();
    }

    c.allocatePaths();

    if (count != 0) {
        // Also this apparently unfathomable code is derived from GNU libavl.
        Entry KEY_GENERIC e, p, q, rp = new Entry KEY_GENERIC_DIAMOND(), rq = new Entry
        KEY_GENERIC_DIAMOND();

        p = rp;
        rp.left(tree);

        q = rq;
        rq.pred(null);

        while(true) {
            if (! p.pred()) {
                e = p.left.clone();
                e.pred(q.left);
                e.succ(q);
                q.left(e);

                p = p.left;
                q = q.left;
            }
            else {
                while(p.succ()) {
                    p = p.right;
                }

                if (p == null) {
                    q.right = null;
                }
            }
        }
    }
}

```

```

    c.tree = rq.left;

    c.firstEntry = c.tree;
    while(c.firstEntry.left != null) c.firstEntry = c.firstEntry.left;
    c.lastEntry = c.tree;
    while(c.lastEntry.right != null) c.lastEntry = c.lastEntry.right;

    return c;
}
q = q.right;
}

p = p.right;
q = q.right;
}

if (! p.succ()) {
    e = p.right.clone();
    e.succ(q.right);
    e.pred(q);
    q.right(e);
}
}
}

return c;
}

private void writeObject(java.io.ObjectOutputStream s) throws java.io.IOException {
    int n = count;
    SetIterator i = new SetIterator();

    s.defaultWriteObject();
    while(n-- != 0) s.WRITE_KEY(i.NEXT_KEY());
}

/** Reads the given number of entries from the input stream, returning the corresponding tree.
 *
 * @param s the input stream.
 * @param n the (positive) number of entries to read.
 * @param pred the entry containing the key that precedes the first key in the tree.
 * @param succ the entry containing the key that follows the last key in the tree.
 */
SUPPRESS_WARNINGS_KEY_UNCHECKED
private Entry KEY_GENERIC readTree(final java.io.ObjectInputStream s, final int n, final Entry KEY_GENERIC
pred, final Entry KEY_GENERIC succ) throws java.io.IOException, ClassNotFoundException {

```

```

if (n == 1) {
    final Entry KEY_GENERIC top = new Entry KEY_GENERIC_DIAMOND(KEY_GENERIC_CAST
s.READ_KEY());
    top.pred(pred);
    top.succ(succ);
    top.black(true);

    return top;
}

if (n == 2) {
    /* We handle separately this case so that recursion will
    *always* be on nonempty subtrees. */
    final Entry KEY_GENERIC top = new Entry KEY_GENERIC_DIAMOND(KEY_GENERIC_CAST
s.READ_KEY());
    top.black(true);
    top.right(new Entry KEY_GENERIC_DIAMOND(KEY_GENERIC_CAST s.READ_KEY()));
    top.right.pred(top);
    top.pred(pred);
    top.right.succ(succ);

    return top;
}

// The right subtree is the largest one.
final int rightN = n / 2, leftN = n - rightN - 1;

final Entry KEY_GENERIC top = new Entry KEY_GENERIC_DIAMOND();

top.left(readTree(s, leftN, pred, top));

top.key = KEY_GENERIC_CAST s.READ_KEY();
top.black(true);

top.right(readTree(s, rightN, top, succ));

if (n + 2 == ((n + 2) & -(n + 2))) top.right.black(false); // Quick test for determining whether n + 2 is a power of 2.

return top;
}

private void readObject(java.io.ObjectInputStream s) throws java.io.IOException, ClassNotFoundException {
    s.defaultReadObject();
    /* The storedComparator is now correctly set, but we must restore
    on-the-fly the actualComparator. */
    setActualComparator();
    allocatePaths();
}

```

```

if (count != 0) {
    tree = readTree(s, count, null, null);
    Entry KEY_GENERIC e;

    e = tree;
    while(e.left() != null) e = e.left();
    firstEntry = e;

    e = tree;
    while(e.right() != null) e = e.right();
    lastEntry = e;
}
}

#ifdef ASSERTS_CODE
private void checkNodePath() {
    for(int i = nodePath.length; i-- != 0;) assert nodePath[i] == null : i;
}

private static KEY_GENERIC int checkTree(Entry KEY_GENERIC e, int d, int D) {
    if (e == null) return 0;
    if (e.black()) d++;
    if (e.left() != null) D = checkTree(e.left(), d, D);
    if (e.right() != null) D = checkTree(e.right(), d, D);
    if (e.left() == null && e.right() == null) {
        if (D == -1) D = d;
        else if (D != d) throw new AssertionError("Mismatch between number of black nodes (" + D + " and " + d + ")");
    }
    return D;
}
#endif

#ifdef TEST

private static long seed = System.currentTimeMillis();
private static java.util.Random r = new java.util.Random(seed);

private static KEY_TYPE genKey() {
#ifdef KEY_CLASS_Byte || KEY_CLASS_Short || KEY_CLASS_Character
    return (KEY_TYPE)(r.nextInt());
#elif KEYS_PRIMITIVE
    return r.NEXT_KEY();
#else
    return Integer.toBinaryString(r.nextInt());
#endif
}
}

```

```

private static java.text.NumberFormat format = new java.text.DecimalFormat("#,###.00");
private static java.text.FieldPosition p = new java.text.FieldPosition(0);

private static String format(double d) {
    StringBuffer s = new StringBuffer();
    return format.format(d, s, p).toString();
}

private static void speedTest(int n, boolean comp) {
    int i, j;
    RB_TREE_SET m;
    java.util.TreeSet t;
    KEY_TYPE k[] = new KEY_TYPE[n];
    KEY_TYPE nk[] = new KEY_TYPE[n];
    long ms;

    for(i = 0; i < n; i++) {
        k[i] = genKey();
        nk[i] = genKey();
    }

    double totAdd = 0, totYes = 0, totNo = 0, totIterFor = 0, totIterBack = 0, totRemYes = 0, d, dd;

    if (comp) {
        for(j = 0; j < 20; j++) {

            t = new java.util.TreeSet();

            /* We first add all pairs to t. */
            for(i = 0; i < n; i++) t.add(KEY2OBJ(k[i]));

            /* Then we remove the first half and put it back. */
            for(i = 0; i < n/2; i++) t.remove(KEY2OBJ(k[i]));

            ms = System.currentTimeMillis();
            for(i = 0; i < n/2; i++) t.add(KEY2OBJ(k[i]));
            d = System.currentTimeMillis() - ms;

            /* Then we remove the other half and put it back again. */
            ms = System.currentTimeMillis();
            for(i = n/2; i < n; i++) t.remove(KEY2OBJ(k[i]));
            dd = System.currentTimeMillis() - ms ;

            ms = System.currentTimeMillis();
            for(i = n/2; i < n; i++) t.add(KEY2OBJ(k[i]));
            d += System.currentTimeMillis() - ms;
            if (j > 2) totAdd += n/d;
        }
    }
}

```

```

System.out.print("Add: " + format(n/d) + " K/s ");

/* Then we remove again the first half. */
ms = System.currentTimeMillis();
for(i = 0; i < n/2; i++) t.remove(KEY2OBJ(k[i]));
dd += System.currentTimeMillis() - ms ;
if (j > 2) totRemYes += n/dd;
System.out.print("RemYes: " + format(n/dd) + " K/s ");

/* And then we put it back. */
for(i = 0; i < n/2; i++) t.add(KEY2OBJ(k[i]));

/* We check for pairs in t. */
ms = System.currentTimeMillis();
for(i = 0; i < n; i++) t.contains(KEY2OBJ(k[i]));
d = 1.0 * n / (System.currentTimeMillis() - ms);
if (j > 2) totYes += d;
System.out.print("Yes: " + format(d) + " K/s ");

/* We check for pairs not in t. */
ms = System.currentTimeMillis();
for(i = 0; i < n; i++) t.contains(KEY2OBJ(nk[i]));
d = 1.0 * n / (System.currentTimeMillis() - ms);
if (j > 2) totNo += d;
System.out.print("No: " + format(d) + " K/s ");

/* We iterate on t. */
ms = System.currentTimeMillis();
for(Iterator it = t.iterator(); it.hasNext(); it.next());
d = 1.0 * n / (System.currentTimeMillis() - ms);
if (j > 2) totIterFor += d;
System.out.print("IterFor: " + format(d) + " K/s ");

System.out.println();
}

System.out.println();
System.out.println("java.util Add: " + format(totAdd/(j-3)) + " K/s RemYes: " + format(totRemYes/(j-3)) + " K/s
Yes: " + format(totYes/(j-3)) + " K/s No: " + format(totNo/(j-3)) + " K/s IterFor: " + format(totIterFor/(j-3)) + "
K/s");

System.out.println();

totAdd = totYes = totNo = totIterFor = totIterBack = totRemYes = 0;

}

for(j = 0; j < 20; j++) {

```

```

m = new RB_TREE_SET();

/* We first add all pairs to m. */
for(i = 0; i < n; i++) m.add(k[i]);

/* Then we remove the first half and put it back. */
for(i = 0; i < n/2; i++) m.remove(k[i]);

ms = System.currentTimeMillis();
for(i = 0; i < n/2; i++) m.add(k[i]);
d = System.currentTimeMillis() - ms;

/* Then we remove the other half and put it back again. */
ms = System.currentTimeMillis();
for(i = n/2; i < n; i++) m.remove(k[i]);
dd = System.currentTimeMillis() - ms ;

ms = System.currentTimeMillis();
for(i = n/2; i < n; i++) m.add(k[i]);
d += System.currentTimeMillis() - ms;
if (j > 2) totAdd += n/d;
System.out.print("Add: " + format(n/d) + " K/s ");

/* Then we remove again the first half. */
ms = System.currentTimeMillis();
for(i = 0; i < n/2; i++) m.remove(k[i]);
dd += System.currentTimeMillis() - ms ;
if (j > 2) totRemYes += n/dd;
System.out.print("RemYes: " + format(n/dd) + " K/s ");

/* And then we put it back. */
for(i = 0; i < n/2; i++) m.add(k[i]);

/* We check for pairs in m. */
ms = System.currentTimeMillis();
for(i = 0; i < n; i++) m.contains(k[i]);
d = 1.0 * n / (System.currentTimeMillis() - ms);
if (j > 2) totYes += d;
System.out.print("Yes: " + format(d) + " K/s ");

/* We check for pairs not in m. */
ms = System.currentTimeMillis();
for(i = 0; i < n; i++) m.contains(nk[i]);
d = 1.0 * n / (System.currentTimeMillis() - ms);
if (j > 2) totNo += d;
System.out.print("No: " + format(d) + " K/s ");

```

```

/* We iterate on m. */
KEY_LIST_ITERATOR it = (KEY_LIST_ITERATOR)m.iterator();
ms = System.currentTimeMillis();
for(; it.hasNext(); it.NEXT_KEY());
d = 1.0 * n / (System.currentTimeMillis() - ms);
if (j > 2) totIterFor += d;
System.out.print("IterFor: " + format(d) + " K/s ");

/* We iterate back on m. */
ms = System.currentTimeMillis();
for(; it.hasPrevious(); it.PREV_KEY());
d = 1.0 * n / (System.currentTimeMillis() - ms);
if (j > 2) totIterBack += d;
System.out.print("IterBack: " + format(d) + " K/s ");

System.out.println();
}

System.out.println();
System.out.println("fastutil Add: " + format(totAdd/(j-3)) + " K/s RemYes: " + format(totRemYes/(j-3)) + " K/s
Yes: " + format(totYes/(j-3)) + " K/s No: " + format(totNo/(j-3)) + " K/s IterFor: " + format(totIterFor/(j-3)) + " K/s
IterBack: " + format(totIterBack/(j-3)) + "K/s");

System.out.println();
}

private static boolean valEquals(Object o1, Object o2) {
    return o1 == null ? o2 == null : o1.equals(o2);
}

private static void fatal(String msg) {
    System.out.println(msg);
    System.exit(1);
}

private static void ensure(boolean cond, String msg) {
    if (cond) return;
    fatal(msg);
}

private static Object[] k, v, nk;
private static KEY_TYPE kt[];
private static KEY_TYPE nkt[];
private static RB_TREE_SET topSet;

```

```

protected static void testSets(SORTED_SET m, SortedSet t, int n, int level) {
    long ms;
    boolean mThrowsIllegal, tThrowsIllegal, mThrowsNoElement, tThrowsNoElement;
    boolean rt = false, rm = false;

    if (level > 4) return;

    /* Now we check that both sets agree on first/last keys. */

    mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

    try {
        m.first();
    }
    catch (NoSuchElementException e) { mThrowsNoElement = true; }
    try {
        t.first();
    }
    catch (NoSuchElementException e) { tThrowsNoElement = true; }

    ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): first() divergence at start in
    NoSuchElementException (" + mThrowsNoElement + ", " + tThrowsNoElement + "));
    if (! mThrowsNoElement) ensure(t.first().equals(m.first()), "Error (" + level + ", " + seed + "): m and t differ at start
    on their first key (" + m.first() + ", " + t.first() + "));

    mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

    try {
        m.last();
    }
    catch (NoSuchElementException e) { mThrowsNoElement = true; }
    try {
        t.last();
    }
    catch (NoSuchElementException e) { tThrowsNoElement = true; }

    ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): last() divergence at start in
    NoSuchElementException (" + mThrowsNoElement + ", " + tThrowsNoElement + "));

    if (! mThrowsNoElement) ensure(t.last().equals(m.last()), "Error (" + level + ", " + seed + "): m and t differ at start
    on their last key (" + m.last() + ", " + t.last() + "));

    /* Now we check that m and t are equal. */
    if (!m.equals(t) || !t.equals(m)) System.err.println("m: " + m + " t: " + t);

```

```
ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) at start");
ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) at start");
```

```
/* Now we check that m actually holds that data. */
```

```
for(Iterator i=t.iterator(); i.hasNext();) {
    ensure(m.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after insertion (iterating on t)");
}
```

```
/* Now we check that m actually holds that data, but iterating on m. */
```

```
for(Iterator i=m.iterator(); i.hasNext();) {
    ensure(t.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after insertion (iterating on m)");
}
```

```
/* Now we check that inquiries about random data give the same answer in m and t. For
m we use the polymorphic method. */
```

```
for(int i=0; i<n; i++) {
    KEY_TYPE T = genKey();
```

```
mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;
```

```
try {
    m.contains(T);
}
catch (NoSuchElementException e) { mThrowsNoElement = true; }
catch (IllegalArgumentException e) { mThrowsIllegal = true; }
```

```
try {
    t.contains(KEY2OBJ(T));
}
catch (NoSuchElementException e) { tThrowsNoElement = true; }
catch (IllegalArgumentException e) { tThrowsIllegal = true; }
```

```
ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): contains() divergence in
NoSuchElementException (" + mThrowsNoElement + ", " + tThrowsNoElement + "));
ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): contains() divergence in
IllegalArgumentException (" + mThrowsIllegal + ", " + tThrowsIllegal + "));
if (!mThrowsNoElement && !mThrowsIllegal) ensure(m.contains(KEY2OBJ(T)) == t.contains(KEY2OBJ(T)),
"Error (" + level + ", " + seed + "): divergence in keys between t and m (polymorphic method)");
}
```

```
/* Again, we check that inquiries about random data give the same answer in m and t, but
for m we use the standard method. */
```

```

for(int i=0; i<n; i++) {
    KEY_TYPE T = genKey();

    mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

    try {
        m.contains(KEY2OBJ(T));
    }
    catch (NoSuchElementException e) { mThrowsNoElement = true; }
    catch (IllegalArgumentException e) { mThrowsIllegal = true; }

    try {
        t.contains(KEY2OBJ(T));
    }
    catch (NoSuchElementException e) { tThrowsNoElement = true; }
    catch (IllegalArgumentException e) { tThrowsIllegal = true; }

    ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): contains() divergence in
    NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");
    ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): contains() divergence in
    IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ")");
    if (!mThrowsNoElement && !mThrowsIllegal) ensure(m.contains(KEY2OBJ(T)) == t.contains(KEY2OBJ(T)),
    "Error (" + level + ", " + seed + "): divergence between t and m (standard method)");
}

/* Now we add and remove random data in m and t, checking that the result is the same. */

for(int i=0; i<20*n; i++) {
    KEY_TYPE T = genKey();

    mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

    try {
        rm = m.add(KEY2OBJ(T));
    }
    catch (NoSuchElementException e) { mThrowsNoElement = true; }
    catch (IllegalArgumentException e) { mThrowsIllegal = true; }

    try {
        rt = t.add(KEY2OBJ(T));
    }
    catch (NoSuchElementException e) { tThrowsNoElement = true; }
    catch (IllegalArgumentException e) { tThrowsIllegal = true; }

    ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): add() divergence in
    NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");
    ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): add() divergence in

```

```

IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ");
    if (!mThrowsNoElement && !mThrowsIllegal) ensure(rm == rt, "Error (" + level + ", " + seed + "): divergence in
add() between t and m");

    T = genKey();

    mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

    try {
        rm = m.remove(KEY2OBJ(T));
    }
    catch (NoSuchElementException e) { mThrowsNoElement = true; }
    catch (IllegalArgumentException e) { mThrowsIllegal = true; }

    try {
        rt = t.remove(KEY2OBJ(T));
    }
    catch (NoSuchElementException e) { tThrowsNoElement = true; }
    catch (IllegalArgumentException e) { tThrowsIllegal = true; }

    ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): remove() divergence in
NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ");
    ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): remove() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ");
    if (!mThrowsNoElement && !mThrowsIllegal) ensure(rm == rt, "Error (" + level + ", " + seed + "): divergence in
remove() between t and m");
    }

    ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after removal");
    ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after removal");

    /* Now we check that m actually holds the same data. */

    for(Iterator i=t.iterator(); i.hasNext(); ) {
        ensure(m.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after removal (iterating on
t)");
    }

    /* Now we check that m actually holds that data, but iterating on m. */

    for(Iterator i=m.iterator(); i.hasNext(); ) {
        ensure(t.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after removal (iterating on
m)");
    }

    /* Now we check that both sets agree on first/last keys. */

```

```
mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;
```

```
try {  
    m.first();  
}  
catch (NoSuchElementException e) { mThrowsNoElement = true; }  
try {  
    t.first();  
}  
catch (NoSuchElementException e) { tThrowsNoElement = true; }
```

```
ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): first() divergence in  
NoSuchElementException (" + mThrowsNoElement + ", " + tThrowsNoElement + "));  
if (! mThrowsNoElement) ensure(t.first().equals(m.first()), "Error (" + level + ", " + seed + "): m and t differ on their  
first key (" + m.first() + ", " + t.first() + "));
```

```
mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;
```

```
try {  
    m.last();  
}  
catch (NoSuchElementException e) { mThrowsNoElement = true; }  
try {  
    t.last();  
}  
catch (NoSuchElementException e) { tThrowsNoElement = true; }
```

```
ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): last() divergence in  
NoSuchElementException (" + mThrowsNoElement + ", " + tThrowsNoElement + "));
```

```
if (! mThrowsNoElement) ensure(t.last().equals(m.last()), "Error (" + level + ", " + seed + "): m and t differ on their  
last key (" + m.last() + ", " + t.last() + "));
```

```
/* Now we check cloning. */
```

```
if (level == 0) {  
    ensure(m.equals(((RB_TREE_SET)m).clone()), "Error (" + level + ", " + seed + "): m does not equal m.clone()");  
    ensure(((RB_TREE_SET)m).clone().equals(m), "Error (" + level + ", " + seed + "): m.clone() does not equal m");  
    m = (RB_TREE_SET)((RB_TREE_SET)m).clone();  
}
```

```
/* Now we play with constructors. */
```

```
ensure(m.equals(new RB_TREE_SET((Collection)m)), "Error (" + level + ", " + seed + "): m does not equal new  
(Collection m)");  
ensure((new RB_TREE_SET((Collection)m)).equals(m), "Error (" + level + ", " + seed + "): new (Collection  
m) does not equal m");  
ensure(m.equals(new RB_TREE_SET((COLLECTION)m)), "Error (" + level + ", " + seed + "): m does not equal  
new (type-specific Collection m)");
```

```

    ensure((new RB_TREE_SET((COLLECTION)m)).equals(m), "Error (" + level + ", " + seed + "): new (type-specific
Collection m) does not equal m");
    ensure(m.equals(new RB_TREE_SET((SortedSet)m)), "Error (" + level + ", " + seed + "): m does not equal new
(SortedSet m)");
    ensure((new RB_TREE_SET((SortedSet)m)).equals(m), "Error (" + level + ", " + seed + "): new (SortedSet m) does
not equal m");
    ensure(m.equals(new RB_TREE_SET((SORTED_SET)m)), "Error (" + level + ", " + seed + "): m does not equal
new (type-specific SortedSet m)");
    ensure((new RB_TREE_SET((SORTED_SET)m)).equals(m), "Error (" + level + ", " + seed + "): new (type-specific
SortedSet m) does not equal m");
    ensure(m.equals(new RB_TREE_SET(m.iterator())), "Error (" + level + ", " + seed + "): m does not equal new
(m.iterator())");
    ensure((new RB_TREE_SET(m.iterator())).equals(m), "Error (" + level + ", " + seed + "): new (m.iterator()) does
not equal m");
    ensure(m.equals(new RB_TREE_SET(m.iterator())), "Error (" + level + ", " + seed + "): m does not equal new
(m.type_specific_iterator())");
    ensure((new RB_TREE_SET(m.iterator())).equals(m), "Error (" + level + ", " + seed + "): new
(m.type_specific_iterator()) does not equal m");

```

```

/* Now we play with conversion to array, wrapping and copying. */

```

```

    ensure(m.equals(new RB_TREE_SET(m.TO_KEY_ARRAY())), "Error (" + level + ", " + seed + "): m does not
equal new (toArray(m))");
    ensure((new RB_TREE_SET(m.TO_KEY_ARRAY())).equals(m), "Error (" + level + ", " + seed + "): new
(toArray(m)) does not equal m");

```

```

int h = m.hashCode();

```

```

/* Now we save and read m. */

```

```

SORTED_SET m2 = null;

```

```

try {
    java.io.File ff = new java.io.File("it.unimi.dsi.fastutil.test");
    java.io.OutputStream os = new java.io.FileOutputStream(ff);
    java.io.ObjectOutputStream oos = new java.io.ObjectOutputStream(os);

```

```

    oos.writeObject(m);
    oos.close();

```

```

    java.io.InputStream is = new java.io.FileInputStream(ff);
    java.io.ObjectInputStream ois = new java.io.ObjectInputStream(is);

```

```

    m2 = (SORTED_SET)ois.readObject();
    ois.close();
    ff.delete();

```

```

}
catch(Exception e) {
    e.printStackTrace();
    System.exit(1);
}

ensure(m2.hashCode() == h, "Error (" + level + ", " + seed + "): hashCode() changed after save/read");

/* Now we check that m2 actually holds that data. */

ensure(m2.equals(t), "Error (" + level + ", " + seed + "): ! m2.equals(t) after save/read");
ensure(t.equals(m2), "Error (" + level + ", " + seed + "): ! t.equals(m2) after save/read");

/* Now we take out of m everything, and check that it is empty. */

for(Iterator i=t.iterator(); i.hasNext(); m2.remove(i.next()));

ensure(m2.isEmpty(), "Error (" + level + ", " + seed + "): m2 is not empty (as it should be)");

/* Now we play with iterators. */

{
    java.util.ListIterator i, j;
    Object J;
    i = (java.util.ListIterator)m.iterator();
    j = new java.util.LinkedList(t).listIterator();

    for(int k = 0; k < 2*n; k++) {
        ensure(i.hasNext() == j.hasNext(), "Error (" + level + ", " + seed + "): divergence in hasNext()");
        ensure(i.hasPrevious() == j.hasPrevious(), "Error (" + level + ", " + seed + "): divergence in hasPrevious()");

        if (r.nextFloat() < .8 && i.hasNext()) {
            ensure(i.next().equals(J = j.next()), "Error (" + level + ", " + seed + "): divergence in next()");

            if (r.nextFloat() < 0.5) {
                i.remove();
                j.remove();
                t.remove(J);
            }
        }
        else if (r.nextFloat() < .2 && i.hasPrevious()) {
            ensure(i.previous().equals(J = j.previous()), "Error (" + level + ", " + seed + "): divergence in previous()");

            if (r.nextFloat() < 0.5) {
                i.remove();
                j.remove();
                t.remove(J);
            }
        }
    }
}

```

```

}

ensure(i.nextIndex() == j.nextIndex(), "Error (" + level + ", " + seed + "): divergence in nextIndex()");
ensure(i.previousIndex() == j.previousIndex(), "Error (" + level + ", " + seed + "): divergence in previousIndex()");

}

}

{
boolean badPrevious = false;
Object previous = null;
it.unimi.dsi.fastutil.BidirectionalIterator i;
java.util.ListIterator j;
Object I, J;
KEY_TYPE from = genKey();
j = new java.util.LinkedList(t).listIterator();
while(j.hasNext()) {
Object k = j.next();
if (((Comparable)k).compareTo(KEY2OBJ(from)) > 0) {
badPrevious = true;
j.previous();
break;
}
previous = k;
}

i = (it.unimi.dsi.fastutil.BidirectionalIterator)m.iterator(from);

for(int k = 0; k < 2*n; k++) {
ensure(i.hasNext() == j.hasNext(), "Error (" + level + ", " + seed + "): divergence in hasNext() (iterator with starting
point " + from + ")");
ensure(i.hasPrevious() == j.hasPrevious() || badPrevious && (i.hasPrevious() == (previous != null)), "Error (" +
level + ", " + seed + "): divergence in hasPrevious() (iterator with starting point " + from + ")" + badPrevious);

if (r.nextFloat() < .8 && i.hasNext()) {
ensure((I = i.next()).equals(J = j.next()), "Error (" + level + ", " + seed + "): divergence in next() (" + I + ", " + J + ",
iterator with starting point " + from + ")");
//System.err.println("Done next " + I + " " + J + " " + badPrevious);

badPrevious = false;

if (r.nextFloat() < 0.5) {
//System.err.println("Removing in next");
i.remove();
j.remove();
t.remove(J);
}
}

```

```

    }
    else if (!badPrevious && r.nextFloat() < .2 && i.hasPrevious()) {
        ensure((I = i.previous()).equals(J = j.previous()), "Error (" + level + ", " + seed + "): divergence in previous() (" + I +
", " + J + ", iterator with starting point " + from + ")");

        if (r.nextFloat() < 0.5) {
            //System.err.println("Removing in prev");
            i.remove();
            j.remove();
            t.remove(J);
        }
    }
}

/* Now we check that m actually holds that data. */

ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after iteration");
ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after iteration");

/* Now we select a pair of keys and create a subset. */

if (! m.isEmpty()) {
    java.util.ListIterator i;
    Object start = m.first(), end = m.first();
    for(i = (java.util.ListIterator)m.iterator(); i.hasNext() && r.nextFloat() < .3; start = end = i.next());
    for(; i.hasNext() && r.nextFloat() < .95; end = i.next());

    //System.err.println("Checking subSet from " + start + " to " + end + " (level=" + (level+1) + ")...");
    testSets((SORTED_SET)m.subSet((KEY_CLASS)start, (KEY_CLASS)end), t.subSet(start, end), n, level + 1);

    ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after subSet");
    ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after subSet");

    //System.err.println("Checking headSet to " + end + " (level=" + (level+1) + ")...");
    testSets((SORTED_SET)m.headSet((KEY_CLASS)end), t.headSet(end), n, level + 1);

    ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after headSet");
    ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after headSet");

    //System.err.println("Checking tailSet from " + start + " (level=" + (level+1) + ")...");
    testSets((SORTED_SET)m.tailSet((KEY_CLASS)start), t.tailSet(start), n, level + 1);

    ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after tailSet");
    ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after tailSet");
}

```

```

}

private static void runTest(int n) {
    RB_TREE_SET m = new RB_TREE_SET();
    SortedSet t = new java.util.TreeSet();
    topSet = m;
    k = new Object[n];
    nk = new Object[n];
    kt = new KEY_TYPE[n];
    nkt = new KEY_TYPE[n];

    for(int i = 0; i < n; i++) {
#ifdef KEY_CLASS_Object
        k[i] = kt[i] = genKey();
        nk[i] = nkt[i] = genKey();
#else
        k[i] = new KEY_CLASS(kt[i] = genKey());
        nk[i] = new KEY_CLASS(nkt[i] = genKey());
#endif
    }

    /* We add pairs to t. */
    for(int i = 0; i < n; i++) t.add(k[i]);

    /* We add to m the same data */
    m.addAll(t);

    testSets(m, t, n, 0);

    System.out.println("Test OK");
    return;
}

public static void main(String args[]) {
    int n = Integer.parseInt(args[1]);
    if (args.length > 2) r = new java.util.Random(seed = Long.parseLong(args[2]));

    try {
        if ("speedTest".equals(args[0]) || "speedComp".equals(args[0])) speedTest(n, "speedComp".equals(args[0]));
        else if ("test".equals(args[0])) runTest(n);
    } catch(Throwable e) {
        e.printStackTrace(System.err);
        System.err.println("seed: " + seed);
    }
}
}

```

```
#endif
```

```
}
```

Found in path(s):

```
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/drv/RBTreeSet.drv
```

No license file was found, but licenses were detected in source scan.

```
/*
```

```
* Copyright (C) 2002-2017 Sebastiano Vigna
```

```
*
```

```
* Licensed under the Apache License, Version 2.0 (the "License");
```

```
* you may not use this file except in compliance with the License.
```

```
* You may obtain a copy of the License at
```

```
*
```

```
* http://www.apache.org/licenses/LICENSE-2.0
```

```
*
```

```
* Unless required by applicable law or agreed to in writing, software
```

```
* distributed under the License is distributed on an "AS IS" BASIS,
```

```
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
```

```
* See the License for the specific language governing permissions and
```

```
* limitations under the License.
```

```
*/
```

```
package PACKAGE;
```

```
import VALUE_PACKAGE.VALUE_COLLECTION;
```

```
import it.unimi.dsi.fastutil.objects.ObjectSortedSet;
```

```
import it.unimi.dsi.fastutil.objects.ObjectBidirectionalIterator;
```

```
import java.util.Map;
```

```
import java.util.SortedMap;
```

```
#if KEYS_REFERENCE
```

```
import java.util.Comparator;
```

```
#endif
```

```
/** A type-specific {@link SortedMap}; provides some additional methods that use polymorphism to avoid (un)boxing.
```

```
*
```

```
* <p>Additionally, this interface strengthens {@link #entrySet()},
```

```
* {@link #keySet()}, {@link #values()},
```

```
* {@link #comparator()}, {@link SortedMap#subMap(Object, Object)}, {@link SortedMap#headMap(Object)} and
```

```

{ @link SortedMap#tailMap(Object)}.
*
* @see SortedMap
*/

public interface SORTED_MAP KEY_VALUE_GENERIC extends MAP KEY_VALUE_GENERIC,
SortedMap<KEY_GENERIC_CLASS, VALUE_GENERIC_CLASS> {

/** Returns a view of the portion of this sorted map whose keys range from { @code fromKey}, inclusive, to
{ @code toKey}, exclusive.
*
* <p>Note that this specification strengthens the one given in { @link SortedMap#subMap(Object,Object)}.
*
* @see SortedMap#subMap(Object,Object)
*/
#if KEYS_REFERENCE
@Override
#endif
SORTED_MAP KEY_VALUE_GENERIC subMap(KEY_GENERIC_TYPE fromKey, KEY_GENERIC_TYPE
toKey);

/** Returns a view of the portion of this sorted map whose keys are strictly less than { @code toKey}.
*
* <p>Note that this specification strengthens the one given in { @link SortedMap#headMap(Object)}.
*
* @see SortedMap#headMap(Object)
*/
#if KEYS_REFERENCE
@Override
#endif
SORTED_MAP KEY_VALUE_GENERIC headMap(KEY_GENERIC_TYPE toKey);

/** Returns a view of the portion of this sorted map whose keys are greater than or equal to { @code fromKey}.
*
* <p>Note that this specification strengthens the one given in { @link SortedMap#tailMap(Object)}.
*
* @see SortedMap#tailMap(Object)
*/
#if KEYS_REFERENCE
@Override
#endif
SORTED_MAP KEY_VALUE_GENERIC tailMap(KEY_GENERIC_TYPE fromKey);

#if KEYS_PRIMITIVE

/** Returns the first (lowest) key currently in this map.
* @see SortedMap#firstKey()
*/

```

```
KEY_GENERIC_TYPE FIRST_KEY();
```

```
/** Returns the last (highest) key currently in this map.
```

```
 * @see SortedMap#lastKey()
```

```
 */
```

```
KEY_GENERIC_TYPE LAST_KEY();
```

```
/** {@inheritDoc}
```

```
 * <p>Note that this specification strengthens the one given in {@link SortedMap#subMap(Object,Object)}.
```

```
 * @deprecated Please use the corresponding type-specific method instead.
```

```
 */
```

```
@Deprecated
```

```
@Override
```

```
default SORTED_MAP KEY_VALUE_GENERIC subMap(final KEY_GENERIC_CLASS from, final  
KEY_GENERIC_CLASS to) {  
    return subMap(KEY_CLASS2TYPE(from), KEY_CLASS2TYPE(to));  
}
```

```
/** {@inheritDoc}
```

```
 * <p>Note that this specification strengthens the one given in {@link SortedMap#headMap(Object)}.
```

```
 * @deprecated Please use the corresponding type-specific method instead.
```

```
 */
```

```
@Deprecated
```

```
@Override
```

```
default SORTED_MAP KEY_VALUE_GENERIC headMap(final KEY_GENERIC_CLASS to) {  
    return headMap(KEY_CLASS2TYPE(to));  
}
```

```
/** {@inheritDoc}
```

```
 * <p>Note that this specification strengthens the one given in {@link SortedMap#tailMap(Object)}.
```

```
 * @deprecated Please use the corresponding type-specific method instead.
```

```
 */
```

```
@Deprecated
```

```
@Override
```

```
default SORTED_MAP KEY_VALUE_GENERIC tailMap(final KEY_GENERIC_CLASS from) {  
    return tailMap(KEY_CLASS2TYPE(from));  
}
```

```
/** {@inheritDoc}
```

```
 * @deprecated Please use the corresponding type-specific method instead.
```

```
 */
```

```
@Deprecated
```

```
@Override
```

```
default KEY_GENERIC_CLASS firstKey() {  
    return KEY2OBJ(FIRST_KEY());  
}
```

```
/** {@inheritDoc}
```

```

* @deprecated Please use the corresponding type-specific method instead.
*/
@Deprecated
@Override
default KEY_GENERIC_CLASS lastKey() {
    return KEY2OBJ(LAST_KEY());
}
#endif

/** A sorted entry set providing fast iteration.
 *
 * <p>In some cases (e.g., hash-based classes) iteration over an entry set requires the creation
 * of a large number of entry objects. Some { @code fastutil }
 * maps might return { @linkplain #entrySet() entry set } objects of type { @code FastSortedEntrySet }: in this case,
 * { @link #fastIterator() fastIterator() }
 * will return an iterator that is guaranteed not to create a large number of objects, <em>possibly
 * by returning always the same entry</em> (of course, mutated).
 */
interface FastSortedEntrySet KEY_VALUE_GENERIC extends ObjectSortedSet<MAP.Entry
KEY_VALUE_GENERIC>, FastEntrySet KEY_VALUE_GENERIC {
    /** { @inheritDoc }
    */
    @Override
    ObjectBidirectionalIterator<MAP.Entry KEY_VALUE_GENERIC> fastIterator();

    /** Returns a fast iterator over this entry set, starting from a given element of the domain (optional operation);
    * the iterator might return always the same entry instance, suitably mutated.
    *
    * @param from an element to start from.
    * @return a fast iterator over this sorted entry set starting at { @code from }; the iterator might return always the
    same entry object, suitably mutated.
    */
    ObjectBidirectionalIterator<MAP.Entry KEY_VALUE_GENERIC> fastIterator(MAP.Entry
KEY_VALUE_GENERIC from);
}

#if KEYS_PRIMITIVE || VALUES_PRIMITIVE
/** Returns a sorted-set view of the mappings contained in this map.
 * <p>Note that this specification strengthens the one given in the
 * corresponding type-specific unsorted map.
 *
 * @return a sorted-set view of the mappings contained in this map.
 * @see SortedMap#entrySet()
 * @deprecated Please use the corresponding type-specific method instead.
 */

```

```

@SuppressWarnings({ "unchecked", "rawtypes" })
@Deprecated
@Override
default ObjectSortedSet<Map.Entry<KEY_GENERIC_CLASS, VALUE_GENERIC_CLASS>> entrySet() {
    return (ObjectSortedSet)ENTRYSET();
}
#else
/** Returns a sorted-set view of the mappings contained in this map.
 * <p>Note that this specification strengthens the one given in the
 * corresponding type-specific unsorted map.
 *
 * @return a sorted-set view of the mappings contained in this map.
 * @see Map#entrySet()
 */

@SuppressWarnings({ "unchecked", "rawtypes" })
@Override
default ObjectSortedSet<Map.Entry<KEY_GENERIC_CLASS, VALUE_GENERIC_CLASS>> entrySet() {
    return (ObjectSortedSet)ENTRYSET();
}
#endif

/** Returns a type-specific sorted-set view of the mappings contained in this map.
 * <p>Note that this specification strengthens the one given in the
 * corresponding type-specific unsorted map.
 *
 * @return a type-specific sorted-set view of the mappings contained in this map.
 * @see #entrySet()
 */

@Override
ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> ENTRYSET();

/** Returns a type-specific sorted-set view of the keys contained in this map.
 * <p>Note that this specification strengthens the one given in the
 * corresponding type-specific unsorted map.
 *
 * @return a sorted-set view of the keys contained in this map.
 * @see SortedMap#keySet()
 */

@Override
SORTED_SET KEY_GENERIC keySet();

/** Returns a type-specific set view of the values contained in this map.
 * <p>Note that this specification strengthens the one given in { @link Map#values() },
 * which was already strengthened in the corresponding type-specific class,
 * but was weakened by the fact that this interface extends { @link SortedMap }.

```

```
*
* @return a set view of the values contained in this map.
* @see SortedMap#values()
*/
```

```
@Override
```

```
VALUE_COLLECTION VALUE_GENERIC values();
```

```
/** Returns the comparator associated with this sorted set, or null if it uses its keys' natural ordering.
```

```
*
```

```
* <p>Note that this specification strengthens the one given in { @link SortedMap#comparator() }.
```

```
*
```

```
* @see SortedMap#comparator()
```

```
*/
```

```
@Override
```

```
KEY_COMPARATOR KEY_SUPER_GENERIC comparator();
```

```
}
```

Found in path(s):

```
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-
a775574/drv/SortedMap.drv
```

No license file was found, but licenses were detected in source scan.

```
/*
```

```
* Copyright (C) 2002-2017 Sebastiano Vigna
```

```
*
```

```
* Licensed under the Apache License, Version 2.0 (the "License");
```

```
* you may not use this file except in compliance with the License.
```

```
* You may obtain a copy of the License at
```

```
*
```

```
* http://www.apache.org/licenses/LICENSE-2.0
```

```
*
```

```
* Unless required by applicable law or agreed to in writing, software
```

```
* distributed under the License is distributed on an "AS IS" BASIS,
```

```
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
```

```
* See the License for the specific language governing permissions and
```

```
* limitations under the License.
```

```
*/
```

```
package PACKAGE;
```

```
/** An abstract class providing basic methods for functions implementing a type-specific interface.
```

```
*
```

```
* <p>This class handles directly a default return
```

```
* value (including { @linkplain #defaultReturnValue() methods to access
```

```
* it}). Instances of classes inheriting from this class have just to return
```

```
* {@code defRetVal} to denote lack of a key in type-specific methods. The value
* is serialized.
*
* <p>Implementing subclasses have just to provide type-specific {@code get()},
* type-specific {@code containsKey()}, and {@code size()} methods.
*
*/
```

```
public abstract class ABSTRACT_FUNCTION_KEY_VALUE_GENERIC implements FUNCTION
KEY_VALUE_GENERIC, java.io.Serializable {
```

```
    private static final long serialVersionUID = -4940583368468432370L;
```

```
    protected ABSTRACT_FUNCTION() {}
```

```
    /**
```

```
     * The default return value for {@code get()}, {@code put()} and
     * {@code remove()}.
    */
```

```
    protected VALUE_GENERIC_TYPE defRetVal;
```

```
    @Override
```

```
    public void defaultReturnValue(final VALUE_GENERIC_TYPE rv) {
        defRetVal = rv;
    }
```

```
    @Override
```

```
    public VALUE_GENERIC_TYPE defaultReturnValue() {
        return defRetVal;
    }
}
```

Found in path(s):

```
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-
a775574/drv/AbstractFunction.drv
```

No license file was found, but licenses were detected in source scan.

```
/*
```

```
* Copyright (C) 2003-2017 Sebastiano Vigna
```

```
*
```

```
* Licensed under the Apache License, Version 2.0 (the "License");
```

```
* you may not use this file except in compliance with the License.
```

```
* You may obtain a copy of the License at
```

```
*
```

```
* http://www.apache.org/licenses/LICENSE-2.0
```

```
*
```

```
* Unless required by applicable law or agreed to in writing, software
```

```
* distributed under the License is distributed on an "AS IS" BASIS,  
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
* See the License for the specific language governing permissions and  
* limitations under the License.  
*/
```

```
package PACKAGE;
```

```
/** An abstract class providing basic methods for sorted sets implementing a type-specific interface. */
```

```
public abstract class ABSTRACT_SORTED_SET KEY_GENERIC extends ABSTRACT_SET KEY_GENERIC  
implements SORTED_SET KEY_GENERIC {
```

```
protected ABSTRACT_SORTED_SET() {}
```

```
@Override
```

```
public abstract KEY_BIDI_ITERATOR KEY_GENERIC iterator();  
}
```

```
Found in path(s):
```

```
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-  
a775574/drv/AbstractSortedSet.drv
```

```
No license file was found, but licenses were detected in source scan.
```

```
/*
```

```
* Copyright (C) 2002-2017 Sebastiano Vigna
```

```
*
```

```
* Licensed under the Apache License, Version 2.0 (the "License");
```

```
* you may not use this file except in compliance with the License.
```

```
* You may obtain a copy of the License at
```

```
*
```

```
* http://www.apache.org/licenses/LICENSE-2.0
```

```
*
```

```
* Unless required by applicable law or agreed to in writing, software
```

```
* distributed under the License is distributed on an "AS IS" BASIS,
```

```
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
```

```
* See the License for the specific language governing permissions and
```

```
* limitations under the License.
```

```
*/
```

```
package PACKAGE;
```

```
import java.util.Collection;
```

```
import it.unimi.dsi.fastutil.objects.ObjectArrays;
```

```

/** A class providing static methods and objects that do useful things with type-specific collections.
 *
 * @see java.util.Collections
 */

public final class COLLECTIONS {

    private COLLECTIONS() {}

    /** An immutable class representing an empty type-specific collection.
     *
     * <p>This class may be useful to implement your own in case you subclass
     * a type-specific collection.
     */

    public abstract static class EmptyCollection KEY_GENERIC extends ABSTRACT_COLLECTION
    KEY_GENERIC {
        protected EmptyCollection() {}

        @Override
        public boolean contains(KEY_TYPE k) { return false; }

        @Override
        public Object[] toArray() { return ObjectArrays.EMPTY_ARRAY; }

        @Override
        SUPPRESS_WARNINGS_KEY_UNCHECKED
        public KEY_BIDI_ITERATOR KEY_GENERIC iterator() { return ITERATORS.EMPTY_ITERATOR; }

        @Override
        public int size() { return 0; }

        @Override
        public void clear() {}

        @Override
        public int hashCode() { return 0; }

        @Override
        public boolean equals(Object o) {
            if (o == this) return true;
            if (! (o instanceof Collection)) return false;
            return ((Collection<?>)o).isEmpty();
        }

        @Override
        public boolean addAll(final Collection<? extends KEY_GENERIC_CLASS> c) { throw new

```

```

UnsupportedOperationException(); }

@Override
public boolean removeAll(final Collection<?> c) { throw new UnsupportedOperationException(); }

@Override
public boolean retainAll(final Collection<?> c) { throw new UnsupportedOperationException(); }

#if KEYS_PRIMITIVE

@Override
public boolean addAll(final COLLECTION c) { throw new UnsupportedOperationException(); }

@Override
public boolean removeAll(final COLLECTION c) { throw new UnsupportedOperationException(); }

@Override
public boolean retainAll(final COLLECTION c) { throw new UnsupportedOperationException(); }

#endif

}

/** A synchronized wrapper class for collections. */

public static class SynchronizedCollection KEY_GENERIC implements COLLECTION KEY_GENERIC,
java.io.Serializable {

    private static final long serialVersionUID = -7046029254386353129L;

    protected final COLLECTION KEY_GENERIC collection;
    protected final Object sync;

    protected SynchronizedCollection(final COLLECTION KEY_GENERIC c, final Object sync) {
        if (c == null) throw new NullPointerException();
        this.collection = c;
        this.sync = sync;
    }

    protected SynchronizedCollection(final COLLECTION KEY_GENERIC c) {
        if (c == null) throw new NullPointerException();
        this.collection = c;
        this.sync = this;
    }

    @Override
    public boolean add(final KEY_GENERIC_TYPE k) { synchronized(sync) { return collection.add(k); } }

```

```

@Override
public boolean contains(final KEY_TYPE k) { synchronized(sync) { return collection.contains(k); } }

@Override
public boolean REMOVE(final KEY_TYPE k) { synchronized(sync) { return collection.REMOVE(k); } }

@Override
public int size() { synchronized(sync) { return collection.size(); } }

@Override
public boolean isEmpty() { synchronized(sync) { return collection.isEmpty(); } }

@Override
public KEY_TYPE[] TO_KEY_ARRAY() { synchronized(sync) { return collection.TO_KEY_ARRAY(); } }

#if KEYS_PRIMITIVE
@Override
public Object[] toArray() { synchronized(sync) { return collection.toArray(); } }

/* @inheritDoc
 * @deprecated Please use {@code toArray()} instead&mdash;this method is redundant and will be removed in the
future.
 */
@Deprecated
@Override
public KEY_TYPE[] TO_KEY_ARRAY(final KEY_TYPE[] a) { return toArray(a); }

@Override
public KEY_TYPE[] toArray(final KEY_TYPE[] a) { synchronized(sync) { return collection.toArray(a); } }

@Override
public boolean addAll(final COLLECTION c) { synchronized(sync) { return collection.addAll(c); } }

@Override
public boolean containsAll(final COLLECTION c) { synchronized(sync) { return collection.containsAll(c); } }

@Override
public boolean removeAll(final COLLECTION c) { synchronized(sync) { return collection.removeAll(c); } }

#ifndef JDK_PRIMITIVE_PREDICATE
@Override
public boolean removeIf(final JDK_PRIMITIVE_PREDICATE filter) { synchronized(sync) { return
collection.removeIf(filter); } }
#endif

@Override
public boolean retainAll(final COLLECTION c) { synchronized(sync) { return collection.retainAll(c); } }

```

```

@Override
@Deprecated
public boolean add(final KEY_GENERIC_CLASS k) { synchronized(sync) { return collection.add(k); } }

@Override
@Deprecated
public boolean contains(final Object k) { synchronized(sync) { return collection.contains(k); } }

@Override
@Deprecated
public boolean remove(final Object k) { synchronized(sync) { return collection.remove(k); } }
#endif

@Override
public <T> T[] toArray(final T[] a) { synchronized(sync) { return collection.toArray(a); } }

@Override
public KEY_ITERATOR KEY_GENERIC iterator() { return collection.iterator(); }

@Override
public boolean addAll(final Collection<? extends KEY_GENERIC_CLASS> c) { synchronized(sync) { return
collection.addAll(c); } }

@Override
public boolean containsAll(final Collection<?> c) { synchronized(sync) { return collection.containsAll(c); } }

@Override
public boolean removeAll(final Collection<?> c) { synchronized(sync) { return collection.removeAll(c); } }

@Override
public boolean retainAll(final Collection<?> c) { synchronized(sync) { return collection.retainAll(c); } }

@Override
public void clear() { synchronized(sync) { collection.clear(); } }

@Override
public String toString() { synchronized(sync) { return collection.toString(); } }

@Override
public int hashCode() { synchronized(sync) { return collection.hashCode(); } }

@Override
public boolean equals(final Object o) { if (o == this) return true; synchronized(sync) { return collection.equals(o); }
}

private void writeObject(java.io.ObjectOutputStream s) throws java.io.IOException {
    synchronized(sync) { s.defaultWriteObject(); }
}

```

```
}  
}
```

```
/** Returns a synchronized collection backed by the specified collection.
```

```
*
```

```
* @param c the collection to be wrapped in a synchronized collection.
```

```
* @return a synchronized view of the specified collection.
```

```
* @see java.util.Collections#synchronizedCollection(Collection)
```

```
*/
```

```
public static KEY_GENERIC COLLECTION KEY_GENERIC synchronize(final COLLECTION KEY_GENERIC  
c) { return new SynchronizedCollection KEY_GENERIC_DIAMOND(c); }
```

```
/** Returns a synchronized collection backed by the specified collection, using an assigned object to synchronize.
```

```
*
```

```
* @param c the collection to be wrapped in a synchronized collection.
```

```
* @param sync an object that will be used to synchronize the list access.
```

```
* @return a synchronized view of the specified collection.
```

```
* @see java.util.Collections#synchronizedCollection(Collection)
```

```
*/
```

```
public static KEY_GENERIC COLLECTION KEY_GENERIC synchronize(final COLLECTION KEY_GENERIC  
c, final Object sync) { return new SynchronizedCollection KEY_GENERIC_DIAMOND(c, sync); }
```

```
/** An unmodifiable wrapper class for collections. */
```

```
public static class UnmodifiableCollection KEY_GENERIC implements COLLECTION KEY_GENERIC,  
java.io.Serializable {
```

```
private static final long serialVersionUID = -7046029254386353129L;
```

```
protected final COLLECTION KEY_GENERIC collection;
```

```
protected UnmodifiableCollection(final COLLECTION KEY_GENERIC c) {
```

```
if (c == null) throw new NullPointerException();
```

```
this.collection = c;
```

```
}
```

```
@Override
```

```
public boolean add(final KEY_GENERIC_TYPE k) { throw new UnsupportedOperationException(); }
```

```
@Override
```

```
public boolean REMOVE(final KEY_TYPE k) { throw new UnsupportedOperationException(); }
```

```
@Override
```

```
public int size() { return collection.size(); }
```

```

@Override
public boolean isEmpty() { return collection.isEmpty(); }

@Override
public boolean contains(final KEY_TYPE o) { return collection.contains(o); }

@Override
public KEY_ITERATOR KEY_GENERIC iterator() { return ITERATORS.unmodifiable(collection.iterator()); }

@Override
public void clear() { throw new UnsupportedOperationException(); }

@Override
public <T> T[] toArray(final T[] a) { return collection.toArray(a); }

@Override
public Object[] toArray() { return collection.toArray(); }

@Override
public boolean containsAll(Collection<?> c) { return collection.containsAll(c); }

@Override
public boolean addAll(Collection<? extends KEY_GENERIC_CLASS> c) { throw new
UnsupportedOperationException(); }

@Override
public boolean removeAll(Collection<?> c) { throw new UnsupportedOperationException(); }

@Override
public boolean retainAll(Collection<?> c) { throw new UnsupportedOperationException(); }

#if KEYS_PRIMITIVE
@Override
@Deprecated
public boolean add(final KEY_GENERIC_CLASS k) { throw new UnsupportedOperationException(); }

@Override
@Deprecated
public boolean contains(final Object k) { return collection.contains(k); }

@Override
@Deprecated
public boolean remove(final Object k) { throw new UnsupportedOperationException(); }

@Override
public KEY_TYPE[] TO_KEY_ARRAY() { return collection.TO_KEY_ARRAY(); }

/* {@inheritDoc}

```

```

* @deprecated Please use {@code toArray()} instead&mdash;this method is redundant.
*/
@Deprecated
@Override
public KEY_TYPE[] TO_KEY_ARRAY(final KEY_TYPE[] a) { return toArray(a); }

@Override
public KEY_TYPE[] toArray(final KEY_TYPE[] a) { return collection.toArray(a); }

@Override
public boolean containsAll(COLLECTION c) { return collection.containsAll(c); }

@Override
public boolean addAll(COLLECTION c) { throw new UnsupportedOperationException(); }

@Override
public boolean removeAll(COLLECTION c) { throw new UnsupportedOperationException(); }

@Override
public boolean retainAll(COLLECTION c) { throw new UnsupportedOperationException(); }
#endif
@Override
public String toString() { return collection.toString(); }

@Override
public int hashCode() { return collection.hashCode(); }

@Override
public boolean equals(final Object o) { if (o == this) return true; return collection.equals(o); }
}

/** Returns an unmodifiable collection backed by the specified collection.
 *
 * @param c the collection to be wrapped in an unmodifiable collection.
 * @return an unmodifiable view of the specified collection.
 * @see java.util.Collections#unmodifiableCollection(Collection)
 */
public static KEY_GENERIC COLLECTION KEY_GENERIC unmodifiable(final COLLECTION
KEY_GENERIC c) { return new UnmodifiableCollection KEY_GENERIC_DIAMOND(c); }

/** A collection wrapper class for iterables. */

public static class IterableCollection KEY_GENERIC extends ABSTRACT_COLLECTION KEY_GENERIC
implements java.io.Serializable {

private static final long serialVersionUID = -7046029254386353129L;

```

```

protected final KEY_ITERABLE KEY_GENERIC iterable;

protected IterableCollection(final KEY_ITERABLE KEY_GENERIC iterable) {
    if (iterable == null) throw new NullPointerException();
    this.iterable = iterable;
}

@Override
public int size() {
    int c = 0;
    final KEY_ITERATOR KEY_GENERIC iterator = iterator();
    while(iterator.hasNext()) {
        iterator.NEXT_KEY();
        c++;
    }

    return c;
}

@Override
public boolean isEmpty() { return ! iterable.iterator().hasNext(); }

@Override
public KEY_ITERATOR KEY_GENERIC iterator() { return iterable.iterator(); }
}

/** Returns an unmodifiable collection backed by the specified iterable.
 *
 * @param iterable the iterable object to be wrapped in an unmodifiable collection.
 * @return an unmodifiable collection view of the specified iterable.
 */
public static KEY_GENERIC COLLECTION KEY_GENERIC asCollection(final KEY_ITERABLE
KEY_GENERIC iterable) {
    if (iterable instanceof COLLECTION) return (COLLECTION KEY_GENERIC)iterable;
    return new IterableCollection KEY_GENERIC_DIAMOND(iterable);
}
}

```

Found in path(s):

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/drv/Collections.drv

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2002-2017 Sebastiano Vigna

*

- * Licensed under the Apache License, Version 2.0 (the "License");
- * you may not use this file except in compliance with the License.
- * You may obtain a copy of the License at
- *
- * <http://www.apache.org/licenses/LICENSE-2.0>
- *
- * Unless required by applicable law or agreed to in writing, software
- * distributed under the License is distributed on an "AS IS" BASIS,
- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.
- */

package PACKAGE;

```
import it.unimi.dsi.fastutil.Hash;
import it.unimi.dsi.fastutil.HashCommon;
import static it.unimi.dsi.fastutil.HashCommon.arraySize;
import static it.unimi.dsi.fastutil.HashCommon.maxFill;
```

```
import java.util.Arrays;
import java.util.Collection;
import java.util.Iterator;
import java.util.NoSuchElementException;
```

```
#ifdef Linked
```

```
#if KEYS_REFERENCE
import java.util.Comparator;
#endif
```

```
/** A type-specific linked hash set with with a fast, small-footprint implementation.
```

```
*
```

```
* <p>Instances of this class use a hash table to represent a set. The table is
* filled up to a specified <em>load factor</em>, and then doubled in size to
* accommodate new entries. If the table is emptied below <em>one fourth</em>
* of the load factor, it is halved in size; however, the table is never reduced to a
* size smaller than that at creation time: this approach makes it
* possible to create sets with a large capacity in which insertions and
* deletions do not cause immediately rehashing. Moreover, halving is
* not performed when deleting entries from an iterator, as it would interfere
* with the iteration process.
```

```
*
```

```
* <p>Note that { @link #clear() } does not modify the hash table size.
* Rather, a family of { @linkplain #trim() trimming
* methods } lets you control the size of the table; this is particularly useful
* if you reuse instances of this class.
```

```

*
* <p>Iterators generated by this set will enumerate elements in the same order in which they
* have been added to the set (addition of elements already present
* in the set does not change the iteration order). Note that this order has nothing in common with the natural
* order of the keys. The order is kept by means of a doubly linked list, represented
* <i>via</i> an array of longs parallel to the table.
*
* <p>This class implements the interface of a sorted set, so to allow easy
* access of the iteration order: for instance, you can get the first element
* in iteration order with { @code first()} without having to create an
* iterator; however, this class partially violates the { @link java.util.SortedSet}
* contract because all subset methods throw an exception and { @link
* #comparator()} returns always { @code null}.
*
* <p>Additional methods, such as { @code addAndMoveToFirst()} , make it easy
* to use instances of this class as a cache (e.g., with LRU policy).
*
* <p>The iterators provided by this class are type-specific { @linkplain
* java.util.ListIterator list iterators}, and can be started at any
* element <em>which is in the set</em> (if the provided element
* is not in the set, a { @link NoSuchElementException} exception will be thrown).
* If, however, the provided element is not the first or last element in the
* set, the first access to the list index will require linear time, as in the worst case
* the entire set must be scanned in iteration order to retrieve the positional
* index of the starting element. If you use just the methods of a type-specific { @link
it.unimi.dsi.fastutil.BidirectionalIterator},
* however, all operations will be performed in constant time.
*
* @see Hash
* @see HashCommon
*/

```

```

public class OPEN_HASH_SET KEY_GENERIC extends ABSTRACT_SORTED_SET KEY_GENERIC
implements java.io.Serializable, Cloneable, Hash {

```

```

#else

```

```

#ifdef Custom

```

```

/** A type-specific hash set with a fast, small-footprint implementation whose { @linkplain
it.unimi.dsi.fastutil.Hash.Strategy hashing strategy }

```

```

* is specified at creation time.
*

```

```

* <p>Instances of this class use a hash table to represent a set. The table is
* filled up to a specified <em>load factor</em>, and then doubled in size to
* accommodate new entries. If the table is emptied below <em>one fourth</em>
* of the load factor, it is halved in size; however, the table is never reduced to a
* size smaller than that at creation time: this approach makes it

```

```

* possible to create sets with a large capacity in which insertions and
* deletions do not cause immediately rehashing. Moreover, halving is
* not performed when deleting entries from an iterator, as it would interfere
* with the iteration process.
*
* <p>Note that { @link #clear() } does not modify the hash table size.
* Rather, a family of { @linkplain #trim() trimming
* methods } lets you control the size of the table; this is particularly useful
* if you reuse instances of this class.
*
* @see Hash
* @see HashCommon
*/

```

```

public class OPEN_HASH_SET KEY_GENERIC extends ABSTRACT_SET KEY_GENERIC implements
java.io.Serializable, Cloneable, Hash {

```

```

#else

```

```

/** A type-specific hash set with with a fast, small-footprint implementation.
*
* <p>Instances of this class use a hash table to represent a set. The table is
* filled up to a specified <em>load factor</em>, and then doubled in size to
* accommodate new entries. If the table is emptied below <em>one fourth</em>
* of the load factor, it is halved in size; however, the table is never reduced to a
* size smaller than that at creation time: this approach makes it
* possible to create sets with a large capacity in which insertions and
* deletions do not cause immediately rehashing. Moreover, halving is
* not performed when deleting entries from an iterator, as it would interfere
* with the iteration process.
*
* <p>Note that { @link #clear() } does not modify the hash table size.
* Rather, a family of { @linkplain #trim() trimming
* methods } lets you control the size of the table; this is particularly useful
* if you reuse instances of this class.
*
* @see Hash
* @see HashCommon
*/

```

```

public class OPEN_HASH_SET KEY_GENERIC extends ABSTRACT_SET KEY_GENERIC implements
java.io.Serializable, Cloneable, Hash {

```

```

#endif

```

```

#endif

```

```

private static final long serialVersionUID = 0L;

```

```

private static final boolean ASSERTS = ASSERTS_VALUE;

/** The array of keys. */
protected transient KEY_GENERIC_TYPE[] key;

/** The mask for wrapping a position counter. */
protected transient int mask;

/** Whether this set contains the null key. */
protected transient boolean containsNull;

#ifdef Custom
/** The hash strategy of this custom set. */
protected STRATEGY KEY_SUPER_GENERIC strategy;
#endif

#ifdef Linked
/** The index of the first entry in iteration order. It is valid iff { @link #size } is nonzero; otherwise, it contains -1. */
protected transient int first = -1;
/** The index of the last entry in iteration order. It is valid iff { @link #size } is nonzero; otherwise, it contains -1. */
protected transient int last = -1;
/** For each entry, the next and the previous entry in iteration order,
 * stored as { @code ((prev & 0xFFFFFFFFFL) << 32) | (next & 0xFFFFFFFFFL)}.
 * The first entry contains predecessor -1, and the last entry
 * contains successor -1. */
protected transient long[] link;
#endif

/** The current table size. Note that an additional element is allocated for storing the null key. */
protected transient int n;

/** Threshold after which we rehash. It must be the table size times { @link #f}. */
protected transient int maxFill;

/** We never resize below this threshold, which is the construction-time {#n}. */
protected final transient int minN;

/** Number of entries in the set (including the null key, if present). */
protected int size;

/** The acceptable load factor. */
protected final float f;

#ifdef Custom
/** Creates a new hash set.
 *
 * <p>The actual table size will be the least power of two greater than { @code expected}/{ @code f}.
 *

```

```

* @param expected the expected number of elements in the hash set.
* @param f the load factor.
* @param strategy the strategy.
*/
SUPPRESS_WARNINGS_KEY_UNCHECKED
public OPEN_HASH_SET(final int expected, final float f, final STRATEGY KEY_SUPER_GENERIC strategy) {
    this.strategy = strategy;
#else
/** Creates a new hash set.
*
* <p>The actual table size will be the least power of two greater than { @code expected}/{ @code f}.
*
* @param expected the expected number of elements in the hash set.
* @param f the load factor.
*/
SUPPRESS_WARNINGS_KEY_UNCHECKED
public OPEN_HASH_SET(final int expected, final float f) {
#endif
    if (f <= 0 || f > 1) throw new IllegalArgumentException("Load factor must be greater than 0 and smaller than or
equal to 1");
    if (expected < 0) throw new IllegalArgumentException("The expected number of elements must be nonnegative");

    this.f = f;

    minN = n = arraySize(expected, f);
    mask = n - 1;
    maxFill = maxFill(n, f);
    key = KEY_GENERIC_ARRAY_CAST new KEY_TYPE[n + 1];
#ifdef Linked
    link = new long[n + 1];
#endif
}

#ifdef Custom
/** Creates a new hash set with { @link Hash#DEFAULT_LOAD_FACTOR} as load factor.
*
* @param expected the expected number of elements in the hash set.
* @param strategy the strategy.
*/

public OPEN_HASH_SET(final int expected, final STRATEGY KEY_SUPER_GENERIC strategy) {
    this(expected, DEFAULT_LOAD_FACTOR, strategy);
}

#else
/** Creates a new hash set with { @link Hash#DEFAULT_LOAD_FACTOR} as load factor.
*

```

```

* @param expected the expected number of elements in the hash set.
*/

public OPEN_HASH_SET(final int expected) {
    this(expected, DEFAULT_LOAD_FACTOR);
}

#endif

#ifdef Custom
/** Creates a new hash set with initial expected {@link Hash#DEFAULT_INITIAL_SIZE} elements
 * and {@link Hash#DEFAULT_LOAD_FACTOR} as load factor.
 * @param strategy the strategy.
 */

public OPEN_HASH_SET(final STRATEGY KEY_SUPER_GENERIC strategy) {
    this(DEFAULT_INITIAL_SIZE, DEFAULT_LOAD_FACTOR, strategy);
}
#else
/** Creates a new hash set with initial expected {@link Hash#DEFAULT_INITIAL_SIZE} elements
 * and {@link Hash#DEFAULT_LOAD_FACTOR} as load factor.
 */

public OPEN_HASH_SET() {
    this(DEFAULT_INITIAL_SIZE, DEFAULT_LOAD_FACTOR);
}
#endif

#ifdef Custom
/** Creates a new hash set copying a given collection.
 *
 * @param c a {@link Collection} to be copied into the new hash set.
 * @param f the load factor.
 * @param strategy the strategy.
 */

public OPEN_HASH_SET(final Collection<? extends KEY_GENERIC_CLASS> c, final float f, final STRATEGY
KEY_SUPER_GENERIC strategy) {
    this(c.size(), f, strategy);
    addAll(c);
}
#else
/** Creates a new hash set copying a given collection.
 *
 * @param c a {@link Collection} to be copied into the new hash set.
 * @param f the load factor.

```

```

*/

public OPEN_HASH_SET(final Collection<? extends KEY_GENERIC_CLASS> c, final float f) {
    this(c.size(), f);
    addAll(c);
}
#endif

#ifdef Custom
/** Creates a new hash set with {@link Hash#DEFAULT_LOAD_FACTOR} as load factor
 * copying a given collection.
 *
 * @param c a {@link Collection} to be copied into the new hash set.
 * @param strategy the strategy.
 */

public OPEN_HASH_SET(final Collection<? extends KEY_GENERIC_CLASS> c, final STRATEGY
KEY_SUPER_GENERIC strategy) {
    this(c, DEFAULT_LOAD_FACTOR, strategy);
}

#else

/** Creates a new hash set with {@link Hash#DEFAULT_LOAD_FACTOR} as load factor
 * copying a given collection.
 *
 * @param c a {@link Collection} to be copied into the new hash set.
 */

public OPEN_HASH_SET(final Collection<? extends KEY_GENERIC_CLASS> c) {
    this(c, DEFAULT_LOAD_FACTOR);
}
#endif

#ifdef Custom
/** Creates a new hash set copying a given type-specific collection.
 *
 * @param c a type-specific collection to be copied into the new hash set.
 * @param f the load factor.
 * @param strategy the strategy.
 */

public OPEN_HASH_SET(final COLLECTION KEY_EXTENDS_GENERIC c, final float f, STRATEGY
KEY_SUPER_GENERIC strategy) {
    this(c.size(), f, strategy);
    addAll(c);
}

```

```

}
#else
/** Creates a new hash set copying a given type-specific collection.
 *
 * @param c a type-specific collection to be copied into the new hash set.
 * @param f the load factor.
 */

public OPEN_HASH_SET(final COLLECTION KEY_EXTENDS_GENERIC c, final float f) {
    this(c.size(), f);
    addAll(c);
}
#endif

#ifdef Custom
/** Creates a new hash set with {@link Hash#DEFAULT_LOAD_FACTOR} as load factor
 * copying a given type-specific collection.
 *
 * @param c a type-specific collection to be copied into the new hash set.
 * @param strategy the strategy.
 */

public OPEN_HASH_SET(final COLLECTION KEY_EXTENDS_GENERIC c, final STRATEGY
KEY_SUPER_GENERIC strategy) {
    this(c, DEFAULT_LOAD_FACTOR, strategy);
}
#else
/** Creates a new hash set with {@link Hash#DEFAULT_LOAD_FACTOR} as load factor
 * copying a given type-specific collection.
 *
 * @param c a type-specific collection to be copied into the new hash set.
 */

public OPEN_HASH_SET(final COLLECTION KEY_EXTENDS_GENERIC c) {
    this(c, DEFAULT_LOAD_FACTOR);
}
#endif

#ifdef Custom
/** Creates a new hash set using elements provided by a type-specific iterator.
 *
 * @param i a type-specific iterator whose elements will fill the set.
 * @param f the load factor.
 * @param strategy the strategy.
 */

```

```

public OPEN_HASH_SET(final STD_KEY_ITERATOR KEY_EXTENDS_GENERIC i, final float f, final
STRATEGY KEY_SUPER_GENERIC strategy) {
    this(DEFAULT_INITIAL_SIZE, f, strategy);
    while(i.hasNext()) add(i.NEXT_KEY());
}
#else
/** Creates a new hash set using elements provided by a type-specific iterator.
 *
 * @param i a type-specific iterator whose elements will fill the set.
 * @param f the load factor.
 */

public OPEN_HASH_SET(final STD_KEY_ITERATOR KEY_EXTENDS_GENERIC i, final float f) {
    this(DEFAULT_INITIAL_SIZE, f);
    while(i.hasNext()) add(i.NEXT_KEY());
}
#endif

#ifdef Custom
/** Creates a new hash set with { @link Hash#DEFAULT_LOAD_FACTOR } as load factor using elements
provided by a type-specific iterator.
 *
 * @param i a type-specific iterator whose elements will fill the set.
 * @param strategy the strategy.
 */

public OPEN_HASH_SET(final STD_KEY_ITERATOR KEY_EXTENDS_GENERIC i, final STRATEGY
KEY_SUPER_GENERIC strategy) {
    this(i, DEFAULT_LOAD_FACTOR, strategy);
}
#else
/** Creates a new hash set with { @link Hash#DEFAULT_LOAD_FACTOR } as load factor using elements
provided by a type-specific iterator.
 *
 * @param i a type-specific iterator whose elements will fill the set.
 */

public OPEN_HASH_SET(final STD_KEY_ITERATOR KEY_EXTENDS_GENERIC i) {
    this(i, DEFAULT_LOAD_FACTOR);
}
#endif

#if KEYS_PRIMITIVE

#ifdef Custom
/** Creates a new hash set using elements provided by an iterator.

```

```

*
* @param i an iterator whose elements will fill the set.
* @param f the load factor.
* @param strategy the strategy.
*/

public OPEN_HASH_SET(final Iterator<?> i, final float f, final STRATEGY KEY_SUPER_GENERIC strategy) {
    this(ITERATORS.AS_KEY_ITERATOR(i), f, strategy);
}
#else
/** Creates a new hash set using elements provided by an iterator.
*
* @param i an iterator whose elements will fill the set.
* @param f the load factor.
*/

public OPEN_HASH_SET(final Iterator<?> i, final float f) {
    this(ITERATORS.AS_KEY_ITERATOR(i), f);
}

#endif

#ifdef Custom
/** Creates a new hash set with { @link Hash#DEFAULT_LOAD_FACTOR } as load factor using elements
provided by an iterator.
*
* @param i an iterator whose elements will fill the set.
* @param strategy the strategy.
*/

public OPEN_HASH_SET(final Iterator<?> i, final STRATEGY KEY_SUPER_GENERIC strategy) {
    this(ITERATORS.AS_KEY_ITERATOR(i), strategy);
}
#else
/** Creates a new hash set with { @link Hash#DEFAULT_LOAD_FACTOR } as load factor using elements
provided by an iterator.
*
* @param i an iterator whose elements will fill the set.
*/

public OPEN_HASH_SET(final Iterator<?> i) {
    this(ITERATORS.AS_KEY_ITERATOR(i));
}
#endif

#endif

```

```

#ifdef Custom
/** Creates a new hash set and fills it with the elements of a given array.
 *
 * @param a an array whose elements will be used to fill the set.
 * @param offset the first element to use.
 * @param length the number of elements to use.
 * @param f the load factor.
 * @param strategy the strategy.
 */

public OPEN_HASH_SET(final KEY_GENERIC_TYPE[] a, final int offset, final int length, final float f, final
STRATEGY KEY_SUPER_GENERIC strategy) {
    this(length < 0 ? 0 : length, f, strategy);
    ARRAYS.ensureOffsetLength(a, offset, length);
    for(int i = 0; i < length; i++) add(a[offset + i]);
}
#else
/** Creates a new hash set and fills it with the elements of a given array.
 *
 * @param a an array whose elements will be used to fill the set.
 * @param offset the first element to use.
 * @param length the number of elements to use.
 * @param f the load factor.
 */

public OPEN_HASH_SET(final KEY_GENERIC_TYPE[] a, final int offset, final int length, final float f) {
    this(length < 0 ? 0 : length, f);
    ARRAYS.ensureOffsetLength(a, offset, length);
    for(int i = 0; i < length; i++) add(a[offset + i]);
}
#endif

#ifdef Custom
/** Creates a new hash set with { @link Hash#DEFAULT_LOAD_FACTOR } as load factor and fills it with the
elements of a given array.
 *
 * @param a an array whose elements will be used to fill the set.
 * @param offset the first element to use.
 * @param length the number of elements to use.
 * @param strategy the strategy.
 */

public OPEN_HASH_SET(final KEY_GENERIC_TYPE[] a, final int offset, final int length, final STRATEGY
KEY_SUPER_GENERIC strategy) {
    this(a, offset, length, DEFAULT_LOAD_FACTOR, strategy);
}

```

```

#else
/** Creates a new hash set with {@link Hash#DEFAULT_LOAD_FACTOR} as load factor and fills it with the
elements of a given array.
*
* @param a an array whose elements will be used to fill the set.
* @param offset the first element to use.
* @param length the number of elements to use.
*/

public OPEN_HASH_SET(final KEY_GENERIC_TYPE[] a, final int offset, final int length) {
    this(a, offset, length, DEFAULT_LOAD_FACTOR);
}
#endif

#ifdef Custom
/** Creates a new hash set copying the elements of an array.
*
* @param a an array to be copied into the new hash set.
* @param f the load factor.
* @param strategy the strategy.
*/

public OPEN_HASH_SET(final KEY_GENERIC_TYPE[] a, final float f, final STRATEGY
KEY_SUPER_GENERIC strategy) {
    this(a, 0, a.length, f, strategy);
}
#else
/** Creates a new hash set copying the elements of an array.
*
* @param a an array to be copied into the new hash set.
* @param f the load factor.
*/

public OPEN_HASH_SET(final KEY_GENERIC_TYPE[] a, final float f) {
    this(a, 0, a.length, f);
}
#endif

#ifdef Custom
/** Creates a new hash set with {@link Hash#DEFAULT_LOAD_FACTOR} as load factor
* copying the elements of an array.
*
* @param a an array to be copied into the new hash set.
* @param strategy the strategy.
*/

public OPEN_HASH_SET(final KEY_GENERIC_TYPE[] a, final STRATEGY KEY_SUPER_GENERIC
strategy) {

```

```

    this(a, DEFAULT_LOAD_FACTOR, strategy);
}
#else
/** Creates a new hash set with {@link Hash#DEFAULT_LOAD_FACTOR} as load factor
 * copying the elements of an array.
 *
 * @param a an array to be copied into the new hash set.
 */

public OPEN_HASH_SET(final KEY_GENERIC_TYPE[] a) {
    this(a, DEFAULT_LOAD_FACTOR);
}
#endif

#ifdef Custom
/** Returns the hashing strategy.
 *
 * @return the hashing strategy of this custom hash set.
 */

public STRATEGY KEY_SUPER_GENERIC strategy() {
    return strategy;
}
#endif

private int realSize() {
    return containsNull ? size - 1 : size;
}

private void ensureCapacity(final int capacity) {
    final int needed = arraySize(capacity, f);
    if (needed > n) rehash(needed);
}

private void tryCapacity(final long capacity) {
    final int needed = (int)Math.min(1 << 30, Math.max(2, HashCommon.nextPowerOfTwo((long)Math.ceil(capacity /
f))));
    if (needed > n) rehash(needed);
}

#if KEYS_PRIMITIVE
@Override
public boolean addAll(COLLECTION c) {
    if (f <= .5) ensureCapacity(c.size()); // The resulting collection will be sized for c.size() elements
    else tryCapacity(size() + c.size()); // The resulting collection will be tentatively sized for size() + c.size() elements
    return super.addAll(c);
}

```

```

#endif

@Override
public boolean addAll(Collection<? extends KEY_GENERIC_CLASS> c) {
    // The resulting collection will be at least c.size() big
    if (f <= .5) ensureCapacity(c.size()); // The resulting collection will be sized for c.size() elements
    else tryCapacity(size() + c.size()); // The resulting collection will be tentatively sized for size() + c.size() elements
    return super.addAll(c);
}

@Override
public boolean add(final KEY_GENERIC_TYPE k) {
    int pos;

    if (KEY_EQUALS_NULL(k)) {
        if (containsNull) return false;
#ifdef Linked
        pos = n;
#endif
        containsNull = true;
#ifdef Custom
        key[n] = k;
#endif
    }
    else {
        KEY_GENERIC_TYPE curr;
        final KEY_GENERIC_TYPE[] key = this.key;

        // The starting point.
        if (! KEY_IS_NULL(curr = key[pos = KEY2INTHASH(k) & mask])) {
            if (KEY_EQUALS_NOT_NULL(curr, k)) return false;
            while(! KEY_IS_NULL(curr = key[pos = (pos + 1) & mask]))
                if (KEY_EQUALS_NOT_NULL(curr, k)) return false;
        }
        key[pos] = k;
    }

#ifdef Linked
    if (size == 0) {
        first = last = pos;
        // Special case of SET_UPPER_LOWER(link[pos], -1, -1);
        link[pos] = -1L;
    }
    else {
        SET_NEXT(link[last], pos);
        SET_UPPER_LOWER(link[pos], last, -1);
        last = pos;
    }
#endif
}

```

```

    }
#endif

    if (size++ >= maxFill) rehash(arraySize(size + 1, f));
    if (ASSERTS) checkTable();
    return true;
}

#if KEY_CLASS_Object
/** Add a random element if not present, get the existing value if already present.
 *
 * This is equivalent to (but faster than) doing a:
 * <pre>
 * K exist = set.get(k);
 * if (exist == null) {
 *   set.add(k);
 *   exist = k;
 * }
 * </pre>
 */
public KEY_GENERIC_TYPE addOrGet(final KEY_GENERIC_TYPE k) {
    int pos;

    if (KEY_EQUALS_NULL(k)) {
        if (containsNull) return key [n];
#ifdef Linked
        pos = n;
#endif
        containsNull = true;
#ifdef Custom
        key [n] = k;
#endif
    }
    else {
        KEY_GENERIC_TYPE curr;
        final KEY_GENERIC_TYPE[] key = this.key;

        // The starting point.
        if (! KEY_IS_NULL(curr = key[pos = KEY2INTHASH(k) & mask])) {
            if (KEY_EQUALS_NOT_NULL(curr, k)) return curr;
            while(! KEY_IS_NULL(curr = key[pos = (pos + 1) & mask]))
                if (KEY_EQUALS_NOT_NULL(curr, k)) return curr;
        }
        key[pos] = k;
    }

#ifdef Linked
    if (size == 0) {

```

```

first = last = pos;
// Special case of SET_UPPER_LOWER(link[pos], -1, -1);
link[pos] = -1L;
}
else {
SET_NEXT(link[last], pos);
SET_UPPER_LOWER(link[pos], last, -1);
last = pos;
}
#endif

if (size++ >= maxFill) rehash(arraySize(size + 1, f));
if (ASSERTS) checkTable();
return k;
}
#endif

/** Shifts left entries with the specified hash code, starting at the specified position,
 * and empties the resulting free entry.
 *
 * @param pos a starting position.
 */
protected final void shiftKeys(int pos) {
// Shift entries with the same hash.
int last, slot;
KEY_GENERIC_TYPE curr;
final KEY_GENERIC_TYPE[] key = this.key;

for(;;) {
pos = ((last = pos) + 1) & mask;

for(;;) {
if (KEY_IS_NULL(curr = key[pos])) {
key[last] = KEY_NULL;
return;
}
slot = KEY2INTHASH(curr) & mask;
if (last <= pos ? last >= slot || slot > pos : last >= slot && slot > pos) break;
pos = (pos + 1) & mask;
}

key[last] = curr;
#ifdef Linked
fixPointers(pos, last);
#endif
}
}

```

```

private boolean removeEntry(final int pos) {
    size--;
#ifdef Linked
    fixPointers(pos);
#endif
    shiftKeys(pos);
    if (n > minN && size < maxFill / 4 && n > DEFAULT_INITIAL_SIZE) rehash(n / 2);
    return true;
}

```

```

private boolean removeNullEntry() {
    containsNull = false;
    key[n] = KEY_NULL;
    size--;
#ifdef Linked
    fixPointers(n);
#endif
    if (n > minN && size < maxFill / 4 && n > DEFAULT_INITIAL_SIZE) rehash(n / 2);
    return true;
}

```

SUPPRESS_WARNINGS_KEY_UNCHECKED

@Override

```

public boolean remove(final KEY_TYPE k) {
    if (KEY_EQUALS_NULL(KEY_GENERIC_CAST k)) {
        if (containsNull) return removeNullEntry();
        return false;
    }
}

```

KEY_GENERIC_TYPE curr;

final KEY_GENERIC_TYPE[] key = this.key;

int pos;

// The starting point.

if (KEY_IS_NULL(curr = key[pos = KEY2INTHASH_CAST(k) & mask])) return false;

if (KEY_EQUALS_NOT_NULL_CAST(k, curr)) return removeEntry(pos);

while(true) {

if (KEY_IS_NULL(curr = key[pos = (pos + 1) & mask])) return false;

if (KEY_EQUALS_NOT_NULL_CAST(k, curr)) return removeEntry(pos);

}

}

SUPPRESS_WARNINGS_KEY_UNCHECKED

@Override

```

public boolean contains(final KEY_TYPE k) {

```

```

    if (KEY_EQUALS_NULL(KEY_GENERIC_CAST k)) return containsNull;

```

```

KEY_GENERIC_TYPE curr;
final KEY_GENERIC_TYPE[] key = this.key;
int pos;

// The starting point.
if (KEY_IS_NULL(curr = key[pos = KEY2INTHASH_CAST(k) & mask])) return false;
if (KEY_EQUALS_NOT_NULL_CAST(k, curr)) return true;

while(true) {
    if (KEY_IS_NULL(curr = key[pos = (pos + 1) & mask])) return false;
    if (KEY_EQUALS_NOT_NULL_CAST(k, curr)) return true;
}
}

#ifdef KEY_CLASS_Object
/** Returns the element of this set that is equal to the given key, or {@code null}.
 * @return the element of this set that is equal to the given key, or {@code null}.
 */
SUPPRESS_WARNINGS_KEY_UNCHECKED
public K get(final Object k) {
    if (KEY_EQUALS_NULL(KEY_GENERIC_CAST k)) return key[pos]; // This is correct independently of the value
of containsNull and of the set being custom

    KEY_GENERIC_TYPE curr;
    final KEY_GENERIC_TYPE[] key = this.key;
    int pos;

    // The starting point.
    if (KEY_IS_NULL(curr = key[pos = KEY2INTHASH_CAST(k) & mask])) return null;
    if (KEY_EQUALS_NOT_NULL_CAST(k, curr)) return curr;
    // There's always an unused entry.
    while(true) {
        if (KEY_IS_NULL(curr = key[pos = (pos + 1) & mask])) return null;
        if (KEY_EQUALS_NOT_NULL_CAST(k, curr)) return curr;
    }
}
#endif

#ifdef Linked
/** Removes the first key in iteration order.
 * @return the first key.
 * @throws NoSuchElementException if this set is empty.
 */
public KEY_GENERIC_TYPE REMOVE_FIRST_KEY() {
    if (size == 0) throw new NoSuchElementException();
    final int pos = first;
    // Abbreviated version of fixPointers(pos)

```

```

first = GET_NEXT(link[pos]);
if (0 <= first) {
    // Special case of SET_PREV(link[first], -1)
    link[first] |= (-1 & 0xFFFFFFFFL) << 32;
}
final KEY_GENERIC_TYPE k = key[pos];
size--;
if (KEY_EQUALS_NULL(k)) {
    containsNull = false;
    key[n] = KEY_NULL;
}
else shiftKeys(pos);
if (n > minN && size < maxFill / 4 && n > DEFAULT_INITIAL_SIZE) rehash(n / 2);
return k;
}

/** Removes the the last key in iteration order.
 * @return the last key.
 * @throws NoSuchElementException is this set is empty.
 */
public KEY_GENERIC_TYPE REMOVE_LAST_KEY() {
    if (size == 0) throw new NoSuchElementException();
    final int pos = last;
    // Abbreviated version of fixPointers(pos)
    last = GET_PREV(link[pos]);
    if (0 <= last) {
        // Special case of SET_NEXT(link[last], -1)
        link[last] |= -1 & 0xFFFFFFFFL;
    }
    final KEY_GENERIC_TYPE k = key[pos];
    size--;
    if (KEY_EQUALS_NULL(k)) {
        containsNull = false;
        key[n] = KEY_NULL;
    }
    else shiftKeys(pos);
    if (n > minN && size < maxFill / 4 && n > DEFAULT_INITIAL_SIZE) rehash(n / 2);
    return k;
}

private void moveIndexToFirst(final int i) {
    if (size == 1 || first == i) return;
    if (last == i) {
        last = GET_PREV(link[i]);
        // Special case of SET_NEXT(link[last], -1);
        link[last] |= -1 & 0xFFFFFFFFL;
    }
    else {

```

```

final long linki = link[i];
final int prev = GET_PREV(linki);
final int next = GET_NEXT(linki);
COPY_NEXT(link[prev], linki);
COPY_PREV(link[next], linki);
}
SET_PREV(link[first], i);
SET_UPPER_LOWER(link[i], -1, first);
first = i;
}

```

```

private void moveIndexToLast(final int i) {
if (size == 1 || last == i) return;
if (first == i) {
first = GET_NEXT(link[i]);
// Special case of SET_PREV(link[first], -1);
link[first] |= (-1 & 0xFFFFFFFFL) << 32;
}
else {
final long linki = link[i];
final int prev = GET_PREV(linki);
final int next = GET_NEXT(linki);
COPY_NEXT(link[prev], linki);
COPY_PREV(link[next], linki);
}
SET_NEXT(link[last], i);
SET_UPPER_LOWER(link[i], last, -1);
last = i;
}

```

```

/** Adds a key to the set; if the key is already present, it is moved to the first position of the iteration order.
 *
 * @param k the key.
 * @return true if the key was not present.
 */

```

```

public boolean addAndMoveToFirst(final KEY_GENERIC_TYPE k) {
int pos;

if (KEY_EQUALS_NULL(k)) {
if (containsNull) {
moveIndexToFirst(n);
return false;
}
containsNull = true;
pos = n;
}
else {
// The starting point.

```

```

final KEY_GENERIC_TYPE key[] = this.key;
pos = KEY2INTHASH(k) & mask;

// There's always an unused entry. TODO
while(! KEY_IS_NULL(key[pos])) {
    if (KEY_EQUALS_NOT_NULL(k, key[pos])) {
        moveIndexToFirst(pos);
        return false;
    }

    pos = (pos + 1) & mask;
}

key[pos] = k;

if (size == 0) {
    first = last = pos;
    // Special case of SET_UPPER_LOWER(link[pos], -1, -1);
    link[pos] = -1L;
}
else {
    SET_PREV(link[first], pos);
    SET_UPPER_LOWER(link[pos], -1, first);
    first = pos;
}

if (size++ >= maxFill) rehash(arraySize(size, f));
if (ASSERTS) checkTable();
return true;
}

/** Adds a key to the set; if the key is already present, it is moved to the last position of the iteration order.
 *
 * @param k the key.
 * @return true if the key was not present.
 */
public boolean addAndMoveToLast(final KEY_GENERIC_TYPE k) {
    int pos;

    if (KEY_EQUALS_NULL(k)) {
        if (containsNull) {
            moveIndexToLast(n);
            return false;
        }
        containsNull = true;
        pos = n;
    }
}

```

```

else {
    // The starting point.
    final KEY_GENERIC_TYPE key[] = this.key;
    pos = KEY2INTHASH(k) & mask;

    // There's always an unused entry.
    while(! KEY_IS_NULL(key[pos])) {
        if (KEY_EQUALS_NOT_NULL(k, key[pos])) {
            moveIndexToLast(pos);
            return false;
        }

        pos = (pos + 1) & mask;
    }
}

key[pos] = k;

if (size == 0) {
    first = last = pos;
    // Special case of SET_UPPER_LOWER(link[pos], -1, -1);
    link[pos] = -1L;
}
else {
    SET_NEXT(link[last], pos);
    SET_UPPER_LOWER(link[pos], last, -1);
    last = pos;
}

if (size++ >= maxFill) rehash(arraySize(size, f));
if (ASSERTS) checkTable();
return true;
}

#endif

/* Removes all elements from this set.
 *
 * <p>To increase object reuse, this method does not change the table size.
 * If you want to reduce the table size, you must use {@link #trim()}.
 *
 */
@Override
public void clear() {
    if (size == 0) return;
    size = 0;
    containsNull = false;
    Arrays.fill(key, KEY_NULL);
}

```

```

#ifdef Linked
    first = last = -1;
#endif
}

@Override
public int size() {
    return size;
}

@Override
public boolean isEmpty() {
    return size == 0;
}

#ifdef Linked

/** Modifies the { @link #link } vector so that the given entry is removed.
 * This method will complete in constant time.
 *
 * @param i the index of an entry.
 */
protected void fixPointers(final int i) {
    if (size == 0) {
        first = last = -1;
        return;
    }
    if (first == i) {
        first = GET_NEXT(link[i]);
        if (0 <= first) {
            // Special case of SET_PREV(link[first], -1)
            link[first] |= (-1 & 0xFFFFFFFFL) << 32;
        }
        return;
    }
    if (last == i) {
        last = GET_PREV(link[i]);
        if (0 <= last) {
            // Special case of SET_NEXT(link[last], -1)
            link[last] |= -1 & 0xFFFFFFFFL;
        }
        return;
    }
    final long linki = link[i];
    final int prev = GET_PREV(linki);
    final int next = GET_NEXT(linki);
    COPY_NEXT(link[prev], linki);
}

```

```

COPY_PREV(link[next], linki);
}

/** Modifies the {@link #link} vector for a shift from s to d.
 * This method will complete in constant time.
 *
 * @param s the source position.
 * @param d the destination position.
 */
protected void fixPointers(int s, int d) {
    if (size == 1) {
        first = last = d;
        // Special case of SET(link[d], -1, -1)
        link[d] = -1L;
        return;
    }
    if (first == s) {
        first = d;
        SET_PREV(link[GET_NEXT(link[s])], d);
        link[d] = link[s];
        return;
    }
    if (last == s) {
        last = d;
        SET_NEXT(link[GET_PREV(link[s])], d);
        link[d] = link[s];
        return;
    }
    final long links = link[s];
    final int prev = GET_PREV(links);
    final int next = GET_NEXT(links);
    SET_NEXT(link[prev], d);
    SET_PREV(link[next], d);
    link[d] = links;
}

/** Returns the first element of this set in iteration order.
 *
 * @return the first element in iteration order.
 */
@Override
public KEY_GENERIC_TYPE FIRST() {
    if (size == 0) throw new NoSuchElementException();
    return key[first];
}

```

```

/** Returns the last element of this set in iteration order.
 *
 * @return the last element in iteration order.
 */
@Override
public KEY_GENERIC_TYPE LAST() {
    if (size == 0) throw new NoSuchElementException();
    return key[last];
}

/** {@inheritDoc}
 * <p>This implementation just throws an {@link UnsupportedOperationException}.*/
@Override
public SORTED_SET KEY_GENERIC tailSet(KEY_GENERIC_TYPE from) { throw new
UnsupportedOperationException(); }

/** {@inheritDoc}
 * <p>This implementation just throws an {@link UnsupportedOperationException}.*/
@Override
public SORTED_SET KEY_GENERIC headSet(KEY_GENERIC_TYPE to) { throw new
UnsupportedOperationException(); }

/** {@inheritDoc}
 * <p>This implementation just throws an {@link UnsupportedOperationException}.*/
@Override
public SORTED_SET KEY_GENERIC subSet(KEY_GENERIC_TYPE from, KEY_GENERIC_TYPE to) { throw
new UnsupportedOperationException(); }

/** {@inheritDoc}
 * <p>This implementation just returns {@code null}.*/
@Override
public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() { return null; }

/** A list iterator over a linked set.
 *
 * <p>This class provides a list iterator over a linked hash set. The constructor runs in constant time.
 */
private class SetIterator implements KEY_LIST_ITERATOR KEY_GENERIC {
    /** The entry that will be returned by the next call to {@link java.util.ListIterator#previous()} (or {@code null} if
no previous entry exists). */
    int prev = -1;
    /** The entry that will be returned by the next call to {@link java.util.ListIterator#next()} (or {@code null} if no
next entry exists). */
    int next = -1;
    /** The last entry that was returned (or -1 if we did not iterate or used {@link #remove()}). */
    int curr = -1;
    /** The current index (in the sense of a {@link java.util.ListIterator}). When -1, we do not know the current

```

```

index.*/
int index = -1;

SetIterator() {
    next = first;
    index = 0;
}

SetIterator(KEY_GENERIC_TYPE from) {
    if (KEY_EQUALS_NULL(from)) {
        if (OPEN_HASH_SET.this.containsNull) {
            next = GET_NEXT(link[n]);
            prev = n;
            return;
        }
        else throw new NoSuchElementException("The key " + from + " does not belong to this set.");
    }

    if (KEY_EQUALS(key[last], from)) {
        prev = last;
        index = size;
        return;
    }

    // The starting point.
    final KEY_GENERIC_TYPE key[] = OPEN_HASH_SET.this.key;
    int pos = KEY2INTHASH(from) & mask;

    // There's always an unused entry.
    while(! KEY_IS_NULL(key[pos])) {
        if (KEY_EQUALS_NOT_NULL(key[pos], from)) {
            // Note: no valid index known.
            next = GET_NEXT(link[pos]);
            prev = pos;
            return;
        }
        pos = (pos + 1) & mask;
    }
    throw new NoSuchElementException("The key " + from + " does not belong to this set.");
}

@Override
public boolean hasNext() { return next != -1; }
@Override
public boolean hasPrevious() { return prev != -1; }

@Override
public KEY_GENERIC_TYPE NEXT_KEY() {

```

```

if (! hasNext()) throw new NoSuchElementException();

curr = next;
next = GET_NEXT(link[curr]);
prev = curr;

if (index >= 0) index++;
if (ASSERTS) assert curr == n || ! KEY_IS_NULL(key[curr]) : "Position " + curr + " is not used";
return key[curr];
}

@Override
public KEY_GENERIC_TYPE PREV_KEY() {
if (! hasPrevious()) throw new NoSuchElementException();

curr = prev;
prev = GET_PREV(link[curr]);
next = curr;

if (index >= 0) index--;

return key[curr];
}

private final void ensureIndexKnown() {
if (index >= 0) return;
if (prev == -1) {
index = 0;
return;
}
if (next == -1) {
index = size;
return;
}
int pos = first;
index = 1;
while(pos != prev) {
pos = GET_NEXT(link[pos]);
index++;
}
}

@Override
public int nextIndex() {
ensureIndexKnown();
return index;
}

```

```

@Override
public int previousIndex() {
    ensureIndexKnown();
    return index - 1;
}

@Override
public void remove() {
    ensureIndexKnown();
    if (curr == -1) throw new IllegalStateException();

    if (curr == prev) {
        /* If the last operation was a next(), we are removing an entry that precedes
           the current index, and thus we must decrement it. */
        index--;
        prev = GET_PREV(link[curr]);
    }
    else
        next = GET_NEXT(link[curr]);

    size--;
    /* Now we manually fix the pointers. Because of our knowledge of next
       and prev, this is going to be faster than calling fixPointers(). */
    if (prev == -1) first = next;
    else
        SET_NEXT(link[prev], next);
    if (next == -1) last = prev;
    else
        SET_PREV(link[next], prev);

    int last, slot, pos = curr;
    curr = -1;

    if (pos == n) {
        OPEN_HASH_SET.this.containsNull = false;
        OPEN_HASH_SET.this.key[pos] = KEY_NULL;
    }
    else {
        KEY_GENERIC_TYPE curr;
        final KEY_GENERIC_TYPE[] key = OPEN_HASH_SET.this.key;
        // We have to horribly duplicate the shiftKeys() code because we need to update next/prev.
        for(;;) {
            pos = ((last = pos) + 1) & mask;
            for(;;) {
                if (KEY_IS_NULL(curr = key[pos])) {
                    key[last] = KEY_NULL;
                    return;
                }
            }
        }
    }
}

```

```

    slot = KEY2INTHASH(curr) & mask;
    if (last <= pos ? last >= slot || slot > pos : last >= slot && slot > pos) break;
    pos = (pos + 1) & mask;
}

key[last] = curr;
if (next == pos) next = last;
if (prev == pos) prev = last;
fixPointers(pos, last);
}
}
}
}

/** Returns a type-specific list iterator on the elements in this set, starting from a given element of the set.
 * Please see the class documentation for implementation details.
 *
 * @param from an element to start from.
 * @return a type-specific list iterator starting at the given element.
 * @throws IllegalArgumentException if { @code from } does not belong to the set.
 */
@Override
public KEY_LIST_ITERATOR KEY_GENERIC iterator(KEY_GENERIC_TYPE from) {
    return new SetIterator(from);
}

/** Returns a type-specific list iterator on the elements in this set, starting from the first element.
 * Please see the class documentation for implementation details.
 *
 * @return a type-specific list iterator starting at the first element.
 */
@Override
public KEY_LIST_ITERATOR KEY_GENERIC iterator() {
    return new SetIterator();
}

#else

/** An iterator over a hash set. */

private class SetIterator implements KEY_ITERATOR KEY_GENERIC {
    /** The index of the last entry returned, if positive or zero; initially, { @link #n}. If negative, the last
     * element returned was that of index { @code - pos - 1 } from the { @link #wrapped } list. */
    int pos = n;
    /** The index of the last entry that has been returned (more precisely, the value of { @link #pos } if { @link #pos } is
     * positive,
     * or { @link Integer#MIN_VALUE } if { @link #pos } is negative). It is -1 if either

```

```

we did not return an entry yet, or the last returned entry has been removed. */
int last = -1;
/** A downward counter measuring how many entries must still be returned. */
int c = size;
/** A boolean telling us whether we should return the null key. */
boolean mustReturnNull = OPEN_HASH_SET.this.containsNull;
/** A lazily allocated list containing elements that have wrapped around the table because of removals. */
ARRAY_LIST KEY_GENERIC wrapped;

@Override
public boolean hasNext() {
    return c != 0;
}

@Override
public KEY_GENERIC_TYPE NEXT_KEY() {
    if (! hasNext()) throw new NoSuchElementException();
    c--;
    if (mustReturnNull) {
        mustReturnNull = false;
        last = n;
        return key[n];
    }
    final KEY_GENERIC_TYPE key[] = OPEN_HASH_SET.this.key;
    for(;;) {
        if (--pos < 0) {
            // We are just enumerating elements from the wrapped list.
            last = Integer.MIN_VALUE;
            return wrapped.GET_KEY(- pos - 1);
        }
        if (! KEY_IS_NULL(key[pos])) return key[last = pos];
    }
}

/** Shifts left entries with the specified hash code, starting at the specified position,
 * and empties the resulting free entry.
 *
 * @param pos a starting position.
 */
private final void shiftKeys(int pos) {
    // Shift entries with the same hash.
    int last, slot;
    KEY_GENERIC_TYPE curr;
    final KEY_GENERIC_TYPE[] key = OPEN_HASH_SET.this.key;

    for(;;) {
        pos = ((last = pos) + 1) & mask;

```

```

for(;;) {
    if (KEY_IS_NULL(curr = key[pos])) {
        key[last] = KEY_NULL;
        return;
    }
    slot = KEY2INTHASH(curr) & mask;
    if (last <= pos ? last >= slot || slot > pos : last >= slot && slot > pos) break;
    pos = (pos + 1) & mask;
}

if (pos < last) { // Wrapped entry.
    if (wrapped == null) wrapped = new ARRAY_LIST KEY_GENERIC_DIAMOND(2);
    wrapped.add(key[pos]);
}

key[last] = curr;
}
}

@Override
public void remove() {
    if (last == -1) throw new IllegalStateException();
    if (last == n) {
        OPEN_HASH_SET.this.containsNull = false;
        OPEN_HASH_SET.this.key[n] = KEY_NULL;
    }
    else if (pos >= 0) shiftKeys(last);
    else {
        // We're removing wrapped entries.
#ifdef KEYS_REFERENCE
        OPEN_HASH_SET.this.remove(wrapped.set(- pos - 1, null));
#else
        OPEN_HASH_SET.this.remove(wrapped.GET_KEY(- pos - 1));
#endif
        last = -1; // Note that we must not decrement size
        return;
    }

    size--;
    last = -1; // You can no longer remove this entry.
    if (ASSERTS) checkTable();
}
}

@Override
public KEY_ITERATOR KEY_GENERIC iterator() {
    return new SetIterator();
}

```

```
}
```

```
#endif
```

```
/** Reshapes this set, making the table as small as possible.
```

```
*
```

```
* <p>This method reshapes the table to the smallest size satisfying the  
* load factor. It can be used when the set will not be changed anymore, so  
* to optimize access speed and size.
```

```
*
```

```
* <p>If the table size is already the minimum possible, this method  
* does nothing.
```

```
*
```

```
* @return true if there was enough memory to trim the set.
```

```
* @see #trim(int)
```

```
*/
```

```
public boolean trim() {  
    final int l = arraySize(size, f);  
    if (l >= n || size > maxFill(l, f)) return true;  
    try {  
        rehash(l);  
    }  
    catch(OutOfMemoryError cantDoIt) { return false; }  
    return true;  
}
```

```
/** Reshapes this set if the table is too large.
```

```
*
```

```
* <p>Let <var>N</var> be the smallest table size that can hold  
* <code>max(n,{ @link #size()})</code> entries, still satisfying the load factor. If the current  
* table size is smaller than or equal to <var>N</var>, this method does  
* nothing. Otherwise, it reshapes this set in a table of size  
* <var>N</var>.
```

```
*
```

```
* <p>This method is useful when reusing sets. { @linkplain #clear() Clearing a  
* set} leaves the table size untouched. If you are reusing a set  
* many times, you can call this method with a typical  
* size to avoid keeping around a very large table just  
* because of a few large transient sets.
```

```
*
```

```
* @param n the threshold for the trimming.
```

```
* @return true if there was enough memory to trim the set.
```

```
* @see #trim()
```

```
*/
```

```

public boolean trim(final int n) {
    final int l = HashCommon.nextPowerOfTwo((int)Math.ceil(n / f));
    if (l >= n || size > maxFill(l, f)) return true;
    try {
        rehash(l);
    }
    catch(OutOfMemoryError cantDoIt) { return false; }
    return true;
}

/** Rehashes the set.
 *
 * <p>This method implements the basic rehashing strategy, and may be
 * overridden by subclasses implementing different rehashing strategies (e.g.,
 * disk-based rehashing). However, you should not override this method
 * unless you understand the internal workings of this class.
 *
 * @param newN the new size
 */

SUPPRESS_WARNINGS_KEY_UNCHECKED
protected void rehash(final int newN) {
    final KEY_GENERIC_TYPE key[] = this.key;

    final int mask = newN - 1; // Note that this is used by the hashing macro
    final KEY_GENERIC_TYPE newKey[] = KEY_GENERIC_ARRAY_CAST new KEY_TYPE[newN + 1];

#ifdef Linked
    int i = first, prev = -1, newPrev = -1, t, pos;
    final long link[] = this.link;
    final long newLink[] = new long[newN + 1];
    first = -1;

    for(int j = size; j-- != 0;) {
        if (KEY_EQUALS_NULL(key[i])) pos = newN;
        else {
            pos = KEY2INTHASH(key[i]) & mask;
            while (! KEY_IS_NULL(newKey[pos])) pos = (pos + 1) & mask;
        }

        newKey[pos] = key[i];

        if (prev != -1) {
            SET_NEXT(newLink[newPrev], pos);
            SET_PREV(newLink[pos], newPrev);
            newPrev = pos;
        }
        else {

```

```

    newPrev = first = pos;
    // Special case of SET(newLink[pos], -1, -1);
    newLink[pos] = -1L;
}

t = i;
i = GET_NEXT(link[i]);
prev = t;
}

this.link = newLink;
this.last = newPrev;
if (newPrev != -1)
    // Special case of SET_NEXT(newLink[newPrev], -1);
    newLink[newPrev] |= -1 & 0xFFFFFFFFL;
#else
int i = n, pos;

for(int j = realSize(); j-- != 0;) {
    while(KEY_IS_NULL(key[--i]));
    if (! KEY_IS_NULL(newKey[pos = KEY2INTHASH(key[i] & mask)])
        while (! KEY_IS_NULL(newKey[pos = (pos + 1) & mask]));
    newKey[pos] = key[i];
}
#endif

n = newN;
this.mask = mask;
maxFill = maxFill(n, f);
this.key = newKey;
}

/** Returns a deep copy of this set.
 *
 * <p>This method performs a deep copy of this hash set; the data stored in the
 * set, however, is not cloned. Note that this makes a difference only for object keys.
 *
 * @return a deep copy of this set.
 */
@Override
SUPPRESS_WARNINGS_KEY_UNCHECKED
public OPEN_HASH_SET KEY_GENERIC clone() {
    OPEN_HASH_SET KEY_GENERIC c;
    try {
        c = (OPEN_HASH_SET KEY_GENERIC)super.clone();
    }
    catch(CloneNotSupportedException cantHappen) {

```

```

        throw new InternalError();
    }
    c.key = key.clone();
    c.containsNull = containsNull;
#ifdef Linked
    c.link = link.clone();
#endif
#ifdef Custom
    c.strategy = strategy;
#endif
    return c;
}

/** Returns a hash code for this set.
 *
 * This method overrides the generic method provided by the superclass.
 * Since {@code equals()} is not overridden, it is important
 * that the value returned by this method is the same value as
 * the one returned by the overridden method.
 *
 * @return a hash code for this set.
 */
@Override
public int hashCode() {
    int h = 0;
    for(int j = realSize(), i = 0; j-- != 0;) {
        while(KEY_IS_NULL(key[i])) i++;
#ifdef KEYS_REFERENCE
        if (this != key[i])
#endif
            h += KEY2JAVAHASH_NOT_NULL(key[i]);
        i++;
    }

    // Zero / null have hash zero.
    return h;
}

private void writeObject(java.io.ObjectOutputStream s) throws java.io.IOException {
    final KEY_ITERATOR KEY_GENERIC i = iterator();
    s.defaultWriteObject();
    for(int j = size; j-- != 0;) s.WRITE_KEY(i.NEXT_KEY());
}

SUPPRESS_WARNINGS_KEY_UNCHECKED
private void readObject(java.io.ObjectInputStream s) throws java.io.IOException, ClassNotFoundException {

```

```

s.defaultReadObject();

n = arraySize(size, f);
maxFill = maxFill(n, f);
mask = n - 1;

final KEY_GENERIC_TYPE key[] = this.key = KEY_GENERIC_ARRAY_CAST new KEY_TYPE[n + 1];
#ifdef Linked
final long link[] = this.link = new long[n + 1];
int prev = -1;
first = last = -1;
#endif

KEY_GENERIC_TYPE k;

for(int i = size, pos; i-- != 0;) {

k = KEY_GENERIC_CAST s.READ_KEY();

if (KEY_EQUALS_NULL(k) {
pos = n;
containsNull = true;
}
else {
if (! KEY_IS_NULL(key[pos = KEY2INTHASH(k) & mask]))
while (! KEY_IS_NULL(key[pos = (pos + 1) & mask]));
}

key[pos] = k;

#ifdef Linked
if (first != -1) {
SET_NEXT(link[prev], pos);
SET_PREV(link[pos], prev);
prev = pos;
}
else {
prev = first = pos;
// Special case of SET_PREV(newLink[pos], -1);
link[pos] |= (-1L & 0xFFFFFFFFL) << 32;
}
#endif
}

#ifdef Linked
last = prev;
if (prev != -1)
// Special case of SET_NEXT(link[prev], -1);

```

```

    link[prev] |= -1 & 0xFFFFFFFFFL;
#endif

    if (ASSERTS) checkTable();
}

#ifdef ASSERTS_CODE
private void checkTable() {
    assert (n & -n) == n : "Table length is not a power of two: " + n;
    assert n == key.length - 1;
    int n = key.length - 1;
    while(n-- != 0)
        if (! KEY_IS_NULL(key[n]) && ! contains(key[n]))
            throw new AssertionError("Hash table has key " + key[n] + " marked as occupied, but the key does not belong to the
table");

#ifdef KEYS_PRIMITIVE
    java.util.HashSet<KEY_GENERIC_CLASS> s = new java.util.HashSet<KEY_GENERIC_CLASS> ();
#else
    java.util.HashSet<Object> s = new java.util.HashSet<Object>();
#endif

    for(int i = key.length - 1; i-- != 0;)
        if (! KEY_IS_NULL(key[i]) && ! s.add(key[i])) throw new AssertionError("Key " + key[i] + " appears twice at
position " + i);

#ifdef Linked
    KEY_LIST_ITERATOR KEY_GENERIC i = iterator();
    KEY_GENERIC_TYPE k;
    n = size();
    while(n-- != 0)
        if (! contains(k = i.NEXT_KEY()))
            throw new AssertionError("Linked hash table forward enumerates key " + k + ", but the key does not belong to the
table");

    if (i.hasNext()) throw new AssertionError("Forward iterator not exhausted");

    n = size();
    if (n > 0) {
        i = iterator(LAST());
        while(n-- != 0)
            if (! contains(k = i.PREV_KEY()))
                throw new AssertionError("Linked hash table backward enumerates key " + k + ", but the key does not belong to the
table");

        if (i.hasPrevious()) throw new AssertionError("Previous iterator not exhausted");
    }
}

```

```

#endif
}
#else
private void checkTable() {}
#endif

#ifdef TEST

private static long seed = System.currentTimeMillis();
private static java.util.Random r = new java.util.Random(seed);

private static KEY_TYPE genKey() {
#ifdef KEY_CLASS_Byte || KEY_CLASS_Short || KEY_CLASS_Character
return (KEY_TYPE)r.nextInt();
#elif KEYS_PRIMITIVE
return r.NEXT_KEY();
#elif KEY_CLASS_Object
#ifdef Custom
int i = r.nextInt(3);
byte a[] = new byte[i];
while(i-- != 0) a[i] = (byte)r.nextInt();
return a;
#else
return Integer.toBinaryString(r.nextInt());
#endif
#else
return new java.io.Serializable() {};
#endif
}

private static final class ArrayComparator implements java.util.Comparator {
public int compare(Object a, Object b) {
byte[] aa = (byte[])a;
byte[] bb = (byte[])b;
int length = Math.min(aa.length, bb.length);
for(int i = 0; i < length; i++) {
if (aa[i] < bb[i]) return -1;
if (aa[i] > bb[i]) return 1;
}
return aa.length == bb.length ? 0 : (aa.length < bb.length ? -1 : 1);
}
}

private static final class MockSet extends java.util.TreeSet {
private java.util.List list = new java.util.ArrayList();

public MockSet(java.util.Comparator c) { super(c); }
}

```

```

public boolean add(Object k) {
    if (! contains(k)) list.add(k);
    return super.add(k);
}

public boolean addAll(Collection c) {
    java.util.Iterator i = c.iterator();
    boolean result = false;
    while(i.hasNext()) result |= add(i.next());
    return result;
}

public boolean removeAll(Collection c) {
    java.util.Iterator i = c.iterator();
    boolean result = false;
    while(i.hasNext()) result |= remove(i.next());
    return result;
}

public boolean remove(Object k) {
    if (contains(k)) {
        int i = list.size();
        while(i-- != 0) if (comparator().compare(list.get(i), k) == 0) {
            list.remove(i);
            break;
        }
    }
    return super.remove(k);
}

private void justRemove(Object k) { super.remove(k); }

public java.util.Iterator iterator() {
    return new java.util.Iterator() {
        final java.util.Iterator iterator = list.iterator();
        Object curr;
        public Object next() { return curr = iterator.next(); }
        public boolean hasNext() { return iterator.hasNext(); }
        public void remove() {
            justRemove(curr);
            iterator.remove();
        }
    };
}

private static java.text.NumberFormat format = new java.text.DecimalFormat("#,###.00");

```

```

private static java.text.FieldPosition fp = new java.text.FieldPosition(0);

private static String format(double d) {
    StringBuffer s = new StringBuffer();
    return format.format(d, s, fp).toString();
}

private static void speedTest(int n, float f, boolean comp) {
#ifdef Custom
    int i, j;
    OPEN_HASH_SET m;
#ifdef Linked
    java.util.LinkedHashSet t;
#else
    java.util.HashSet t;
#endif

    KEY_TYPE k[] = new KEY_TYPE[n];
    KEY_TYPE nk[] = new KEY_TYPE[n];
    long ns;

    for(i = 0; i < n; i++) {
        k[i] = genKey();
        nk[i] = genKey();
    }

    double totAdd = 0, totYes = 0, totNo = 0, totIter = 0, totRemYes = 0, totRemNo = 0, d;

    if (comp) { for(j = 0; j < 20; j++) {

#ifdef Linked
        t = new java.util.LinkedHashSet(16);
#else
        t = new java.util.HashSet(16);
#endif

        /* We add pairs to t. */
        ns = System.nanoTime();
        for(i = 0; i < n; i++) t.add(KEY2OBJ(k[i]));
        d = (System.nanoTime() - ns) / (double)n;
        if (j > 2) totAdd += d;
        System.out.print("Add: " + format(d) + "ns ");

        /* We check for pairs in t. */
        ns = System.nanoTime();
        for(i = 0; i < n; i++) t.contains(KEY2OBJ(k[i]));
        d = (System.nanoTime() - ns) / (double)n;
        if (j > 2) totYes += d;

```

```

System.out.print("Yes: " + format(d) + "ns ");

/* We check for pairs not in t. */
ns = System.nanoTime();
for(i = 0; i < n; i++) t.contains(KEY2OBJ(nk[i]));
d = (System.nanoTime() - ns) / (double)n;
if (j > 2) totNo += d;
System.out.print("No: " + format(d) + "ns ");

/* We iterate on t. */
ns = System.nanoTime();
for(java.util.Iterator it = t.iterator(); it.hasNext(); it.next());
d = (System.nanoTime() - ns) / (double)n;
if (j > 2) totIter += d;
System.out.print("Iter: " + format(d) + "ns ");

// Too expensive in the linked case
#ifdef Linked
/* We delete pairs not in t. */
ns = System.nanoTime();
for(i = 0; i < n; i++) t.remove(KEY2OBJ(nk[i]));
d = (System.nanoTime() - ns) / (double)n;
if (j > 2) totRemNo += d;
System.out.print("RemNo: " + format(d) + "ns ");

/* We delete pairs in t. */
ns = System.nanoTime();
for(i = 0; i < n; i++) t.remove(KEY2OBJ(k[i]));
d = (System.nanoTime() - ns) / (double)n;
if (j > 2) totRemYes += d;
System.out.print("RemYes: " + format(d) + "ns ");
#endif

System.out.println();
}

System.out.println();
System.out.println("java.util Add: " + format(totAdd/(j-3)) + "ns Yes: " + format(totYes/(j-3)) + "ns No: " +
format(totNo/(j-3)) + "ns Iter: " + format(totIter/(j-3)) + "ns RemNo: " + format(totRemNo/(j-3)) + "ns RemYes: " +
format(totRemYes/(j-3)) + "ns");

System.out.println();

totAdd = totYes = totNo = totIter = totRemYes = totRemNo = 0;
}

for(j = 0; j < 20; j++) {

```

```

m = new OPEN_HASH_SET(16, f);

/* We add pairs to m. */
ns = System.nanoTime();
for(i = 0; i < n; i++) m.add(k[i]);
d = (System.nanoTime() - ns) / (double)n;
if (j > 2) totAdd += d;
System.out.print("Add: " + format(d) + "ns ");

/* We check for pairs in m. */
ns = System.nanoTime();
for(i = 0; i < n; i++) m.contains(k[i]);
d = (System.nanoTime() - ns) / (double)n;
if (j > 2) totYes += d;
System.out.print("Yes: " + format(d) + "ns ");

/* We check for pairs not in m. */
ns = System.nanoTime();
for(i = 0; i < n; i++) m.contains(nk[i]);
d = (System.nanoTime() - ns) / (double)n;
if (j > 2) totNo += d;
System.out.print("No: " + format(d) + "ns ");

/* We iterate on m. */
ns = System.nanoTime();
for(KEY_ITERATOR it = (KEY_ITERATOR)m.iterator(); it.hasNext(); it.NEXT_KEY());
d = (System.nanoTime() - ns) / (double)n;
if (j > 2) totIter += d;
System.out.print("Iter: " + format(d) + "ns ");

// Too expensive in the linked case
#ifdef Linked
/* We delete pairs not in m. */
ns = System.nanoTime();
for(i = 0; i < n; i++) m.remove(nk[i]);
d = (System.nanoTime() - ns) / (double)n;
if (j > 2) totRemNo += d;
System.out.print("RemNo: " + format(d) + "ns ");

/* We delete pairs in m. */
ns = System.nanoTime();
for(i = 0; i < n; i++) m.remove(k[i]);
d = (System.nanoTime() - ns) / (double)n;
if (j > 2) totRemYes += d;
System.out.print("RemYes: " + format(d) + "ns ");
#endif

System.out.println();

```

```

}

System.out.println();
System.out.println("fastutil Add: " + format(totAdd/(j-3)) + "ns Yes: " + format(totYes/(j-3)) + "ns No: " +
format(totNo/(j-3)) + "ns Iter: " + format(totIter/(j-3)) + "ns RemNo: " + format(totRemNo/(j-3)) + "ns RemYes: " +
format(totRemYes/(j-3)) + "ns");

System.out.println();
#endif
}

private static void fatal(String msg) {
System.out.println(msg);
System.exit(1);
}

private static void ensure(boolean cond, String msg) {
if (cond) return;
fatal(msg);
}

private static void printProbes(OPEN_HASH_SET m) {
long totProbes = 0;
double totSquareProbes = 0;
int maxProbes = 0;
final double f = (double)m.size / m.n;
for(int i = 0, c = 0; i < m.n; i++) {
if (! KEY_IS_NULL(m.key[i])) c++;
else {
if (c != 0) {
final long p = (c + 1) * (c + 2) / 2;
totProbes += p;
totSquareProbes += (double)p * p;
}
maxProbes = Math.max(c, maxProbes);
c = 0;
totProbes++;
totSquareProbes++;
}
}

final double expected = (double)totProbes / m.n;
System.err.println("Expected probes: " + (
3 * Math.sqrt(3) * (f / ((1 - f) * (1 - f))) + 4 / (9 * f) - 1
) + "; actual: " + expected + "; stddev: " + Math.sqrt(totSquareProbes / m.n - expected * expected) + "; max probes:

```

```

" + maxProbes);
}

private static void runTest(int n, float f) {
#if !defined(Custom) || KEYS_REFERENCE

    int c;
#ifdef Custom
    OPEN_HASH_SET m = new OPEN_HASH_SET(Hash.DEFAULT_INITIAL_SIZE, f,
it.unimi.dsi.fastutil.bytes.ByteArrays.HASH_STRATEGY);
#else
    OPEN_HASH_SET m = new OPEN_HASH_SET(Hash.DEFAULT_INITIAL_SIZE, f);
#endif
#ifdef Linked
#ifdef Custom
    java.util.Set t = new MockSet(new ArrayComparator());
#else
    java.util.Set t = new java.util.LinkedHashSet();
#endif
#else
#ifdef Custom
    java.util.Set t = new java.util.TreeSet(new ArrayComparator());
#else
    java.util.Set t = new java.util.HashSet();
#endif
#endif

    /* First of all, we fill t with random data. */

    for(int i=0; i<f * n; i++) t.add(KEY2OBJ(genKey()));

    /* Now we add to m the same data */

    m.addAll(t);

    if (!m.equals(t)) System.out.println("Error (" + seed + "): !m.equals(t) after insertion");
    if (!t.equals(m)) System.out.println("Error (" + seed + "): !t.equals(m) after insertion");
    printProbes(m);

    /* Now we check that m actually holds that data. */

    for(java.util.Iterator i=t.iterator(); i.hasNext();) {
        Object e = i.next();
        if (!m.contains(e)) {
            System.out.println("Error (" + seed + "): m and t differ on a key (" + e + ") after insertion (iterating on t)");
            System.exit(1);
        }
    }
}

```

```

}

/* Now we check that m actually holds that data, but iterating on m. */

c = 0;
for(java.util.Iterator i=m.iterator(); i.hasNext();) {
    Object e = i.next();
    c++;
    if (!t.contains(e)) {
        System.out.println("Error (" + seed + "): m and t differ on a key (" + e + ") after insertion (iterating on m)");
        System.exit(1);
    }
}

if (c != t.size()) {
    System.out.println("Error (" + seed + "): m has only " + c + " keys instead of " + t.size() + " after insertion (iterating
on m)");
    System.exit(1);
}

/* Now we check that inquiries about random data give the same answer in m and t. For
m we use the polymorphic method. */

for(int i=0; i<n; i++) {
    KEY_TYPE T = genKey();
    if (m.contains(T) != t.contains(KEY2OBJ(T))) {
        System.out.println("Error (" + seed + "): divergence in keys between t and m (polymorphic method)");
        System.exit(1);
    }
}

/* Again, we check that inquiries about random data give the same answer in m and t, but
for m we use the standard method. */

for(int i=0; i<n; i++) {
    KEY_TYPE T = genKey();
    if (m.contains(KEY2OBJ(T)) != t.contains(KEY2OBJ(T))) {
        System.out.println("Error (" + seed + "): divergence between t and m (standard method)");
        System.exit(1);
    }
}

/* Now we put and remove random data in m and t, checking that the result is the same. */

for(int i=0; i<20*n; i++) {
    KEY_TYPE T = genKey();
    if (m.add(KEY2OBJ(T)) != t.add(KEY2OBJ(T))) {
        System.out.println("Error (" + seed + "): divergence in add() between t and m");
    }
}

```

```

    System.exit(1);
}
T = genKey();
if (m.remove(KEY2OBJ(T)) != t.remove(KEY2OBJ(T))) {
    System.out.println("Error (" + seed + "): divergence in remove() between t and m");
    System.exit(1);
}
}

if (!m.equals(t)) System.out.println("Error (" + seed + "): !m.equals(t) after removal");
if (!t.equals(m)) System.out.println("Error (" + seed + "): !t.equals(m) after removal");

/* Now we check that m actually holds that data. */

for(java.util.Iterator i=t.iterator(); i.hasNext();) {
    Object e = i.next();
    if (!m.contains(e)) {
        System.out.println("Error (" + seed + "): m and t differ on a key (" + e + ") after removal (iterating on t)");
        System.exit(1);
    }
}

/* Now we check that m actually holds that data, but iterating on m. */

for(java.util.Iterator i=m.iterator(); i.hasNext();) {
    Object e = i.next();
    if (!t.contains(e)) {
        System.out.println("Error (" + seed + "): m and t differ on a key (" + e + ") after removal (iterating on m)");
        System.exit(1);
    }
}

printProbes(m);

/* Now we make m into an array, make it again a set and check it is OK. */
KEY_TYPE a[] = m.TO_KEY_ARRAY();

#ifdef Custom
if (!new OPEN_HASH_SET(a, m.strategy()).equals(m))
    System.out.println("Error (" + seed + "): toArray() output (or array-based constructor) is not OK");
#else
if (!new OPEN_HASH_SET(a).equals(m))
    System.out.println("Error (" + seed + "): toArray() output (or array-based constructor) is not OK");
#endif

/* Now we check cloning. */

ensure(m.equals(((OPEN_HASH_SET)m).clone()), "Error (" + seed + "): m does not equal m.clone()");

```

```

ensure(((OPEN_HASH_SET)m).clone().equals(m), "Error (" + seed + "): m.clone() does not equal m");

int h = m.hashCode();

/* Now we save and read m. */

try {
    java.io.File ff = new java.io.File("it.unimi.dsi.fastutil.test");
    java.io.OutputStream os = new java.io.FileOutputStream(ff);
    java.io.ObjectOutputStream oos = new java.io.ObjectOutputStream(os);

    oos.writeObject(m);
    oos.close();

    java.io.InputStream is = new java.io.FileInputStream(ff);
    java.io.ObjectInputStream ois = new java.io.ObjectInputStream(is);

    m = (OPEN_HASH_SET)ois.readObject();
    ois.close();
    ff.delete();
}
catch(Exception e) {
    e.printStackTrace();
    System.exit(1);
}

#if !KEY_CLASS_Reference
if (m.hashCode() != h) System.out.println("Error (" + seed + "): hashCode() changed after save/read");

printProbes(m);

/* Now we check that m actually holds that data, but iterating on m. */

for(java.util.Iterator i=m.iterator(); i.hasNext();) {
    Object e = i.next();
    if (!t.contains(e)) {
        System.out.println("Error (" + seed + "): m and t differ on a key (" + e + ") after save/read");
        System.exit(1);
    }
}
#else
m.clear();
m.addAll(t);
#endif

/* Now we put and remove random data in m and t, checking that the result is the same. */

for(int i=0; i<20*n; i++) {

```

```

KEY_TYPE T = genKey();
if (m.add(KEY2OBJ(T)) != t.add(KEY2OBJ(T))) {
    System.out.println("Error (" + seed + "): divergence in add() between t and m after save/read");
    System.exit(1);
}
T = genKey();
if (m.remove(KEY2OBJ(T)) != t.remove(KEY2OBJ(T))) {
    System.out.println("Error (" + seed + "): divergence in remove() between t and m after save/read");
    System.exit(1);
}
}

if (!m.equals(t)) System.out.println("Error (" + seed + "): !m.equals(t) after post-save/read removal");
if (!t.equals(m)) System.out.println("Error (" + seed + "): !t.equals(m) after post-save/read removal");

#ifdef Linked

/* Now we play with iterators, but only in the linked case. */

{
    java.util.ListIterator i, j;
    Object I, J;
    i = (java.util.ListIterator)m.iterator();
    j = new java.util.LinkedList(t).listIterator();

    for(int k = 0; k < 2*n; k++) {
        ensure(i.hasNext() == j.hasNext(), "Error (" + seed + "): divergence in hasNext()");
        ensure(i.hasPrevious() == j.hasPrevious(), "Error (" + seed + "): divergence in hasPrevious()");

        if (r.nextFloat() < .8 && i.hasNext()) {
#ifdef Custom
            ensure(m.strategy().equals(i.next(), J = j.next()), "Error (" + seed + "): divergence in next()");
#else
            ensure(i.next().equals(J = j.next()), "Error (" + seed + "): divergence in next()");
#endif
        }
        if (r.nextFloat() < 0.5) {
            i.remove();
            j.remove();
            t.remove(J);
        }
        else if (r.nextFloat() < .2 && i.hasPrevious()) {
#ifdef Custom
            ensure(m.strategy().equals(i.previous(), J = j.previous()), "Error (" + seed + "): divergence in previous()");
#else
            ensure(i.previous().equals(J = j.previous()), "Error (" + seed + "): divergence in previous()");
#endif
        }
    }
}

```

```

#endif
    if (r.nextFloat() < 0.5) {
        i.remove();
        j.remove();
        t.remove(J);
    }
}

ensure(i.nextIndex() == j.nextIndex(), "Error (" + seed + "): divergence in nextIndex()");
ensure(i.previousIndex() == j.previousIndex(), "Error (" + seed + "): divergence in previousIndex()");

}

}

if (t.size() > 0) {
    java.util.ListIterator i, j;
    Object J;
    j = new java.util.LinkedList(t).listIterator();
    int e = r.nextInt(t.size());
    Object from;
    do from = j.next(); while(e-- != 0);

    i = (java.util.ListIterator)m.iterator(KEY_OBJ2TYPE(from));

    for(int k = 0; k < 2*n; k++) {
        ensure(i.hasNext() == j.hasNext(), "Error (" + seed + "): divergence in hasNext() (iterator with starting point " +
from + ")");
        ensure(i.hasPrevious() == j.hasPrevious(), "Error (" + seed + "): divergence in hasPrevious() (iterator with starting
point " + from + ")");

        if (r.nextFloat() < .8 && i.hasNext()) {
#ifdef Custom
            ensure(m.strategy().equals(i.next(), J = j.next()), "Error (" + seed + "): divergence in next() (iterator with starting
point " + from + ")");
#else
            ensure(i.next().equals(J = j.next()), "Error (" + seed + "): divergence in next() (iterator with starting point " + from +
)");
#endif
        }
    }

    if (r.nextFloat() < 0.5) {
        i.remove();
        j.remove();
        t.remove(J);
    }
}

else if (r.nextFloat() < .2 && i.hasPrevious()) {
#ifdef Custom

```

```

    ensure(m.strategy().equals(i.previous(), J = j.previous()), "Error (" + seed + "): divergence in previous() (iterator
with starting point " + from + ")");
#else
    ensure(i.previous().equals(J = j.previous()), "Error (" + seed + "): divergence in previous() (iterator with starting
point " + from + ")");
#endif

    if (r.nextFloat() < 0.5) {
        i.remove();
        j.remove();
        t.remove(J);
    }
}

    ensure(i.nextIndex() == j.nextIndex(), "Error (" + seed + "): divergence in nextIndex() (iterator with starting point "
+ from + ")");
    ensure(i.previousIndex() == j.previousIndex(), "Error (" + seed + "): divergence in previousIndex() (iterator with
starting point " + from + ")");

}

}

/* Now we check that m actually holds that data. */

ensure(m.equals(t), "Error (" + seed + "): ! m.equals(t) after iteration");
ensure(t.equals(m), "Error (" + seed + "): ! t.equals(m) after iteration");

#endif

/* Now we take out of m everything, and check that it is empty. */

for(java.util.Iterator i=m.iterator(); i.hasNext();) { i.next(); i.remove();}

if (!m.isEmpty()) {
    System.out.println("Error (" + seed + "): m is not empty (as it should be)");
    System.exit(1);
}

#if KEY_CLASS_Integer || KEY_CLASS_Long
    m = new OPEN_HASH_SET(n, f);
    t.clear();
    int x;

/* Now we torture-test the hash table. This part is implemented only for integers and longs. */

```

```

int p = m.key.length - 1;

for(int i=0; i<p; i++) {
  for (int j=0; j<20; j++) {
    m.add(i+(r.nextInt() % 10)*p);
    m.remove(i+(r.nextInt() % 10)*p);
  }

  for (int j=-10; j<10; j++) m.remove(i+j*p);
}

t.addAll(m);

/* Now all table entries are REMOVED. */

int k = 0;
for(int i=0; i<(p*f)/10; i++) {
  for (int j=0; j<10; j++) {
    k++;
    x = i+(r.nextInt() % 10)*p;
    if (m.add(x) != t.add(KEY2OBJ(x)))
      System.out.println("Error (" + seed + "): m and t differ on a key during torture-test insertion.");
  }
}

if (!m.equals(t)) System.out.println("Error (" + seed + "): !m.equals(t) after torture-test insertion");
if (!t.equals(m)) System.out.println("Error (" + seed + "): !t.equals(m) after torture-test insertion");

for(int i=0; i<(p*f)/10; i++) {
  for (int j=0; j<10; j++) {
    x = i+(r.nextInt() % 10)*p;
    if (m.remove(x) != t.remove(KEY2OBJ(x)))
      System.out.println("Error (" + seed + "): m and t differ on a key during torture-test removal.");
  }
}

if (!m.equals(t)) System.out.println("Error (" + seed + "): !m.equals(t) after torture-test removal");
if (!t.equals(m)) System.out.println("Error (" + seed + "): !t.equals(m) after torture-test removal");

if (!m.equals(m.clone())) System.out.println("Error (" + seed + "): !m.equals(m.clone()) after torture-test removal");
if (!((OPEN_HASH_SET)m.clone()).equals(m)) System.out.println("Error (" + seed + "): !m.clone().equals(m) after
torture-test removal");

m.trim();

if (!m.equals(t)) System.out.println("Error (" + seed + "): !m.equals(t) after trim()");
if (!t.equals(m)) System.out.println("Error (" + seed + "): !t.equals(m) after trim()");
#endif

```

```

System.out.println("Test OK");
return;
#endif
}

public static void main(String args[]) {
float f = Hash.DEFAULT_LOAD_FACTOR;
int n = Integer.parseInt(args[1]);
if (args.length>2) f = Float.parseFloat(args[2]);
if (args.length > 3) r = new java.util.Random(seed = Long.parseLong(args[3]));

try {
if ("speedTest".equals(args[0]) || "speedComp".equals(args[0])) speedTest(n, f, "speedComp".equals(args[0]));
else if ("test".equals(args[0])) runTest(n, f);
} catch(Throwable e) {
e.printStackTrace(System.err);
System.err.println("seed: " + seed);
}
}

#endif

}

```

Found in path(s):

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/drv/OpenHashSet.drv

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2002-2017 Sebastiano Vigna

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

```

package PACKAGE;

import java.util.Set;

/** An abstract class providing basic methods for sets implementing a type-specific interface.
 *
 * <p>Note that the type-specific { @link Set } interface adds a type-specific { @code remove() }
 * method, as it is no longer harmful for subclasses. Thus, concrete subclasses of this class
 * must implement { @code remove() } (the { @code rem() } implementation of this
 * class just delegates to { @code remove() }).
 */

public abstract class ABSTRACT_SET KEY_GENERIC extends ABSTRACT_COLLECTION KEY_GENERIC
implements Cloneable, SET KEY_GENERIC {

    protected ABSTRACT_SET() {}

    @Override
    public abstract KEY_ITERATOR KEY_GENERIC iterator();

    @Override
    public boolean equals(final Object o) {
        if (o == this) return true;
        if (!(o instanceof Set)) return false;

        Set<?> s = (Set<?>) o;
        if (s.size() != size()) return false;
        return containsAll(s);
    }

    /** Returns a hash code for this set.
     *
     * The hash code of a set is computed by summing the hash codes of
     * its elements.
     *
     * @return a hash code for this set.
     */
    @Override
    public int hashCode() {
        int h = 0, n = size();
        KEY_ITERATOR KEY_GENERIC i = iterator();
        KEY_GENERIC_TYPE k;

        while(n-- != 0) {
            k = i.NEXT_KEY(); // We need k because KEY2JAVAHASH() is a macro with repeated evaluation.
            h += KEY2JAVAHASH(k);
        }
    }
}

```

```
return h;
}
```

```
#if KEYS_PRIMITIVE
```

```
/** {@inheritDoc}
```

```
 * Delegates to the type-specific {@code rem()} method
```

```
 * implemented by type-specific abstract {@link java.util.Collection} superclass.
```

```
 */
```

```
@Override
```

```
public boolean remove(KEY_TYPE k) {
```

```
    return super.rem(k);
```

```
}
```

```
/** {@inheritDoc}
```

```
 * Delegates to the type-specific {@code remove()} method
```

```
 * specified in the type-specific {@link Set} interface.
```

```
 * @deprecated Please use {@code remove()} instead.
```

```
 */
```

```
@Deprecated
```

```
@Override
```

```
public boolean rem(KEY_TYPE k) {
```

```
    return remove(k);
```

```
}
```

```
#endif
```

```
}
```

Found in path(s):

```
*/opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-
a775574/drv/AbstractSet.drv
```

No license file was found, but licenses were detected in source scan.

```
/*
```

```
 * Copyright (C) 2002-2017 Sebastiano Vigna
```

```
 *
```

```
 * Licensed under the Apache License, Version 2.0 (the "License");
```

```
 * you may not use this file except in compliance with the License.
```

```
 * You may obtain a copy of the License at
```

```
 *
```

```
 * http://www.apache.org/licenses/LICENSE-2.0
```

```
 *
```

```
 * Unless required by applicable law or agreed to in writing, software
```

```
 * distributed under the License is distributed on an "AS IS" BASIS,
```

```
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
```

```
 * See the License for the specific language governing permissions and
```

```
 * limitations under the License.
```

```
 */
```

```

package PACKAGE;

import java.util.Set;

/** A type-specific {@link Set}; provides some additional methods that use polymorphism to avoid (un)boxing.
 *
 * <p>Additionally, this interface strengthens (again) {@link #iterator()}.
 *
 * @see Set
 */

public interface SET KEY_GENERIC extends COLLECTION KEY_GENERIC, Set<KEY_GENERIC_CLASS> {

    /** Returns a type-specific iterator on the elements of this set.
     *
     * <p>Note that this specification strengthens the one given in {@link java.lang.Iterable#iterator()},
     * which was already strengthened in the corresponding type-specific class,
     * but was weakened by the fact that this interface extends {@link Set}.
     *
     * @return a type-specific iterator on the elements of this set.
     */
    @Override
    KEY_ITERATOR KEY_GENERIC iterator();

    #if KEYS_PRIMITIVE
    /** Removes an element from this set.
     *
     * <p>Note that the corresponding method of a type-specific collection is {@code rem()}.
     * This unfortunate situation is caused by the clash
     * with the similarly named index-based method in the {@link java.util.List} interface.
     *
     * @see java.util.Collection#remove(Object)
     */
    boolean remove(KEY_TYPE k);

    /** {@inheritDoc}
     * @deprecated Please use the corresponding type-specific method instead.
     */
    @SuppressWarnings("deprecation")
    @Deprecated
    @Override
    default boolean remove(final Object o) {
        return COLLECTION.super.remove(o);
    }

    /** {@inheritDoc}

```

```

* @deprecated Please use the corresponding type-specific method instead.
*/
@SuppressWarnings("deprecation")
@Deprecated
@Override
default boolean add(final KEY_GENERIC_CLASS o) {
    return COLLECTION.super.add(o);
}

/** { @inheritDoc}
* @deprecated Please use the corresponding type-specific method instead.
*/
@SuppressWarnings("deprecation")
@Deprecated
@Override
default boolean contains(final Object o) {
    return COLLECTION.super.contains(o);
}

/** Removes an element from this set.
*
* <p>This method is inherited from the type-specific collection this
* type-specific set is based on, but it should not used as
* this interface reinstates { @code remove()} as removal method.
*
* @deprecated Please use { @code remove()} instead.
*/
@Deprecated
@Override
default boolean rem(KEY_TYPE k) {
    return remove(k);
}

#endif
}

```

Found in path(s):

```

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-
a775574/drv/Set.drv

```

No license file was found, but licenses were detected in source scan.

```

/*
* Copyright (C) 2003-2017 Paolo Boldi and Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*

```

```
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

```
package PACKAGE;
```

```
#if KEY_CLASS_Object
import java.util.Comparator;
#endif
```

```
/** A class providing static methods and objects that do useful things with heaps.
 *
 * <p>The static methods of this class allow to treat arrays as 0-based heaps. They
 * are used in the implementation of heap-based queues, but they may be also used
 * directly.
 *
 */
```

```
public final class HEAPS {
```

```
    private HEAPS() {}
```

```
    /** Moves the given element down into the heap until it reaches the lowest possible position.
     *
     * @param heap the heap (starting at 0).
     * @param size the number of elements in the heap.
     * @param i the index of the element that must be moved down.
     * @param c a type-specific comparator, or { @code null } for the natural order.
     * @return the new position of the element of index { @code i }.
     */
```

```
    SUPPRESS_WARNINGS_KEY_UNCHECKED
```

```
    public static KEY_GENERIC int downHeap(final KEY_GENERIC_TYPE[] heap, final int size, int i, final
    KEY_COMPARATOR KEY_SUPER_GENERIC c) {
```

```
        assert i < size;
```

```
        final KEY_GENERIC_TYPE e = heap[i];
```

```
        int child;
```

```
        if (c == null)
```

```
            while ((child = (i << 1) + 1) < size) {
```

```
                KEY_GENERIC_TYPE t = heap[child];
```

```

    final int right = child + 1;
    if (right < size && KEY_LESS(heap[right], t)) t = heap[child = right];
    if (KEY_LESSEQ(e, t)) break;
    heap[i] = t;
    i = child;
}
else
while ((child = (i << 1) + 1) < size) {
    KEY_GENERIC_TYPE t = heap[child];
    final int right = child + 1;
    if (right < size && c.compare(heap[right], t) < 0) t = heap[child = right];
    if (c.compare(e, t) <= 0) break;
    heap[i] = t;
    i = child;
}

```

```

heap[i] = e;

```

```

return i;
}

```

```

/** Moves the given element up in the heap until it reaches the highest possible position.

```

```

*
* @param heap the heap (starting at 0).
* @param size the number of elements in the heap.
* @param i the index of the element that must be moved up.
* @param c a type-specific comparator, or { @code null } for the natural order.
* @return the new position of the element of index { @code i }.
*/

```

```

SUPPRESS_WARNINGS_KEY_UNCHECKED

```

```

public static KEY_GENERIC int upHeap(final KEY_GENERIC_TYPE[] heap, final int size, int i, final
KEY_COMPARATOR KEY_GENERIC c) {

```

```

    assert i < size;

```

```

    final KEY_GENERIC_TYPE e = heap[i];

```

```

    if (c == null)
        while (i != 0) {
            final int parent = (i - 1) >>> 1;
            final KEY_GENERIC_TYPE t = heap[parent];
            if (KEY_LESSEQ(t, e)) break;
            heap[i] = t;
            i = parent;
        }
    else

```

```

        while (i != 0) {
            final int parent = (i - 1) >>> 1;

```

```

    final KEY_GENERIC_TYPE t = heap[parent];
    if (c.compare(t, e) <= 0) break;
    heap[i] = t;
    i = parent;
}

heap[i] = e;

return i;
}

/** Makes an array into a heap.
 *
 * @param heap the heap (starting at 0).
 * @param size the number of elements in the heap.
 * @param c a type-specific comparator, or { @code null } for the natural order.
 */

public static KEY_GENERIC void makeHeap(final KEY_GENERIC_TYPE[] heap, final int size, final
KEY_COMPARATOR KEY_GENERIC c) {
    int i = size >>> 1;
    while(i-- != 0) downHeap(heap, size, i, c);
}
}

```

Found in path(s):

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/drv/Heaps.drv

No license file was found, but licenses were detected in source scan.

```

/*
 * Copyright (C) 2003-2017 Paolo Boldi and Sebastiano Vigna
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

```

package PACKAGE;

```

#if KEY_CLASS_Object
import java.util.Arrays;
import java.util.Comparator;
import it.unimi.dsi.fastutil.PriorityQueue;
#endif

import java.util.NoSuchElementException;

/** A type-specific array-based priority queue.
 *
 * <p>Instances of this class represent a priority queue using a backing
 * array&mdash;all operations are performed directly on the array. The array is
 * enlarged as needed, but it is never shrunk. Use the {@link #trim()} method
 * to reduce its size, if necessary.
 *
 * <p>This implementation is extremely inefficient, but it is difficult to beat
 * when the size of the queue is very small.
 */

public class ARRAY_PRIORITY_QUEUE KEY_GENERIC implements PRIORITY_QUEUE KEY_GENERIC,
java.io.Serializable {
    private static final long serialVersionUID = 1L;

    /** The backing array. */
    SUPPRESS_WARNINGS_KEY_UNCHECKED
    protected transient KEY_GENERIC_TYPE array[] = KEY_GENERIC_ARRAY_CAST
    ARRAYS.EMPTY_ARRAY;

    /** The number of elements in this queue. */
    protected int size;

    /** The type-specific comparator used in this queue. */
    protected KEY_COMPARATOR KEY_SUPER_GENERIC c;

    /** The first index, cached, if {@link #firstIndexValid} is true. */
    protected transient int firstIndex;

    /** Whether {@link #firstIndex} contains a valid value. */
    protected transient boolean firstIndexValid;

    /** Creates a new empty queue with a given capacity and comparator.
 *
 * @param capacity the initial capacity of this queue.
 * @param c the comparator used in this queue, or {@code null} for the natural order.
 */
    SUPPRESS_WARNINGS_KEY_UNCHECKED
    public ARRAY_PRIORITY_QUEUE(int capacity, KEY_COMPARATOR KEY_SUPER_GENERIC c) {

```

```

if (capacity > 0) this.array = KEY_GENERIC_ARRAY_CAST new KEY_TYPE[capacity];
this.c = c;
}

/** Creates a new empty queue with a given capacity and using the natural order.
 *
 * @param capacity the initial capacity of this queue.
 */
public ARRAY_PRIORITY_QUEUE(int capacity) {
    this(capacity, null);
}

/** Creates a new empty queue with a given comparator.
 *
 * @param c the comparator used in this queue, or { @code null } for the natural order.
 */
public ARRAY_PRIORITY_QUEUE(KEY_COMPARATOR KEY_SUPER_GENERIC c) {
    this(0, c);
}

/** Creates a new empty queue using the natural order.
 */
public ARRAY_PRIORITY_QUEUE() {
    this(0, null);
}

/** Wraps a given array in a queue using a given comparator.
 *
 * <p>The queue returned by this method will be backed by the given array.
 *
 * @param a an array.
 * @param size the number of elements to be included in the queue.
 * @param c the comparator used in this queue, or { @code null } for the natural order.
 */
public ARRAY_PRIORITY_QUEUE(final KEY_GENERIC_TYPE[] a, int size, final KEY_COMPARATOR
KEY_SUPER_GENERIC c) {
    this(c);
    this.array = a;
    this.size = size;
}

/** Wraps a given array in a queue using a given comparator.
 *
 * <p>The queue returned by this method will be backed by the given array.
 *
 * @param a an array.
 * @param c the comparator used in this queue, or { @code null } for the natural order.

```

```

*/
public ARRAY_PRIORITY_QUEUE(final KEY_GENERIC_TYPE[] a, final KEY_COMPARATOR
KEY_SUPER_GENERIC c) {
    this(a, a.length, c);
}

```

```

/** Wraps a given array in a queue using the natural order.
*
* <p>The queue returned by this method will be backed by the given array.
*
* @param a an array.
* @param size the number of elements to be included in the queue.
*/

```

```

public ARRAY_PRIORITY_QUEUE(final KEY_GENERIC_TYPE[] a, int size) {
    this(a, size, null);
}

```

```

/** Wraps a given array in a queue using the natural order.
*
* <p>The queue returned by this method will be backed by the given array.
*
* @param a an array.
*/

```

```

public ARRAY_PRIORITY_QUEUE(final KEY_GENERIC_TYPE[] a) {
    this(a, a.length);
}

```

```

/** Returns the index of the smallest element. */

```

```

SUPPRESS_WARNINGS_KEY_UNCHECKED

```

```

private int findFirst() {
    if (firstIndexValid) return this.firstIndex;
    firstIndexValid = true;
    int i = size;
    int firstIndex = --i;
    KEY_GENERIC_TYPE first = array[firstIndex];

    if (c == null) { while(i-- != 0) if (KEY_LESS(array[i], first)) first = array[firstIndex = i]; }
    else while(i-- != 0) { if (c.compare(array[i], first) < 0) first = array[firstIndex = i]; }

    return this.firstIndex = firstIndex;
}

```

```

private void ensureNonEmpty() {
    if (size == 0) throw new NoSuchElementException();
}

```

```

}

@Override
SUPPRESS_WARNINGS_KEY_UNCHECKED
public void enqueue(KEY_GENERIC_TYPE x) {
    if (size == array.length) array = ARRAYS.grow(array, size + 1);
    if (firstIndexValid) {
        if (c == null) { if (KEY_LESS(x, array[firstIndex])) firstIndex = size; }
        else if (c.compare(x, array[firstIndex]) < 0) firstIndex = size;
    }
    else firstIndexValid = false;
    array[size++] = x;
}

```

```

@Override
public KEY_GENERIC_TYPE DEQUEUE() {
    ensureNonEmpty();
    final int first = findFirst();
    final KEY_GENERIC_TYPE result = array[first];
    System.arraycopy(array, first + 1, array, first, --size - first);
    #if KEY_CLASS_Object
    array[size] = null;
    #endif
    firstIndexValid = false;
    return result;
}

```

```

@Override
public KEY_GENERIC_TYPE FIRST() {
    ensureNonEmpty();
    return array[findFirst()];
}

```

```

@Override
public void changed() {
    ensureNonEmpty();
    firstIndexValid = false;
}

```

```

@Override
public int size() { return size; }

```

```

@Override
public void clear() {
    #if KEY_CLASS_Object
    Arrays.fill(array, 0, size, null);
    #endif
    size = 0;
}

```

```

    firstIndexValid = false;
}

/** Trims the underlying array so that it has exactly {@link #size()} elements. */

public void trim() {
    array = ARRAYS.trim(array, size);
}

@Override
public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() { return c; }

private void writeObject(java.io.ObjectOutputStream s) throws java.io.IOException {
    s.defaultWriteObject();
    s.writeInt(array.length);
    for(int i = 0; i < size; i++) s.WRITE_KEY(array[i]);
}

SUPPRESS_WARNINGS_KEY_UNCHECKED
private void readObject(java.io.ObjectInputStream s) throws java.io.IOException, ClassNotFoundException {
    s.defaultReadObject();
    array = KEY_GENERIC_ARRAY_CAST new KEY_TYPE[s.readInt()];
    for(int i = 0; i < size; i++) array[i] = KEY_GENERIC_CAST s.READ_KEY();
}
}
}

```

Found in path(s):

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/drv/ArrayPriorityQueue.drv

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2003-2017 Paolo Boldi and Sebastiano Vigna

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

```

package PACKAGE;

import java.util.Comparator;

/** A class providing static methods and objects that do useful things with comparators.
 */

public final class COMPARATORS {

    private COMPARATORS() {}

    /** A type-specific comparator mimicking the natural order. */
    #if KEYS_REFERENCE
    SUPPRESS_WARNINGS_KEY_UNCHECKED_RAWTYPES
    protected static class NaturalImplicitComparator implements Comparator, java.io.Serializable {
    #else
    protected static class NaturalImplicitComparator implements KEY_COMPARATOR KEY_GENERIC,
    java.io.Serializable {
    #endif
        private static final long serialVersionUID = 1L;

        @Override
        public final int compare(final KEY_TYPE a, final KEY_TYPE b) {
    #if KEYS_PRIMITIVE
            return KEY_CMP(a, b);
    #else
            return ((Comparable)a).compareTo(b);
    #endif
        }
        private Object readResolve() { return NATURAL_COMPARATOR; }
    };

    SUPPRESS_WARNINGS_KEY_RAWTYPES
    public static final KEY_COMPARATOR NATURAL_COMPARATOR = new NaturalImplicitComparator();

    /** A type-specific comparator mimicking the opposite of the natural order. */
    #if KEYS_REFERENCE
    SUPPRESS_WARNINGS_KEY_UNCHECKED_RAWTYPES
    protected static class OppositeImplicitComparator implements Comparator, java.io.Serializable {
    #else
    protected static class OppositeImplicitComparator implements KEY_COMPARATOR KEY_GENERIC,
    java.io.Serializable {
    #endif
        private static final long serialVersionUID = 1L;

        @Override
        public final int compare(final KEY_TYPE a, final KEY_TYPE b) {
    #if KEYS_PRIMITIVE

```

```

    return - KEY_CMP(a, b);
#else
    return ((Comparable)b).compareTo(a);
#endif
}
private Object readResolve() { return OPPOSITE_COMPARATOR; }
};

SUPPRESS_WARNINGS_KEY_RAWTYPES
public static final KEY_COMPARATOR OPPOSITE_COMPARATOR = new OppositeImplicitComparator();

#if KEYS_REFERENCE
protected static class OppositeComparator KEY_GENERIC implements Comparator KEY_GENERIC,
java.io.Serializable {
#else
protected static class OppositeComparator KEY_GENERIC implements KEY_COMPARATOR KEY_GENERIC,
java.io.Serializable {
#endif
private static final long serialVersionUID = 1L;

private final KEY_COMPARATOR KEY_GENERIC comparator;

protected OppositeComparator(final KEY_COMPARATOR KEY_GENERIC c) { comparator = c; }

@Override
public final int compare(final KEY_GENERIC_TYPE a, final KEY_GENERIC_TYPE b) { return
comparator.compare(b, a); }
};

/** Returns a comparator representing the opposite order of the given comparator.
 *
 * @param c a comparator.
 * @return a comparator representing the opposite order of { @code c}.
 */
public static KEY_GENERIC KEY_COMPARATOR KEY_GENERIC oppositeComparator(final
KEY_COMPARATOR KEY_GENERIC c) { return new OppositeComparator KEY_GENERIC_DIAMOND(c); }

#if KEYS_PRIMITIVE
/** Returns a type-specific comparator that is equivalent to the given comparator.
 *
 * @param c a comparator, or { @code null}.
 * @return a type-specific comparator representing the order of { @code c}.
 */
public static KEY_COMPARATOR AS_KEY_COMPARATOR(final Comparator<? super
KEY_GENERIC_CLASS> c) {
if (c == null || c instanceof KEY_COMPARATOR) return (KEY_COMPARATOR) c;

return new KEY_COMPARATOR() {

```

```

    @Override
    public int compare(KEY_GENERIC_TYPE x, KEY_GENERIC_TYPE y) { return c.compare(KEY2OBJ(x),
KEY2OBJ(y)); }
    @SuppressWarnings("deprecation")
    @Override
    public int compare(KEY_GENERIC_CLASS x, KEY_GENERIC_CLASS y) { return c.compare(x, y); }
};
}
#endif
}

```

Found in path(s):

```
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-
a775574/drv/Comparators.drv
```

No license file was found, but licenses were detected in source scan.

```

/*
 * Copyright (C) 2003-2017 Paolo Boldi and Sebastiano Vigna
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

```

```
package PACKAGE;
```

```

#if KEY_CLASS_Object
import java.util.Comparator;
#endif

```

```
import it.unimi.dsi.fastutil.ints.IntArrays;
```

```

import java.util.Arrays;
import java.util.NoSuchElementException;

```

```
/** A type-specific heap-based indirect priority queue.
```

```
*
```

```
* <p>Instances of this class use an additional <em>inversion array</em>, of the same length of the reference array,
```

```
* to keep track of the heap position containing a given element of the reference array. The priority queue is
```

```

* represented using a heap. The heap is enlarged as needed, but it is never
* shrunk. Use the {@link #trim()} method to reduce its size, if necessary.
*
* <p>This implementation does <em>not</em> allow one to enqueue several times the same index.
*/

```

```

public class HEAP_INDIRECT_PRIORITY_QUEUE KEY_GENERIC extends
HEAP_SEMI_INDIRECT_PRIORITY_QUEUE KEY_GENERIC {

```

```

/** The inversion array. */
protected final int inv[];

```

```

/** Creates a new empty queue with a given capacity and comparator.
*
* @param refArray the reference array.
* @param capacity the initial capacity of this queue.
* @param c the comparator used in this queue, or {@code null} for the natural order.
*/

```

```

public HEAP_INDIRECT_PRIORITY_QUEUE(KEY_GENERIC_TYPE[] refArray, int capacity,
KEY_COMPARATOR KEY_SUPER_GENERIC c) {
    super(refArray, capacity, c);
    if (capacity > 0) this.heap = new int[capacity];

```

```

    this.c = c;

```

```

    this.inv = new int[refArray.length];
    Arrays.fill(inv, -1);
}

```

```

/** Creates a new empty queue with a given capacity and using the natural order.
*
* @param refArray the reference array.
* @param capacity the initial capacity of this queue.
*/

```

```

public HEAP_INDIRECT_PRIORITY_QUEUE(KEY_GENERIC_TYPE[] refArray, int capacity) {
    this(refArray, capacity, null);
}

```

```

/** Creates a new empty queue with capacity equal to the length of the reference array and a given comparator.
*
* @param refArray the reference array.
* @param c the comparator used in this queue, or {@code null} for the natural order.
*/

```

```

public HEAP_INDIRECT_PRIORITY_QUEUE(KEY_GENERIC_TYPE[] refArray, KEY_COMPARATOR
KEY_SUPER_GENERIC c) {
    this(refArray, refArray.length, c);
}

```

```

/** Creates a new empty queue with capacity equal to the length of the reference array and using the natural order.
 * @param refArray the reference array.
 */
public HEAP_INDIRECT_PRIORITY_QUEUE(KEY_GENERIC_TYPE[] refArray) {
    this(refArray, refArray.length, null);
}

/** Wraps a given array in a queue using a given comparator.
 *
 * <p>The queue returned by this method will be backed by the given array.
 * The first { @code size } element of the array will be rearranged so to form a heap (this is
 * more efficient than enqueueing the elements of { @code a } one by one).
 *
 * @param refArray the reference array.
 * @param a an array of indices into { @code refArray }.
 * @param size the number of elements to be included in the queue.
 * @param c the comparator used in this queue, or { @code null } for the natural order.
 */
public HEAP_INDIRECT_PRIORITY_QUEUE(final KEY_GENERIC_TYPE[] refArray, final int[] a, final int
size, final KEY_COMPARATOR KEY_SUPER_GENERIC c) {
    this(refArray, 0, c);
    this.heap = a;
    this.size = size;
    int i = size;
    while(i-- != 0) {
        if (inv[a[i]] != -1) throw new IllegalArgumentException("Index " + a[i] + " appears twice in the heap");
        inv[a[i]] = i;
    }
    INDIRECT_HEAPPS.makeHeap(refArray, a, inv, size, c);
}

/** Wraps a given array in a queue using a given comparator.
 *
 * <p>The queue returned by this method will be backed by the given array.
 * The elements of the array will be rearranged so to form a heap (this is
 * more efficient than enqueueing the elements of { @code a } one by one).
 *
 * @param refArray the reference array.
 * @param a an array of indices into { @code refArray }.
 * @param c the comparator used in this queue, or { @code null } for the natural order.
 */
public HEAP_INDIRECT_PRIORITY_QUEUE(final KEY_GENERIC_TYPE[] refArray, final int[] a, final
KEY_COMPARATOR KEY_SUPER_GENERIC c) {
    this(refArray, a, a.length, c);
}

```

```

/** Wraps a given array in a queue using the natural order.
 *
 * <p>The queue returned by this method will be backed by the given array.
 * The first { @code size } element of the array will be rearranged so to form a heap (this is
 * more efficient than enqueueing the elements of { @code a } one by one).
 *
 * @param refArray the reference array.
 * @param a an array of indices into { @code refArray }.
 * @param size the number of elements to be included in the queue.
 */
public HEAP_INDIRECT_PRIORITY_QUEUE(final KEY_GENERIC_TYPE[] refArray, final int[] a, int size) {
    this(refArray, a, size, null);
}

```

```

/** Wraps a given array in a queue using the natural order.
 *
 * <p>The queue returned by this method will be backed by the given array.
 * The elements of the array will be rearranged so to form a heap (this is
 * more efficient than enqueueing the elements of { @code a } one by one).
 *
 * @param refArray the reference array.
 * @param a an array of indices into { @code refArray }.
 */
public HEAP_INDIRECT_PRIORITY_QUEUE(final KEY_GENERIC_TYPE[] refArray, final int[] a) {
    this(refArray, a, a.length);
}

```

```

@Override
public void enqueue(final int x) {
    if (inv[x] >= 0) throw new IllegalArgumentException("Index " + x + " belongs to the queue");

    if (size == heap.length) heap = IntArrays.grow(heap, size + 1);

    inv[heap[size] = x] = size++;

    INDIRECT_HEAPS.upHeap(refArray, heap, inv, size, size - 1, c);
}

```

```

@Override
public boolean contains(final int index) {
    return inv[index] >= 0;
}

```

```

@Override
public int dequeue() {
    if (size == 0) throw new NoSuchElementException();
    final int result = heap[0];

```

```

if (--size != 0) inv[heap[0] = heap[size]] = 0;
inv[result] = -1;

if (size != 0) INDIRECT_HEAPS.downHeap(refArray, heap, inv, size, 0, c);
return result;
}

@Override
public void changed() {
    INDIRECT_HEAPS.downHeap(refArray, heap, inv, size, 0, c);
}

@Override
public void changed(final int index) {
    final int pos = inv[index];
    if (pos < 0) throw new IllegalArgumentException("Index " + index + " does not belong to the queue");
    final int newPos = INDIRECT_HEAPS.upHeap(refArray, heap, inv, size, pos, c);
    INDIRECT_HEAPS.downHeap(refArray, heap, inv, size, newPos, c);
}

/** Rebuilds this queue in a bottom-up fashion (in linear time). */
@Override
public void allChanged() { INDIRECT_HEAPS.makeHeap(refArray, heap, inv, size, c); }

@Override
public boolean remove(final int index) {
    final int result = inv[index];
    if (result < 0) return false;
    inv[index] = -1;

    if (result < --size) {
        inv[heap[result] = heap[size]] = result;
        final int newPos = INDIRECT_HEAPS.upHeap(refArray, heap, inv, size, result, c);
        INDIRECT_HEAPS.downHeap(refArray, heap, inv, size, newPos, c);
    }

    return true;
}

@Override
public void clear() {
    size = 0;
    Arrays.fill(inv, -1);
}

#ifdef TEST

/** The original class, now just used for testing. */

```

```

private static class TestQueue {

    /** The reference array */
    private KEY_TYPE refArray[];
    /** Its length */
    private int N;
    /** The number of elements in the heaps */
    private int n;
    /** The two comparators */
    private KEY_COMPARATOR primaryComp, secondaryComp;
    /** Two indirect heaps are used, called { @code primary } and { @code secondary }. Each of them contains
    a permutation of { @code n } among the indices 0, 1, ..., { @code N }-1 in such a way that the corresponding
    objects be sorted with respect to the two comparators.
    We also need an array { @code inSec[] } so that { @code inSec[k] } is the index of { @code secondary }
    containing { @code k }.
    */
    private int primary[], secondary[], inSec[];

    /** Builds a double indirect priority queue.
    * @param refArray The reference array.
    * @param primaryComp The primary comparator.
    * @param secondaryComp The secondary comparator.
    */
    public TestQueue(KEY_TYPE refArray[], KEY_COMPARATOR primaryComp, KEY_COMPARATOR
secondaryComp) {
        this.refArray = refArray;
        this.N = refArray.length;
        assert this.N != 0;
        this.n = 0;
        this.primaryComp = primaryComp;
        this.secondaryComp = secondaryComp;
        this.primary = new int[N];
        this.secondary = new int[N];
        this.inSec = new int[N];
        java.util.Arrays.fill(inSec, -1);
    }

    /** Adds an index to the queue. Notice that the index should not be already present in the queue.
    * @param i The index to be added
    */
    public void add(int i) {
        if (i < 0 || i >= refArray.length) throw new IndexOutOfBoundsException();
        if (inSec[i] >= 0) throw new IllegalArgumentException();
        primary[n] = i;
        secondary[n] = i; inSec[i] = n;
        n++;
        swimPrimary(n-1);
    }
}

```

```

swimSecondary(n-1);
}

/** Heapify the primary heap.
 * @param i The index of the heap to be heapified.
 */
private void heapifyPrimary(int i) {
    int dep = primary[i];
    int child;

    while ((child = 2*i+1) < n) {
        if (child+1 < n && primaryComp.compare(refArray[primary[child+1]], refArray[primary[child]]) < 0) child++;
        if (primaryComp.compare(refArray[dep], refArray[primary[child]]) <= 0) break;
        primary[i] = primary[child];
        i = child;
    }
    primary[i] = dep;
}

/** Heapify the secondary heap.
 * @param i The index of the heap to be heapified.
 */
private void heapifySecondary(int i) {
    int dep = secondary[i];
    int child;

    while ((child = 2*i+1) < n) {
        if (child+1 < n && secondaryComp.compare(refArray[secondary[child+1]], refArray[secondary[child]]) < 0)
child++;
        if (secondaryComp.compare(refArray[dep], refArray[secondary[child]]) <= 0) break;
        secondary[i] = secondary[child]; inSec[secondary[i]] = i;
        i = child;
    }
    secondary[i] = dep; inSec[secondary[i]] = i;
}

/** Swim and heapify the primary heap.
 * @param i The index to be moved.
 */
private void swimPrimary(int i) {
    int dep = primary[i];
    int parent;

    while (i != 0 && (parent = (i - 1) / 2) >= 0) {
        if (primaryComp.compare(refArray[primary[parent]], refArray[dep]) <= 0) break;
        primary[i] = primary[parent];
        i = parent;
    }
}

```

```

primary[i] = dep;
heapifyPrimary(i);
}

/** Swim and heapify the secondary heap.
 * @param i The index to be moved.
 */
private void swimSecondary(int i) {
    int dep = secondary[i];
    int parent;

    while (i != 0 && (parent = (i - 1) / 2) >= 0) {
        if (secondaryComp.compare(refArray[secondary[parent]], refArray[dep]) <= 0) break;
        secondary[i] = secondary[parent]; inSec[secondary[i]] = i;
        i = parent;
    }
    secondary[i] = dep; inSec[secondary[i]] = i;
    heapifySecondary(i);
}

/** Returns the minimum element with respect to the primary comparator.
 * @return the minimum element.
 */
public int top() {
    if (n == 0) throw new java.util.NoSuchElementException();
    return primary[0];
}

/** Returns the minimum element with respect to the secondary comparator.
 * @return the minimum element.
 */
public int secTop() {
    if (n == 0) throw new java.util.NoSuchElementException();
    return secondary[0];
}

/** Removes the minimum element with respect to the primary comparator.
 * @return the removed element.
 */
public boolean remove() {
    if (n == 0) throw new java.util.NoSuchElementException();
    if (inSec[primary[0]] == -1) return false;
    int result = primary[0];
    int ins = inSec[result];
    inSec[result] = -1;
    // Copy a leaf
    primary[0] = primary[n-1];
    if (ins == n-1) {

```

```

    n--;
    heapifyPrimary(0);
    return true;
}
secondary[ins] = secondary[n-1];
inSec[secondary[ins]] = ins;
// Heapify
n--;
heapifyPrimary(0);
swimSecondary(ins);
return true;
}

public void clear() {
    while(size() != 0) remove();
}

public void remove(int index) {
    if (index >= refArray.length) throw new IndexOutOfBoundsException();
    if (inSec[index] == -1) return;
    int ins = inSec[index];
    inSec[index] = -1;
    // Copy a leaf
    primary[ins] = primary[n-1];
    if (ins == n-1) {
        n--;
        swimPrimary(ins);
        return;
    }
    secondary[ins] = secondary[n-1];
    inSec[secondary[ins]] = ins;
    // Heapify
    n--;
    swimPrimary(ins);
    swimSecondary(ins);
}

/** Signals that the minimum element with respect to the comparator has changed.
 */
public void change() {
    if (n == 0) throw new java.util.NoSuchElementException();
    if (inSec[primary[0]] == -1) throw new IllegalArgumentException();
    int ins = inSec[primary[0]];
    heapifyPrimary(0);
    swimSecondary(ins);
}

public void change(int index) {

```

```

if (index >= refArray.length) throw new IndexOutOfBoundsException();
if (inSec[index] == -1) throw new IllegalArgumentException();
if (n == 0) throw new java.util.NoSuchElementException();
int ins = inSec[index];
swimPrimary(ins);
swimSecondary(ins);
}

/** Returns the number of elements in the queue.
 * @return the size of the queue
 */
public int size() {
    return n;
}

public String toString() {
    String s = "[";
    for (int i = 0; i < n; i++)
        s += refArray[primary[i]]+", ";
    return s+ "]";
}

private static long seed = System.currentTimeMillis();
private static java.util.Random r = new java.util.Random(seed);

private static KEY_TYPE genKey() {
    #if KEY_CLASS_Byte || KEY_CLASS_Short || KEY_CLASS_Character
        return (KEY_TYPE)(r.nextInt());
    #elif KEYS_PRIMITIVE
        return r.NEXT_KEY();
    #elif KEY_CLASS_Object
        return Integer.toBinaryString(r.nextInt());
    #else
        return new java.io.Serializable() {};
    #endif
}

private static java.text.NumberFormat format = new java.text.DecimalFormat("#,###.00");
private static java.text.FieldPosition p = new java.text.FieldPosition(0);

private static String format(double d) {
    StringBuffer s = new StringBuffer();
    return format.format(d, s, p).toString();
}

```

```

private static void speedTest(int n, boolean comp) {
    System.out.println("There are presently no speed tests for this class.");
}

private static void fatal(String msg) {
    System.out.println(msg);
    System.exit(1);
}

private static void ensure(boolean cond, String msg) {
    if (cond) return;
    fatal(msg);
}

private static boolean heapEqual(int[] a, int[] b, int sizea, int sizeb) {
    if (sizea != sizeb) return false;
    while(sizea-- != 0) if (a[sizea] != b[sizea]) return false;
    return true;
}

private static boolean invEqual(int inva[], int[] invb) {
    int i = inva.length;
    while(i-- != 0) if (inva[i] != invb[i]) return false;
    return true;
}

protected static void runTest(int n) {
    long ms;
    Exception mThrowsIllegal, tThrowsIllegal, mThrowsOutOfBounds, tThrowsOutOfBounds, mThrowsNoElement,
    tThrowsNoElement;
    int rm = 0, rt = 0;
    KEY_TYPE[] refArray = new KEY_TYPE[n];

    for(int i = 0; i < n; i++) refArray[i] = genKey();

    HEAP_INDIRECT_PRIORITY_QUEUE m = new HEAP_INDIRECT_PRIORITY_QUEUE(refArray,
    COMPARATORS.NATURAL_COMPARATOR);
    TestQueue t = new TestQueue(refArray, COMPARATORS.NATURAL_COMPARATOR,
    COMPARATORS.NATURAL_COMPARATOR);

    /* We add pairs to t. */
    for(int i = 0; i < n / 2; i++) {
        t.add(i);
        m.enqueue(i);
    }

    ensure(heapEqual(m.heap, t.primary, m.size(), t.size()), "Error (" + seed + "): m and t differ after creation (" + m + ",

```

```

" + t + ")");
    ensure(invEqual(m.inv, t.inSec), "Error (" + seed + "): m and t differ in inversion arrays after creation (" +
java.util.Arrays.toString(m.inv) + ", " + java.util.Arrays.toString(t.inSec) + ")");

    /* Now we add and remove random data in m and t, checking that the result is the same. */

    for(int i=0; i<2*n; i++) {
        if (r.nextDouble() < 0.01) {
            t.clear();
            m.clear();
            for(int j = 0; j < n / 2; j++) {
                t.add(j);
                m.enqueue(j);
            }
        }
    }

    int T = r.nextInt(2 * n);

    mThrowsNoElement = tThrowsNoElement = mThrowsOutOfBounds = tThrowsOutOfBounds = mThrowsIllegal =
tThrowsIllegal = null;

    try {
        m.enqueue(T);
    }
    catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }
    catch (IllegalArgumentException e) { mThrowsIllegal = e; }

    try {
        t.add(T);
    }
    catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }
    catch (IllegalArgumentException e) { tThrowsIllegal = e; }

    ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + seed + "): enqueue()
divergence in IndexOutOfBoundsException for " + T + " (" + mThrowsOutOfBounds + ", " + tThrowsOutOfBounds
+ ")");
    ensure((mThrowsIllegal == null) == (tThrowsIllegal == null), "Error (" + seed + "): enqueue() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ")");

    ensure(heapEqual(m.heap, t.primary, m.size(), t.size()), "Error (" + seed + "): m and t differ after enqueue (" + m +
", " + t + ")");
    ensure(invEqual(m.inv, t.inSec), "Error (" + seed + "): m and t differ in inversion arrays after enqueue (" +
java.util.Arrays.toString(m.inv) + ", " + java.util.Arrays.toString(t.inSec) + ")");

    if (m.size() != 0) {
        ensure(m.first() == t.top(), "Error (" + seed + "): m and t differ in first element after enqueue (" + m.first() + ", " +
t.top() + ")");
    }

```

```
mThrowsNoElement = tThrowsNoElement = mThrowsOutOfBounds = tThrowsOutOfBounds = mThrowsIllegal =
tThrowsIllegal = null;
```

```
try {
    rm = m.dequeue();
}
catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }
catch (IllegalArgumentException e) { mThrowsIllegal = e; }
catch (java.util.NoSuchElementException e) { mThrowsNoElement = e; }
```

```
try {
    rt = t.top();
    t.remove();
}
catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }
catch (IllegalArgumentException e) { tThrowsIllegal = e; }
catch (java.util.NoSuchElementException e) { tThrowsNoElement = e; }
```

```
ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + seed + "): dequeue()
divergence in IndexOutOfBoundsException (" + mThrowsOutOfBounds + ", " + tThrowsOutOfBounds + "));
ensure((mThrowsIllegal == null) == (tThrowsIllegal == null), "Error (" + seed + "): dequeue() divergence in
IllegalArgumentException (" + mThrowsIllegal + ", " + tThrowsIllegal + "));
ensure((mThrowsNoElement == null) == (tThrowsNoElement == null), "Error (" + seed + "): dequeue() divergence
in java.util.NoSuchElementException (" + mThrowsNoElement + ", " + tThrowsNoElement + "));
if (mThrowsOutOfBounds == null) ensure(rt == rm, "Error (" + seed + "): divergence in dequeue() between t and m
(" + rt + ", " + rm + "));
```

```
ensure(heapEqual(m.heap, t.primary, m.size(), t.size()), "Error (" + seed + "): m and t differ after dequeue (" + m +
", " + t + "));
```

```
ensure(invEqual(m.inv, t.inSec), "Error (" + seed + "): m and t differ in inversion arrays after dequeue (" +
java.util.Arrays.toString(m.inv) + ", " + java.util.Arrays.toString(t.inSec) + "));
```

```
if (m.size() != 0) {
    ensure(m.first() == t.top(), "Error (" + seed + "): m and t differ in first element after dequeue (" + m.first() + ", " +
t.top() + "));
}
```

```
int pos = r.nextInt(n * 2);
```

```
try {
    m.remove(pos);
}
catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }
catch (IllegalArgumentException e) { mThrowsIllegal = e; }
catch (java.util.NoSuchElementException e) { mThrowsNoElement = e; }
```

```

try {
    t.remove(pos);
}
catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }
catch (IllegalArgumentException e) { tThrowsIllegal = e; }
catch (java.util.NoSuchElementException e) { tThrowsNoElement = e; }

ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + seed + "): remove(int)
divergence in IndexOutOfBoundsException (" + mThrowsOutOfBounds + ", " + tThrowsOutOfBounds + "));
ensure((mThrowsIllegal == null) == (tThrowsIllegal == null), "Error (" + seed + "): remove(int) divergence in
IllegalArgumentException (" + mThrowsIllegal + ", " + tThrowsIllegal + "));
ensure((mThrowsNoElement == null) == (tThrowsNoElement == null), "Error (" + seed + "): remove(int)
divergence in java.util.NoSuchElementException (" + mThrowsNoElement + ", " + tThrowsNoElement + "));
if (mThrowsOutOfBounds == null) ensure(rt == rm, "Error (" + seed + "): divergence in remove(int) between t and
m (" + rt + ", " + rm + "));

ensure(heapEqual(m.heap, t.primary, m.size(), t.size()), "Error (" + seed + "): m and t differ after remove(int) (" + m
+ ", " + t + "));
ensure(invEqual(m.inv, t.inSec), "Error (" + seed + "): m and t differ in inversion arrays after remove(int) (" +
java.util.Arrays.toString(m.inv) + ", " + java.util.Arrays.toString(t.inSec) + "));

if (m.size() != 0) {
    ensure(m.first() == t.top(), "Error (" + seed + "): m and t differ in first element after remove(int) (" + m.first() + ", "
+ t.top() + "));
}

pos = r.nextInt(n * 2);

try {
    m.changed(pos);
}
catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }
catch (IllegalArgumentException e) { mThrowsIllegal = e; }
catch (java.util.NoSuchElementException e) { mThrowsNoElement = e; }

try {
    t.change(pos);
}
catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }
catch (IllegalArgumentException e) { tThrowsIllegal = e; }
catch (java.util.NoSuchElementException e) { tThrowsNoElement = e; }

ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + seed + "): change(int)
divergence in IndexOutOfBoundsException (" + mThrowsOutOfBounds + ", " + tThrowsOutOfBounds + "));
ensure((mThrowsIllegal == null) == (tThrowsIllegal == null), "Error (" + seed + "): change(int) divergence in
IllegalArgumentException (" + mThrowsIllegal + ", " + tThrowsIllegal + "));
ensure((mThrowsNoElement == null) == (tThrowsNoElement == null), "Error (" + seed + "): change(int)

```

```

divergence in java.util.NoSuchElementException (" + mThrowsNoElement + ", " + tThrowsNoElement + "));
    if (mThrowsOutOfBounds == null) ensure(rt == rm, "Error (" + seed + "): divergence in change(int) between t and
m (" + rt + ", " + rm + "));

    ensure(heapEqual(m.heap, t.primary, m.size(), t.size()), "Error (" + seed + "): m and t differ after change(int) (" + m
+ ", " + t + "));
    ensure(invEqual(m.inv, t.inSec), "Error (" + seed + "): m and t differ in inversion arrays after change(int) (" +
java.util.Arrays.toString(m.inv) + ", " + java.util.Arrays.toString(t.inSec) + "));

    if (m.size() != 0) {
        ensure(m.first() == t.top(), "Error (" + seed + "): m and t differ in first element after change(int) (" + m.first() + ", " +
t.top() + "));
    }

    if (m.size() != 0) {

        refArray[m.first()] = genKey();

        m.changed();
        t.change();

        ensure(heapEqual(m.heap, t.primary, m.size(), t.size()), "Error (" + seed + "): m and t differ after change (" + m + ",
" + t + "));
        ensure(invEqual(m.inv, t.inSec), "Error (" + seed + "): m and t differ in inversion arrays after change (" +
java.util.Arrays.toString(m.inv) + ", " + java.util.Arrays.toString(t.inSec) + "));

        if (m.size() != 0) {
            ensure(m.first() == t.top(), "Error (" + seed + "): m and t differ in first element after change (" + m.first() + ", " +
t.top() + "));
        }
    }

    }

    /* Now we check that m actually holds the same data. */

    m.clear();
    ensure(m.isEmpty(), "Error (" + seed + "): m is not empty after clear()");

    System.out.println("Test OK");
}

public static void main(String args[]) {
    int n = Integer.parseInt(args[1]);
    if (args.length > 2) r = new java.util.Random(seed = Long.parseLong(args[2]));

```

```
try {
  if ("speedTest".equals(args[0]) || "speedComp".equals(args[0])) speedTest(n, "speedComp".equals(args[0]));
  else if ("test".equals(args[0])) runTest(n);
} catch(Throwable e) {
  e.printStackTrace(System.err);
  System.err.println("seed: " + seed);
}
}

#endif

}
```

Found in path(s):

```
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-
a775574/drv/HeapIndirectPriorityQueue.drv
```

No license file was found, but licenses were detected in source scan.

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work,

where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or

for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason

of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Found in path(s):

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/LICENSE-2.0

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2002-2017 Sebastiano Vigna

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

```

*/

package PACKAGE;

import it.unimi.dsi.fastutil.objects.ObjectIterator;
import it.unimi.dsi.fastutil.objects.ObjectIterable;
import it.unimi.dsi.fastutil.objects.ObjectSet;
import it.unimi.dsi.fastutil.objects.ObjectSets;

import VALUE_PACKAGE.VALUE_COLLECTION;
import VALUE_PACKAGE.VALUE_COLLECTIONS;
#if ! VALUE_CLASS_Object
import VALUE_PACKAGE.VALUE_SETS;
#endif

import java.util.Map;
import java.util.function.Consumer;
import PACKAGE.MAP.FastEntrySet;

/** A class providing static methods and objects that do useful things with type-specific maps.
 *
 * @see java.util.Collections
 */

public final class MAPS {

    private MAPS() {}

    /** Returns an iterator that will be {@linkplain FastEntrySet fast}, if possible, on the {@linkplain Map#entrySet()
    entry set} of the provided {@code map}.
    * @param map a map from which we will try to extract a (fast) iterator on the entry set.
    * @return an iterator on the entry set of the given map that will be fast, if possible.
    * @since 8.0.0
    */
    SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
    public static KEY_VALUE_GENERIC ObjectIterator<MAP.Entry KEY_VALUE_GENERIC> fastIterator(MAP
    KEY_VALUE_GENERIC map) {
        final ObjectSet<MAP.Entry KEY_VALUE_GENERIC> entries = map.ENTRYSET();
        return entries instanceof MAP.FastEntrySet ? ((MAP.FastEntrySet KEY_VALUE_GENERIC) entries).fastIterator()
: entries.iterator();
    }

    /** Iterates {@linkplain FastEntrySet#fastForEach(Consumer) quickly}, if possible, on the {@linkplain
    Map#entrySet() entry set} of the provided {@code map}.
    * @param map a map on which we will try to iterate {@linkplain FastEntrySet#fastForEach(Consumer) quickly}.
    * @param consumer the consumer that will be passed to {@link FastEntrySet#fastForEach(Consumer)}, if
    possible, or to {@link Iterable#forEach(Consumer)}.

```

```

* @since 8.1.0
*/
SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
public static KEY_VALUE_GENERIC void fastForEach(MAP KEY_VALUE_GENERIC map, final Consumer<?
super MAP.Entry KEY_VALUE_GENERIC> consumer) {
    final ObjectSet<MAP.Entry KEY_VALUE_GENERIC> entries = map.ENTRYSET();
    if (entries instanceof MAP.FastEntrySet) ((MAP.FastEntrySet KEY_VALUE_GENERIC)
entries).fastForEach(consumer);
    else entries.forEach(consumer);
}

/** Returns an iterable yielding an iterator that will be { @linkplain FastEntrySet fast}, if possible, on the
{ @linkplain Map#entrySet() entry set} of the provided { @code map}.
* @param map a map from which we will try to extract an iterable yielding a (fast) iterator on the entry set.
* @return an iterable yielding an iterator on the entry set of the given map that will be
* fast, if possible.
* @since 8.0.0
*/
SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
public static KEY_VALUE_GENERIC ObjectIterable<MAP.Entry KEY_VALUE_GENERIC> fastIterable(MAP
KEY_VALUE_GENERIC map) {
    final ObjectSet<MAP.Entry KEY_VALUE_GENERIC> entries = map.ENTRYSET();
    return entries instanceof MAP.FastEntrySet ? new ObjectIterable<MAP.Entry KEY_VALUE_GENERIC>() {
        public ObjectIterator<MAP.Entry KEY_VALUE_GENERIC> iterator() { return ((MAP.FastEntrySet
KEY_VALUE_GENERIC)entries).fastIterator(); }
    } : entries;
}

/** An immutable class representing an empty type-specific map.
*
* <p>This class may be useful to implement your own in case you subclass
* a type-specific map.
*/

public static class EmptyMap KEY_VALUE_GENERIC extends FUNCTIONS.EmptyFunction
KEY_VALUE_GENERIC implements MAP KEY_VALUE_GENERIC, java.io.Serializable, Cloneable {

    private static final long serialVersionUID = -7046029254386353129L;

    protected EmptyMap() {}

    @Override
    public boolean containsValue(final VALUE_TYPE v) { return false; }

    #if VALUES_PRIMITIVE
    /** { @inheritDoc}

```

```

* @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public boolean containsValue(final Object ov) { return false; }
#endif

@Override
public void putAll(final Map<? extends KEY_GENERIC_CLASS, ? extends VALUE_GENERIC_CLASS> m) {
throw new UnsupportedOperationException(); }

@SuppressWarnings("unchecked")
@Override
public ObjectSet<MAP.Entry KEY_VALUE_GENERIC> ENTRYSET() { return ObjectSets.EMPTY_SET; }

SUPPRESS_WARNINGS_KEY_UNCHECKED
@Override
public SET KEY_GENERIC keySet() { return SETS.EMPTY_SET; }

SUPPRESS_WARNINGS_VALUE_UNCHECKED
@Override
public VALUE_COLLECTION VALUE_GENERIC values() { return VALUE_SETS.EMPTY_SET; }

@Override
public Object clone() { return EMPTY_MAP; }

@Override
public boolean isEmpty() { return true; }

@Override
public int hashCode() { return 0; }

@Override
public boolean equals(final Object o) {
if (!(o instanceof Map)) return false;
return ((Map<?,?>)o).isEmpty();
}

@Override
public String toString() { return "{}"; }
}

/** An empty type-specific map (immutable). It is serializable and cloneable.
*/
SUPPRESS_WARNINGS_KEY_VALUE_RAWTYPES
public static final EmptyMap EMPTY_MAP = new EmptyMap();

#if KEYS_REFERENCE || VALUES_REFERENCE

```

```

/** Returns an empty map (immutable). It is serializable and cloneable.
 *
 * <p>This method provides a typesafe access to {@link #EMPTY_MAP}.
 * @return an empty map (immutable).
 */
@SuppressWarnings("unchecked")
public static KEY_VALUE_GENERIC MAP KEY_VALUE_GENERIC emptyMap() {
    return EMPTY_MAP;
}
#endif

/** An immutable class representing a type-specific singleton map.
 *
 * <p>This class may be useful to implement your own in case you subclass
 * a type-specific map.
 */

public static class Singleton KEY_VALUE_GENERIC extends FUNCTIONS.Singleton KEY_VALUE_GENERIC
implements MAP KEY_VALUE_GENERIC, java.io.Serializable, Cloneable {

    private static final long serialVersionUID = -7046029254386353129L;

    protected transient ObjectSet<MAP.Entry KEY_VALUE_GENERIC> entries;
    protected transient SET KEY_GENERIC keys;
    protected transient VALUE_COLLECTION VALUE_GENERIC values;

    protected Singleton(final KEY_GENERIC_TYPE key, final VALUE_GENERIC_TYPE value) {
        super(key, value);
    }

    @Override
    public boolean containsValue(final VALUE_TYPE v) { return VALUE_EQUALS(value, v); }
#if VALUES_PRIMITIVE
    /** {@inheritDoc}
     * @deprecated Please use the corresponding type-specific method instead. */
    @Deprecated
    @Override
    public boolean containsValue(final Object ov) { return VALUE_EQUALS(VALUE_OBJ2TYPE(ov), value); }
#endif

    @Override
    public void putAll(final Map<? extends KEY_GENERIC_CLASS, ? extends VALUE_GENERIC_CLASS> m) {
        throw new UnsupportedOperationException(); }

    @Override
    public ObjectSet<MAP.Entry KEY_VALUE_GENERIC> ENTRYSET() { if (entries == null) entries =
ObjectSets.singleton(new ABSTRACT_MAP.BasicEntry KEY_VALUE_GENERIC_DIAMOND(key, value));
return entries; }

```

```

#if KEYS_PRIMITIVE || VALUES_PRIMITIVE
/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
#else
/** {@inheritDoc} */
#endif
@Override
@SuppressWarnings({ "rawtypes", "unchecked" })
public ObjectSet<Map.Entry<KEY_GENERIC_CLASS, VALUE_GENERIC_CLASS>> entrySet() { return
(ObjectSet)ENTRYSET(); }

@Override
public SET KEY_GENERIC keySet() { if (keys == null) keys = SETS.singleton(key); return keys; }

@Override
public VALUE_COLLECTION VALUE_GENERIC values() { if (values == null) values =
VALUE_SETS.singleton(value); return values; }

@Override
public boolean isEmpty() { return false; }

@Override
public int hashCode() { return KEY2JAVAHASH(key) ^ VALUE2JAVAHASH(value); }

@Override
public boolean equals(final Object o) {
if (o == this) return true;
if (!(o instanceof Map)) return false;

Map<?,?> m = (Map<?,?>)o;
if (m.size() != 1) return false;
return m.entrySet().iterator().next().equals(entrySet().iterator().next());
}

@Override
public String toString() { return "{" + key + "=>" + value + "}"; }
}

/** Returns a type-specific immutable map containing only the specified pair. The returned map is serializable and
cloneable.
 *
 * <p>Note that albeit the returned map is immutable, its default return value may be changed.
 *
 * @param key the only key of the returned map.
 * @param value the only value of the returned map.
 * @return a type-specific immutable map containing just the pair {@code <key,value>}.

```

```

*/

public static KEY_VALUE_GENERIC MAP KEY_VALUE_GENERIC singleton(final KEY_GENERIC_TYPE
key, VALUE_GENERIC_TYPE value) { return new Singleton KEY_VALUE_GENERIC_DIAMOND(key, value);
}

#if KEYS_PRIMITIVE || VALUES_PRIMITIVE

/** Returns a type-specific immutable map containing only the specified pair. The returned map is serializable and
cloneable.
*
* <p>Note that albeit the returned map is immutable, its default return value may be changed.
*
* @param key the only key of the returned map.
* @param value the only value of the returned map.
* @return a type-specific immutable map containing just the pair { @code &lt;key,value> }.
*/

public static KEY_VALUE_GENERIC MAP KEY_VALUE_GENERIC singleton(final KEY_GENERIC_CLASS
key, final VALUE_GENERIC_CLASS value) { return new Singleton
KEY_VALUE_GENERIC_DIAMOND(KEY_CLASS2TYPE(key), VALUE_CLASS2TYPE(value)); }

#endif

/** A synchronized wrapper class for maps. */

public static class SynchronizedMap KEY_VALUE_GENERIC extends FUNCTIONS.SynchronizedFunction
KEY_VALUE_GENERIC implements MAP KEY_VALUE_GENERIC, java.io.Serializable {

private static final long serialVersionUID = -7046029254386353129L;

protected final MAP KEY_VALUE_GENERIC map;

protected transient ObjectSet<MAP.Entry KEY_VALUE_GENERIC> entries;
protected transient SET KEY_GENERIC keys;
protected transient VALUE_COLLECTION VALUE_GENERIC values;

protected SynchronizedMap(final MAP KEY_VALUE_GENERIC m, final Object sync) {
super(m, sync);
this.map = m;
}

protected SynchronizedMap(final MAP KEY_VALUE_GENERIC m) {
super(m);
this.map = m;
}
}

```

```

@Override
public boolean containsValue(final VALUE_TYPE v) { synchronized(sync) { return map.containsValue(v); } }

#if VALUES_PRIMITIVE
/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public boolean containsValue(final Object ov) { synchronized(sync) { return map.containsValue(ov); } }
#endif

@Override
public void putAll(final Map<? extends KEY_GENERIC_CLASS, ? extends VALUE_GENERIC_CLASS> m) {
synchronized(sync) { map.putAll(m); } }

@Override
public ObjectSet<MAP.Entry KEY_VALUE_GENERIC> ENTRYSET() { synchronized(sync) { if (entries == null)
entries = ObjectSets.synchronize(map.ENTRYSET(), sync); return entries; } }

#if KEYS_PRIMITIVE || VALUES_PRIMITIVE
/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
#else
/** {@inheritDoc} */
#endif
@Override
@SuppressWarnings({ "unchecked", "rawtypes" })
public ObjectSet<Map.Entry<KEY_GENERIC_CLASS, VALUE_GENERIC_CLASS>> entrySet() { return
(ObjectSet)ENTRYSET(); }

@Override
public SET KEY_GENERIC keySet() {
synchronized(sync) { if (keys == null) keys = SETS.synchronize(map.keySet(), sync); return keys; }
}

@Override
public VALUE_COLLECTION VALUE_GENERIC values() {
synchronized(sync) { if (values == null) return VALUE_COLLECTIONS.synchronize(map.values(), sync); return
values; }
}

@Override
public boolean isEmpty() { synchronized(sync) { return map.isEmpty(); } }

@Override
public int hashCode() { synchronized(sync) { return map.hashCode(); } }

```

```

@Override
public boolean equals(final Object o) {
    if (o == this) return true;
    synchronized(sync) { return map.equals(o); }
}

private void writeObject(java.io.ObjectOutputStream s) throws java.io.IOException {
    synchronized(sync) { s.defaultWriteObject(); }
}

// Defaultable methods

@Override
public VALUE_GENERIC_TYPE getOrDefault(final KEY_TYPE key, final VALUE_GENERIC_TYPE
defaultValue) { synchronized(sync) { return map.getOrDefault(key, defaultValue); } }

@Override
public void forEach(final java.util.function.BiConsumer<? super KEY_GENERIC_CLASS, ? super
VALUE_GENERIC_CLASS> action) { synchronized (sync) { map.forEach(action); } }

@Override
public void replaceAll(final java.util.function.BiFunction<? super KEY_GENERIC_CLASS, ? super
VALUE_GENERIC_CLASS, ? extends VALUE_GENERIC_CLASS> function) { synchronized (sync) {
map.replaceAll(function); } }

@Override
public VALUE_GENERIC_TYPE putIfAbsent(final KEY_GENERIC_TYPE key, final VALUE_GENERIC_TYPE
value) { synchronized(sync) { return map.putIfAbsent(key, value); } }

@Override
public boolean remove(final KEY_TYPE key, final VALUE_TYPE value) { synchronized(sync) { return
map.remove(key, value); } }

@Override
public VALUE_GENERIC_TYPE replace(final KEY_GENERIC_TYPE key, final VALUE_GENERIC_TYPE
value) { synchronized(sync) { return map.replace(key, value); } }

@Override
public boolean replace(final KEY_GENERIC_TYPE key, final VALUE_GENERIC_TYPE oldValue, final
VALUE_GENERIC_TYPE newValue) { synchronized(sync) { return map.replace(key, oldValue, newValue); } }

#ifdef JDK_PRIMITIVE_FUNCTION
@Override
public VALUE_GENERIC_TYPE COMPUTE_IF_ABSENT_JDK(final KEY_GENERIC_TYPE key, final
JDK_PRIMITIVE_FUNCTION KEY_SUPER_GENERIC VALUE_EXTENDS_GENERIC mappingFunction) {
synchronized(sync) { return map.COMPUTE_IF_ABSENT_JDK(key, mappingFunction); } }
#endif

```

```

#if KEYS_PRIMITIVE && VALUES_PRIMITIVE
    @Override
    public VALUE_GENERIC_TYPE COMPUTE_IF_ABSENT_NULLABLE(final KEY_GENERIC_TYPE key,
final JDK_KEY_TO_GENERIC_FUNCTION<? extends VALUE_GENERIC_CLASS> mappingFunction) {
synchronized(sync) { return map.COMPUTE_IF_ABSENT_NULLABLE(key, mappingFunction); } }
#endif

#if KEYS_PRIMITIVE || VALUES_PRIMITIVE
    @Override
    public VALUE_GENERIC_TYPE COMPUTE_IF_ABSENT_PARTIAL(final KEY_GENERIC_TYPE key, final
FUNCTION KEY_SUPER_GENERIC VALUE_EXTENDS_GENERIC mappingFunction) { synchronized(sync) {
return map.COMPUTE_IF_ABSENT_PARTIAL(key, mappingFunction); } }
#endif

    @Override
    public VALUE_GENERIC_TYPE COMPUTE_IF_PRESENT(final KEY_GENERIC_TYPE key, final
java.util.function.BiFunction<? super KEY_GENERIC_CLASS, ? super VALUE_GENERIC_CLASS, ? extends
VALUE_GENERIC_CLASS> remappingFunction) {
    synchronized (sync) { return map.COMPUTE_IF_PRESENT(key, remappingFunction); }
    }

    @Override
    public VALUE_GENERIC_TYPE COMPUTE(final KEY_GENERIC_TYPE key, final
java.util.function.BiFunction<? super KEY_GENERIC_CLASS, ? super VALUE_GENERIC_CLASS, ? extends
VALUE_GENERIC_CLASS> remappingFunction) {
    synchronized (sync) { return map.COMPUTE(key, remappingFunction); }
    }

    @Override
    public VALUE_GENERIC_TYPE MERGE(final KEY_GENERIC_TYPE key, final VALUE_GENERIC_TYPE
value, final java.util.function.BiFunction<? super VALUE_GENERIC_CLASS, ? super
VALUE_GENERIC_CLASS, ? extends VALUE_GENERIC_CLASS> remappingFunction) {
    synchronized (sync) { return map.MERGE(key, value, remappingFunction); }
    }

#if KEYS_PRIMITIVE || VALUES_PRIMITIVE
    /** {@inheritDoc}
    * @deprecated Please use the corresponding type-specific method instead. */
    @Deprecated
    @Override
    public VALUE_GENERIC_CLASS getOrDefault(final Object key, final VALUE_GENERIC_CLASS
defaultValue) { synchronized (sync) { return map.getOrDefault(key, defaultValue); } }

    /** {@inheritDoc}
    * @deprecated Please use the corresponding type-specific method instead. */
    @Deprecated
    @Override
    public boolean remove(final Object key, final Object value) { synchronized (sync) { return map.remove(key, value); } }

```

```

} }

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public VALUE_GENERIC_CLASS replace(final KEY_GENERIC_CLASS key, final VALUE_GENERIC_CLASS
value) { synchronized (sync) { return map.replace(key, value); } }

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public boolean replace(final KEY_GENERIC_CLASS key, final VALUE_GENERIC_CLASS oldValue, final
VALUE_GENERIC_CLASS newValue) { synchronized (sync) { return map.replace(key, oldValue, newValue); } }

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public VALUE_GENERIC_CLASS putIfAbsent(final KEY_GENERIC_CLASS key, final
VALUE_GENERIC_CLASS value) { synchronized (sync) { return map.putIfAbsent(key, value); } }

#if KEYS_PRIMITIVE
/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
#endif
@Override
public VALUE_GENERIC_CLASS computeIfAbsent(final KEY_GENERIC_CLASS key, final
java.util.function.Function<? super KEY_GENERIC_CLASS, ? extends VALUE_GENERIC_CLASS>
mappingFunction) { synchronized (sync) { return map.computeIfAbsent(key, mappingFunction); } }

#if KEYS_PRIMITIVE
/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
#endif
@Override
public VALUE_GENERIC_CLASS computeIfPresent(final KEY_GENERIC_CLASS key, final
java.util.function.BiFunction<? super KEY_GENERIC_CLASS, ? super VALUE_GENERIC_CLASS, ? extends
VALUE_GENERIC_CLASS> remappingFunction) { synchronized (sync) { return map.computeIfPresent(key,
remappingFunction); } }

#if KEYS_PRIMITIVE
/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated

```

```

#endif
    @Override
    public VALUE_GENERIC_CLASS compute(final KEY_GENERIC_CLASS key, final
java.util.function.BiFunction<? super KEY_GENERIC_CLASS, ? super VALUE_GENERIC_CLASS, ? extends
VALUE_GENERIC_CLASS> remappingFunction) { synchronized (sync) { return map.compute(key,
remappingFunction); } }

    /** {@inheritDoc}
    * @deprecated Please use the corresponding type-specific method instead. */
    @Deprecated
    @Override
    public VALUE_GENERIC_CLASS merge(final KEY_GENERIC_CLASS key, final VALUE_GENERIC_CLASS
value, final java.util.function.BiFunction<? super VALUE_GENERIC_CLASS, ? super
VALUE_GENERIC_CLASS, ? extends VALUE_GENERIC_CLASS> remappingFunction) { synchronized (sync)
{ return map.merge(key, value, remappingFunction); } }
#endif

}

/** Returns a synchronized type-specific map backed by the given type-specific map.
*
* @param m the map to be wrapped in a synchronized map.
* @return a synchronized view of the specified map.
* @see java.util.Collections#synchronizedMap(Map)
*/
public static KEY_VALUE_GENERIC MAP KEY_VALUE_GENERIC synchronize(final MAP
KEY_VALUE_GENERIC m) { return new SynchronizedMap KEY_VALUE_GENERIC_DIAMOND(m); }

/** Returns a synchronized type-specific map backed by the given type-specific map, using an assigned object to
synchronize.
*
* @param m the map to be wrapped in a synchronized map.
* @param sync an object that will be used to synchronize the access to the map.
* @return a synchronized view of the specified map.
* @see java.util.Collections#synchronizedMap(Map)
*/

public static KEY_VALUE_GENERIC MAP KEY_VALUE_GENERIC synchronize(final MAP
KEY_VALUE_GENERIC m, final Object sync) { return new SynchronizedMap
KEY_VALUE_GENERIC_DIAMOND(m, sync); }

/** An unmodifiable wrapper class for maps. */

public static class UnmodifiableMap KEY_VALUE_GENERIC extends FUNCTIONS.UnmodifiableFunction
KEY_VALUE_GENERIC implements MAP KEY_VALUE_GENERIC, java.io.Serializable {

```

```

private static final long serialVersionUID = -7046029254386353129L;

protected final MAP KEY_VALUE_GENERIC map;

protected transient ObjectSet<MAP.Entry KEY_VALUE_GENERIC> entries;
protected transient SET KEY_GENERIC keys;
protected transient VALUE_COLLECTION VALUE_GENERIC values;

protected UnmodifiableMap(final MAP KEY_VALUE_GENERIC m) {
    super(m);
    this.map = m;
}

@Override
public boolean containsValue(final VALUE_TYPE v) { return map.containsValue(v); }
#if VALUES_PRIMITIVE
/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public boolean containsValue(final Object ov) { return map.containsValue(ov); }
#endif

@Override
public void putAll(final Map<? extends KEY_GENERIC_CLASS, ? extends VALUE_GENERIC_CLASS> m) {
    throw new UnsupportedOperationException(); }

@Override
public ObjectSet<MAP.Entry KEY_VALUE_GENERIC> ENTRYSET() { if (entries == null) entries =
ObjectSets.unmodifiable(map.ENTRYSET()); return entries; }

#if KEYS_PRIMITIVE || VALUES_PRIMITIVE
/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
#else
/** {@inheritDoc} */
#endif
@Override
@SuppressWarnings({ "unchecked", "rawtypes" })
public ObjectSet<Map.Entry<KEY_GENERIC_CLASS, VALUE_GENERIC_CLASS>> entrySet() { return
(ObjectSet)ENTRYSET(); }

@Override
public SET KEY_GENERIC keySet() { if (keys == null) keys = SETS.unmodifiable(map.keySet()); return keys; }

@Override
public VALUE_COLLECTION VALUE_GENERIC values() { if (values == null) return

```

```

VALUE_COLLECTIONS.unmodifiable(map.values()); return values; }

@Override
public boolean isEmpty() { return map.isEmpty(); }

@Override
public int hashCode() { return map.hashCode(); }

@Override
public boolean equals(final Object o) {
    if (o == this) return true;
    return map.equals(o);
}

// Defaultable methods

@Override
public VALUE_GENERIC_TYPE getOrDefault(final KEY_TYPE key, final VALUE_GENERIC_TYPE
defaultValue) { return map.getOrDefault(key, defaultValue); }

@Override
public void forEach(final java.util.function.BiConsumer<? super KEY_GENERIC_CLASS, ? super
VALUE_GENERIC_CLASS> action) { map.forEach(action); }

@Override
public void replaceAll(final java.util.function.BiFunction<? super KEY_GENERIC_CLASS, ? super
VALUE_GENERIC_CLASS, ? extends VALUE_GENERIC_CLASS> function) { throw new
UnsupportedOperationException(); }

@Override
public VALUE_GENERIC_TYPE putIfAbsent(final KEY_GENERIC_TYPE key, final VALUE_GENERIC_TYPE
value) { throw new UnsupportedOperationException(); }

@Override
public boolean remove(final KEY_TYPE key, final VALUE_TYPE value) { throw new
UnsupportedOperationException(); }

@Override
public VALUE_GENERIC_TYPE replace(final KEY_GENERIC_TYPE key, final VALUE_GENERIC_TYPE
value) { throw new UnsupportedOperationException(); }

@Override
public boolean replace(final KEY_GENERIC_TYPE key, final VALUE_GENERIC_TYPE oldValue, final
VALUE_GENERIC_TYPE newValue) { throw new UnsupportedOperationException(); }

#ifdef JDK_PRIMITIVE_FUNCTION
@Override
public VALUE_GENERIC_TYPE COMPUTE_IF_ABSENT_JDK(final KEY_GENERIC_TYPE key, final

```

```

JDK_PRIMITIVE_FUNCTION KEY_SUPER_GENERIC VALUE_EXTENDS_GENERIC mappingFunction) {
throw new UnsupportedOperationException(); }
#endif

#if KEYS_PRIMITIVE && VALUES_PRIMITIVE
@Override
public VALUE_GENERIC_TYPE COMPUTE_IF_ABSENT_NULLABLE(final KEY_GENERIC_TYPE key,
final JDK_KEY_TO_GENERIC_FUNCTION<? extends VALUE_GENERIC_CLASS> mappingFunction) { throw
new UnsupportedOperationException(); }
#endif

#if KEYS_PRIMITIVE || VALUES_PRIMITIVE
@Override
public VALUE_GENERIC_TYPE COMPUTE_IF_ABSENT_PARTIAL(final KEY_GENERIC_TYPE key, final
FUNCTION KEY_SUPER_GENERIC VALUE_EXTENDS_GENERIC mappingFunction) { throw new
UnsupportedOperationException(); }
#endif

@Override
public VALUE_GENERIC_TYPE COMPUTE_IF_PRESENT(final KEY_GENERIC_TYPE key, final
java.util.function.BiFunction<? super KEY_GENERIC_CLASS, ? super VALUE_GENERIC_CLASS, ? extends
VALUE_GENERIC_CLASS> remappingFunction) { throw new UnsupportedOperationException(); }

@Override
public VALUE_GENERIC_TYPE COMPUTE(final KEY_GENERIC_TYPE key, final
java.util.function.BiFunction<? super KEY_GENERIC_CLASS, ? super VALUE_GENERIC_CLASS, ? extends
VALUE_GENERIC_CLASS> remappingFunction) { throw new UnsupportedOperationException(); }

@Override
public VALUE_GENERIC_TYPE MERGE(final KEY_GENERIC_TYPE key, final VALUE_GENERIC_TYPE
value, final java.util.function.BiFunction<? super VALUE_GENERIC_CLASS, ? super
VALUE_GENERIC_CLASS, ? extends VALUE_GENERIC_CLASS> remappingFunction) { throw new
UnsupportedOperationException(); }

#if KEYS_PRIMITIVE || VALUES_PRIMITIVE
/** {@inheritDoc}
* @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public VALUE_GENERIC_CLASS getOrDefault(final Object key, final VALUE_GENERIC_CLASS
defaultValue) { return map.getOrDefault(key, defaultValue); }

/** {@inheritDoc}
* @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public boolean remove(final Object key, final Object value) { throw new UnsupportedOperationException(); }

```

```

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public VALUE_GENERIC_CLASS replace(final KEY_GENERIC_CLASS key, final VALUE_GENERIC_CLASS
value) { throw new UnsupportedOperationException(); }

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public boolean replace(final KEY_GENERIC_CLASS key, final VALUE_GENERIC_CLASS oldValue, final
VALUE_GENERIC_CLASS newValue) { throw new UnsupportedOperationException(); }

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public VALUE_GENERIC_CLASS putIfAbsent(final KEY_GENERIC_CLASS key, final
VALUE_GENERIC_CLASS value) { throw new UnsupportedOperationException(); }

#if KEYS_PRIMITIVE
/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
#endif
@Override
public VALUE_GENERIC_CLASS computeIfAbsent(final KEY_GENERIC_CLASS key, final
java.util.function.Function<? super KEY_GENERIC_CLASS, ? extends VALUE_GENERIC_CLASS>
mappingFunction) { throw new UnsupportedOperationException(); }

#if KEYS_PRIMITIVE
/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
#endif
@Override
public VALUE_GENERIC_CLASS computeIfPresent(final KEY_GENERIC_CLASS key, final
java.util.function.BiFunction<? super KEY_GENERIC_CLASS, ? super VALUE_GENERIC_CLASS, ? extends
VALUE_GENERIC_CLASS> remappingFunction) { throw new UnsupportedOperationException(); }

#if KEYS_PRIMITIVE
/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
#endif
@Override
public VALUE_GENERIC_CLASS compute(final KEY_GENERIC_CLASS key, final

```

```

java.util.function.BiFunction<? super KEY_GENERIC_CLASS, ? super VALUE_GENERIC_CLASS, ? extends
VALUE_GENERIC_CLASS> remappingFunction) { throw new UnsupportedOperationException(); }

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public VALUE_GENERIC_CLASS merge(final KEY_GENERIC_CLASS key, final VALUE_GENERIC_CLASS
value, final java.util.function.BiFunction<? super VALUE_GENERIC_CLASS, ? super
VALUE_GENERIC_CLASS, ? extends VALUE_GENERIC_CLASS> remappingFunction) { throw new
UnsupportedOperationException(); }
#endif

}

/** Returns an unmodifiable type-specific map backed by the given type-specific map.
 *
 * @param m the map to be wrapped in an unmodifiable map.
 * @return an unmodifiable view of the specified map.
 * @see java.util.Collections#unmodifiableMap(Map)
 */
public static KEY_VALUE_GENERIC MAP KEY_VALUE_GENERIC unmodifiable(final MAP
KEY_VALUE_GENERIC m) { return new UnmodifiableMap KEY_VALUE_GENERIC_DIAMOND(m); }

}

```

Found in path(s):

```

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-
a775574/drv/Maps.drv

```

No license file was found, but licenses were detected in source scan.

```

/*
 * Copyright (C) 2002-2017 Sebastiano Vigna
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

```

```

package PACKAGE;

import it.unimi.dsi.fastutil.Stack;

/** A type-specific {@link Stack}; provides some additional methods that use polymorphism to avoid (un)boxing.
 */

public interface STACK KEY_GENERIC extends Stack<KEY_GENERIC_CLASS> {

    /** Pushes the given object on the stack.
     * @param k the object to push on the stack.
     * @see Stack#push(Object)
     */
    void push(KEY_TYPE k);

    /** Pops the top off the stack.
     *
     * @return the top of the stack.
     * @see Stack#pop()
     */
    KEY_TYPE POP();

    /** Peeks at the top of the stack (optional operation).
     * @return the top of the stack.
     * @see Stack#top()
     */
    KEY_TYPE TOP();

    /** Peeks at an element on the stack (optional operation).
     * @param i an index from the stop of the stack (0 represents the top).
     * @return the <code>i</code>-th element on the stack.
     * @see Stack#peek(int)
     */
    KEY_TYPE PEEK(int i);

    /** {@inheritDoc}
     * <p>This default implementation delegates to the corresponding type-specific method.
     * @deprecated Please use the corresponding type-specific method instead. */
    @Deprecated
    @Override
    default void push(KEY_GENERIC_CLASS o) {
        push(o.KEY_VALUE());
    }

    /** {@inheritDoc}
     * <p>This default implementation delegates to the corresponding type-specific method.
     * @deprecated Please use the corresponding type-specific method instead. */
    @Deprecated

```

```

@Override
default KEY_GENERIC_CLASS pop() {
    return KEY_CLASS.valueOf(POP());
}

/** { @inheritDoc}
 * <p>This default implementation delegates to the corresponding type-specific method.
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default KEY_GENERIC_CLASS top() {
    return KEY_CLASS.valueOf(TOP());
}

/** { @inheritDoc}
 * <p>This default implementation delegates to the corresponding type-specific method.
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default KEY_GENERIC_CLASS peek(final int i) {
    return KEY_CLASS.valueOf(PEEK(i));
}
}

```

Found in path(s):

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/drv/Stack.drv

No license file was found, but licenses were detected in source scan.

```

/*
 * Copyright (C) 2002-2017 Sebastiano Vigna
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

```

package PACKAGE;

```

import it.unimi.dsi.fastutil.BigList;
import java.util.Collection;
import java.util.List;
import java.util.Random;

/** A class providing static methods and objects that do useful things with type-specific big lists.
 *
 * @see java.util.Collections
 * @see it.unimi.dsi.fastutil.BigList
 */

public final class BIG_LISTS {

    private BIG_LISTS() {}

    /** Shuffles the specified big list using the specified pseudorandom number generator.
     *
     * @param l the big list to be shuffled.
     * @param random a pseudorandom number generator.
     * @return { @code l }.
     */
    public static KEY_GENERIC BIG_LIST KEY_GENERIC shuffle(final BIG_LIST KEY_GENERIC l, final
Random random) {
        for(long i = l.size64(); i-- != 0;) {
            final long p = (random.nextLong() & 0x7FFFFFFFFFFFFFFFL) % (i + 1);
            final KEY_GENERIC_TYPE t = l.GET_KEY(i);
            l.set(i, l.GET_KEY(p));
            l.set(p, t);
        }
        return l;
    }

    /** An immutable class representing an empty type-specific big list.
     *
     * <p>This class may be useful to implement your own in case you subclass
     * a type-specific list.
     */

    public static class EmptyBigList KEY_GENERIC extends COLLECTIONS.EmptyCollection KEY_GENERIC
implements BIG_LIST KEY_GENERIC, java.io.Serializable, Cloneable {

        private static final long serialVersionUID = -7046029254386353129L;

        protected EmptyBigList() {}

        @Override
        public KEY_GENERIC_TYPE GET_KEY(long i) { throw new IndexOutOfBoundsException(); }

```

```

@Override
public boolean REMOVE(KEY_TYPE k) { throw new UnsupportedOperationException(); }

@Override
public KEY_GENERIC_TYPE REMOVE_KEY(long i) { throw new UnsupportedOperationException(); }

@Override
public void add(final long index, final KEY_GENERIC_TYPE k) { throw new UnsupportedOperationException();
}

@Override
public KEY_GENERIC_TYPE set(final long index, final KEY_GENERIC_TYPE k) { throw new
UnsupportedOperationException(); }

@Override
public long indexOf(KEY_TYPE k) { return -1; }

@Override
public long lastIndexOf(KEY_TYPE k) { return -1; }

@Override
public boolean addAll(long i, Collection<? extends KEY_GENERIC_CLASS> c) { throw new
UnsupportedOperationException(); }

#if KEYS_PRIMITIVE
@Override
public boolean addAll(COLLECTION c) { throw new UnsupportedOperationException(); }

@Override
public boolean addAll(BIG_LIST c) { throw new UnsupportedOperationException(); }

@Override
public boolean addAll(long i, COLLECTION c) { throw new UnsupportedOperationException(); }

@Override
public boolean addAll(long i, BIG_LIST c) { throw new UnsupportedOperationException(); }

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public void add(final long index, final KEY_GENERIC_CLASS k) { throw new UnsupportedOperationException();
}

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated

```

```

@Override
public boolean add(final KEY_GENERIC_CLASS k) { throw new UnsupportedOperationException(); }

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public KEY_GENERIC_CLASS get(long i) { throw new IndexOutOfBoundsException(); }

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public KEY_GENERIC_CLASS set(final long index, final KEY_GENERIC_CLASS k) { throw new
UnsupportedOperationException(); }

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public KEY_GENERIC_CLASS remove(long k) { throw new UnsupportedOperationException(); }

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public long indexOf(Object k) { return -1; }

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public long lastIndexOf(Object k) { return -1; }
#endif

@Override
SUPPRESS_WARNINGS_KEY_UNCHECKED
public KEY_BIG_LIST_ITERATOR KEY_GENERIC listIterator() { return
BIG_LIST_ITERATORS.EMPTY_BIG_LIST_ITERATOR; }

@Override
SUPPRESS_WARNINGS_KEY_UNCHECKED
public KEY_BIG_LIST_ITERATOR KEY_GENERIC iterator() { return
BIG_LIST_ITERATORS.EMPTY_BIG_LIST_ITERATOR; }

@Override
SUPPRESS_WARNINGS_KEY_UNCHECKED
public KEY_BIG_LIST_ITERATOR KEY_GENERIC listIterator(long i) { if (i == 0) return

```

```

BIG_LIST_ITERATORS.EMPTY_BIG_LIST_ITERATOR; throw new
IndexOutOfBoundsException(String.valueOf(i)); }

@Override
public BIG_LIST KEY_GENERIC subList(long from, long to) { if (from == 0 && to == 0) return this; throw new
IndexOutOfBoundsException(); }

@Override
public void getElements(long from, KEY_TYPE[][] a, long offset, long length) {
BIG_ARRAYS.ensureOffsetLength(a, offset, length); if (from != 0) throw new IndexOutOfBoundsException(); }

@Override
public void removeElements(long from, long to) { throw new UnsupportedOperationException(); }

@Override
public void addElements(long index, final KEY_GENERIC_TYPE a[], long offset, long length) { throw new
UnsupportedOperationException(); }

@Override
public void addElements(long index, final KEY_GENERIC_TYPE a[]) { throw new
UnsupportedOperationException(); }

@Override
public void size(long s) { throw new UnsupportedOperationException(); }

@Override
public long size64() { return 0; }

#if ! KEY_CLASS_Reference
@Override
public int compareTo(final BigList<? extends KEY_GENERIC_CLASS> o) {
if (o == this) return 0;
return ((BigList<?>)o).isEmpty() ? 0 : -1;
}
#endif
@Override
public Object clone() { return EMPTY_BIG_LIST; }

@Override
public int hashCode() { return 1; }

@Override
@SuppressWarnings("rawtypes")
public boolean equals(Object o) { return o instanceof BigList && ((BigList)o).isEmpty(); }

@Override
public String toString() { return "[]"; }

```

```

private Object readResolve() { return EMPTY_BIG_LIST; }
}

/** An empty big list (immutable). It is serializable and cloneable.
 */
SUPPRESS_WARNINGS_KEY_RAWTYPES
public static final EmptyBigList EMPTY_BIG_LIST = new EmptyBigList();

#if KEYS_REFERENCE
/** Returns an empty big list (immutable). It is serializable and cloneable.
 *
 * <p>This method provides a typesafe access to { @link #EMPTY_BIG_LIST}.
 * @return an empty big list (immutable).
 */
@SuppressWarnings("unchecked")
public static KEY_GENERIC BIG_LIST KEY_GENERIC emptyList() {
    return EMPTY_BIG_LIST;
}
#endif

/** An immutable class representing a type-specific singleton big list.
 *
 * <p>This class may be useful to implement your own in case you subclass
 * a type-specific big list.
 */

public static class Singleton KEY_GENERIC extends ABSTRACT_BIG_LIST KEY_GENERIC implements
java.io.Serializable, Cloneable {

    private static final long serialVersionUID = -7046029254386353129L;

    private final KEY_GENERIC_TYPE element;

    protected Singleton(final KEY_GENERIC_TYPE element) { this.element = element; }

    @Override
    public KEY_GENERIC_TYPE GET_KEY(final long i) { if (i == 0) return element; throw new
IndexOutOfBoundsException(); }

    @Override
    public boolean REMOVE(KEY_TYPE k) { throw new UnsupportedOperationException(); }

    @Override
    public KEY_GENERIC_TYPE REMOVE_KEY(final long i) { throw new UnsupportedOperationException(); }

    @Override
    public boolean contains(final KEY_TYPE k) { return KEY_EQUALS(k, element); }

```

```
/* Slightly optimized w.r.t. the one in ABSTRACT_SET. */
```

```
@Override  
public KEY_TYPE[] TO_KEY_ARRAY() {  
    KEY_TYPE a[] = new KEY_TYPE[1];  
    a[0] = element;  
    return a;  
}
```

```
@Override  
public KEY_BIG_LIST_ITERATOR KEY_GENERIC listIterator() { return  
BIG_LIST_ITERATORS.singleton(element); }
```

```
@Override  
public KEY_BIG_LIST_ITERATOR KEY_GENERIC listIterator(long i) {  
    if (i > 1 || i < 0) throw new IndexOutOfBoundsException();  
    KEY_BIG_LIST_ITERATOR KEY_GENERIC l = listIterator();  
    if (i == 1) l.NEXT_KEY();  
    return l;  
}
```

```
@Override  
SUPPRESS_WARNINGS_KEY_UNCHECKED  
public BIG_LIST KEY_GENERIC subList(final long from, final long to) {  
    ensureIndex(from);  
    ensureIndex(to);  
    if (from > to) throw new IndexOutOfBoundsException("Start index (" + from + ") is greater than end index (" + to +  
    ")");  
  
    if (from != 0 || to != 1) return EMPTY_BIG_LIST;  
    return this;  
}
```

```
@Override  
public boolean addAll(long i, Collection<? extends KEY_GENERIC_CLASS> c) { throw new  
UnsupportedOperationException(); }
```

```
@Override  
public boolean addAll(final Collection<? extends KEY_GENERIC_CLASS> c) { throw new  
UnsupportedOperationException(); }
```

```
@Override  
public boolean removeAll(final Collection<?> c) { throw new UnsupportedOperationException(); }
```

```
@Override  
public boolean retainAll(final Collection<?> c) { throw new UnsupportedOperationException(); }
```

```
#if KEYS_PRIMITIVE
```

```

@Override
public boolean addAll(BIG_LIST c) { throw new UnsupportedOperationException(); }

@Override
public boolean addAll(long i, BIG_LIST c) { throw new UnsupportedOperationException(); }

@Override
public boolean addAll(long i, COLLECTION c) { throw new UnsupportedOperationException(); }

@Override
public boolean addAll(final COLLECTION c) { throw new UnsupportedOperationException(); }

@Override
public boolean removeAll(final COLLECTION c) { throw new UnsupportedOperationException(); }

@Override
public boolean retainAll(final COLLECTION c) { throw new UnsupportedOperationException(); }

#endif

@Override
public void clear() { throw new UnsupportedOperationException(); }

@Override
public long size64() { return 1; }

@Override
public Object clone() { return this; }
}

/** Returns a type-specific immutable big list containing only the specified element. The returned big list is
serializable and cloneable.
*
* @param element the only element of the returned big list.
* @return a type-specific immutable big list containing just { @code element }.
*/

public static KEY_GENERIC BIG_LIST KEY_GENERIC singleton(final KEY_GENERIC_TYPE element) {
return new Singleton KEY_GENERIC_DIAMOND(element); }

#if KEYS_PRIMITIVE

/** Returns a type-specific immutable big list containing only the specified element. The returned big list is
serializable and cloneable.
*
* @param element the only element of the returned big list.
* @return a type-specific immutable big list containing just { @code element }.

```

```

*/

public static KEY_GENERIC BIG_LIST KEY_GENERIC singleton(final Object element) { return new Singleton
KEY_GENERIC_DIAMOND(KEY_OBJ2TYPE(element)); }

#endif

/** A synchronized wrapper class for big lists. */

public static class SynchronizedBigList KEY_GENERIC extends COLLECTIONS.SynchronizedCollection
KEY_GENERIC implements BIG_LIST KEY_GENERIC, java.io.Serializable {

private static final long serialVersionUID = -7046029254386353129L;

protected final BIG_LIST KEY_GENERIC list; // Due to the large number of methods that are not in
COLLECTION, this is worth caching.

protected SynchronizedBigList(final BIG_LIST KEY_GENERIC l, final Object sync) {
super(l, sync);
this.list = l;
}

protected SynchronizedBigList(final BIG_LIST KEY_GENERIC l) {
super(l);
this.list = l;
}

@Override
public KEY_GENERIC_TYPE GET_KEY(final long i) { synchronized(sync) { return list.GET_KEY(i); } }

@Override
public KEY_GENERIC_TYPE set(final long i, final KEY_GENERIC_TYPE k) { synchronized(sync) { return
list.set(i, k); } }

@Override
public void add(final long i, final KEY_GENERIC_TYPE k) { synchronized(sync) { list.add(i, k); } }

@Override
public KEY_GENERIC_TYPE REMOVE_KEY(final long i) { synchronized(sync) { return list.REMOVE_KEY(i);
} }

@Override
public long indexOf(final KEY_TYPE k) { synchronized(sync) { return list.indexOf(k); } }

@Override
public long lastIndexOf(final KEY_TYPE k) { synchronized(sync) { return list.lastIndexOf(k); } }

```

```

@Override
public boolean addAll(final long index, final Collection<? extends KEY_GENERIC_CLASS> c) {
synchronized(sync) { return list.addAll(index, c); } }

@Override
public void getElements(final long from, final KEY_TYPE a[], final long offset, final long length) {
synchronized(sync) { list.getElements(from, a, offset, length); } }

@Override
public void removeElements(final long from, final long to) { synchronized(sync) { list.removeElements(from, to); }
}

@Override
public void addElements(long index, final KEY_GENERIC_TYPE a[], long offset, long length) {
synchronized(sync) { list.addElements(index, a, offset, length); } }

@Override
public void addElements(long index, final KEY_GENERIC_TYPE a[]) { synchronized(sync) {
list.addElements(index, a); } }

/* @inheritDoc
 * @deprecated Use {@link #size64()} instead.
 */
@Deprecated
@Override
public void size(final long size) { synchronized(sync) { list.size(size); } }

@Override
public long size64() { synchronized(sync) { return list.size64(); } }

@Override
public KEY_BIG_LIST_ITERATOR KEY_GENERIC iterator() { return list.listIterator(); }

@Override
public KEY_BIG_LIST_ITERATOR KEY_GENERIC listIterator() { return list.listIterator(); }

@Override
public KEY_BIG_LIST_ITERATOR KEY_GENERIC listIterator(final long i) { return list.listIterator(i); }

@Override
public BIG_LIST KEY_GENERIC subList(final long from, final long to) { synchronized(sync) { return
synchronize(list.subList(from, to), sync); } }

@Override
public boolean equals(final Object o) { if (o == this) return true; synchronized(sync) { return list.equals(o); } }

@Override
public int hashCode() { synchronized(sync) { return list.hashCode(); } }

```

```

#if ! KEY_CLASS_Reference
    @Override
    public int compareTo(final BigList<? extends KEY_GENERIC_CLASS> o) { synchronized(sync) { return
list.compareTo(o); } }
#endif

#if KEYS_PRIMITIVE
    @Override
    public boolean addAll(final long index, final COLLECTION c) { synchronized(sync) { return list.addAll(index, c);
} }

    @Override
    public boolean addAll(final long index, BIG_LIST l) { synchronized(sync) { return list.addAll(index, l); } }

    @Override
    public boolean addAll(BIG_LIST l) { synchronized(sync) { return list.addAll(l); } }

    /** {@inheritDoc}
    * @deprecated Please use the corresponding type-specific method instead. */
    @Deprecated
    @Override
    public void add(final long i, KEY_GENERIC_CLASS k) { synchronized(sync) { list.add(i, k); } }

    /** {@inheritDoc}
    * @deprecated Please use the corresponding type-specific method instead. */
    @Deprecated
    @Override
    public KEY_GENERIC_CLASS get(final long i) { synchronized(sync) { return list.get(i); } }

    /** {@inheritDoc}
    * @deprecated Please use the corresponding type-specific method instead. */
    @Deprecated
    @Override
    public KEY_GENERIC_CLASS set(final long index, KEY_GENERIC_CLASS k) { synchronized(sync) { return
list.set(index, k); } }

    /** {@inheritDoc}
    * @deprecated Please use the corresponding type-specific method instead. */
    @Deprecated
    @Override
    public KEY_GENERIC_CLASS remove(final long i) { synchronized(sync) { return list.remove(i); } }

    /** {@inheritDoc}
    * @deprecated Please use the corresponding type-specific method instead. */
    @Deprecated
    @Override
    public long indexOf(final Object o) { synchronized(sync) { return list.indexOf(o); } }

```

```

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public long lastIndexOf(final Object o) { synchronized(sync) { return list.lastIndexOf(o); } }
#endif
}

/** Returns a synchronized type-specific big list backed by the given type-specific big list.
 *
 * @param l the big list to be wrapped in a synchronized big list.
 * @return a synchronized view of the specified big list.
 * @see java.util.Collections#synchronizedList(List)
 */
public static KEY_GENERIC BIG_LIST KEY_GENERIC synchronize(final BIG_LIST KEY_GENERIC l) {
return new SynchronizedBigList KEY_GENERIC_DIAMOND(l); }

/** Returns a synchronized type-specific big list backed by the given type-specific big list, using an assigned object
to synchronize.
 *
 * @param l the big list to be wrapped in a synchronized big list.
 * @param sync an object that will be used to synchronize the access to the big list.
 * @return a synchronized view of the specified big list.
 * @see java.util.Collections#synchronizedList(List)
 */
public static KEY_GENERIC BIG_LIST KEY_GENERIC synchronize(final BIG_LIST KEY_GENERIC l, final
Object sync) { return new SynchronizedBigList KEY_GENERIC_DIAMOND(l, sync); }

/** An unmodifiable wrapper class for big lists. */

public static class UnmodifiableBigList KEY_GENERIC extends COLLECTIONS.UnmodifiableCollection
KEY_GENERIC implements BIG_LIST KEY_GENERIC, java.io.Serializable {

private static final long serialVersionUID = -7046029254386353129L;

protected final BIG_LIST KEY_GENERIC list; // Due to the large number of methods that are not in
COLLECTION, this is worth caching.

protected UnmodifiableBigList(final BIG_LIST KEY_GENERIC l) {
super(l);
this.list = l;
}
}

```

```

@Override
public KEY_GENERIC_TYPE GET_KEY(final long i) { return list.GET_KEY(i); }

@Override
public KEY_GENERIC_TYPE set(final long i, final KEY_GENERIC_TYPE k) { throw new
UnsupportedOperationException(); }

@Override
public void add(final long i, final KEY_GENERIC_TYPE k) { throw new UnsupportedOperationException(); }

@Override
public KEY_GENERIC_TYPE REMOVE_KEY(final long i) { throw new UnsupportedOperationException(); }

@Override
public long indexOf(final KEY_TYPE k) { return list.indexOf(k); }

@Override
public long lastIndexOf(final KEY_TYPE k) { return list.lastIndexOf(k); }

@Override
public boolean addAll(final long index, final Collection<? extends KEY_GENERIC_CLASS> c) { throw new
UnsupportedOperationException(); }

@Override
public void getElements(final long from, final KEY_TYPE a[], final long offset, final long length) {
list.getElements(from, a, offset, length); }

@Override
public void removeElements(final long from, final long to) { throw new UnsupportedOperationException(); }

@Override
public void addElements(long index, final KEY_GENERIC_TYPE a[], long offset, long length) { throw new
UnsupportedOperationException(); }

@Override
public void addElements(long index, final KEY_GENERIC_TYPE a[]) { throw new
UnsupportedOperationException(); }

/* {@inheritDoc}
 * @deprecated Use {@link #size64()} instead.
 */
@Deprecated
@Override
public void size(final long size) { list.size(size); }

@Override
public long size64() { return list.size64(); }

```

```

@Override
public KEY_BIG_LIST_ITERATOR KEY_GENERIC iterator() { return listIterator(); }

@Override
public KEY_BIG_LIST_ITERATOR KEY_GENERIC listIterator() { return
BIG_LIST_ITERATORS.unmodifiable(list.listIterator()); }

@Override
public KEY_BIG_LIST_ITERATOR KEY_GENERIC listIterator(final long i) { return
BIG_LIST_ITERATORS.unmodifiable(list.listIterator(i)); }

@Override
public BIG_LIST KEY_GENERIC subList(final long from, final long to) { return unmodifiable(list.subList(from,
to)); }

@Override
public boolean equals(final Object o) { if (o == this) return true; return list.equals(o); }

@Override
public int hashCode() { return list.hashCode(); }

#if ! KEY_CLASS_Reference
@Override
public int compareTo(final BigList<? extends KEY_GENERIC_CLASS> o) { return list.compareTo(o); }
#endif

#if KEYS_PRIMITIVE
@Override
public boolean addAll(final long index, final COLLECTION c) { throw new UnsupportedOperationException(); }

@Override
public boolean addAll(final BIG_LIST l) { throw new UnsupportedOperationException(); }

@Override
public boolean addAll(final long index, final BIG_LIST l) { throw new UnsupportedOperationException(); }

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public KEY_GENERIC_CLASS get(final long i) { return list.get(i); }

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public void add(final long i, KEY_GENERIC_CLASS k) { throw new UnsupportedOperationException(); }

```

```

/** { @inheritDoc }
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public KEY_GENERIC_CLASS set(final long index, KEY_GENERIC_CLASS k) { throw new
UnsupportedOperationException(); }

/** { @inheritDoc }
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public KEY_GENERIC_CLASS remove(final long i) { throw new UnsupportedOperationException(); }

/** { @inheritDoc }
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public long indexOf(final Object o) { return list.indexOf(o); }

/** { @inheritDoc }
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public long lastIndexOf(final Object o) { return list.lastIndexOf(o); }
#endif
}

/** Returns an unmodifiable type-specific big list backed by the given type-specific big list.
 *
 * @param l the big list to be wrapped in an unmodifiable big list.
 * @return an unmodifiable view of the specified big list.
 * @see java.util.Collections#unmodifiableList(List)
 */
public static KEY_GENERIC BIG_LIST KEY_GENERIC unmodifiable(final BIG_LIST KEY_GENERIC l) {
return new UnmodifiableBigList KEY_GENERIC_DIAMOND(l); }

/** A class exposing a list as a big list. */

public static class ListBigList KEY_GENERIC extends ABSTRACT_BIG_LIST KEY_GENERIC implements
java.io.Serializable {

private static final long serialVersionUID = -7046029254386353129L;

private final LIST KEY_GENERIC list;

protected ListBigList(final LIST KEY_GENERIC list) {
this.list = list;

```

```

    }

    private int intIndex(long index) {
        if (index >= Integer.MAX_VALUE) throw new IndexOutOfBoundsException("This big list is restricted to 32-bit
indices");
        return (int)index;
    }

    @Override
    public long size64() { return list.size(); }

    @Override
    public void size(final long size) { list.size(intIndex(size)); }

    @Override
    public KEY_BIG_LIST_ITERATOR KEY_GENERIC iterator() { return
BIG_LIST_ITERATORS.asBigListIterator(list.iterator()); }

    @Override
    public KEY_BIG_LIST_ITERATOR KEY_GENERIC listIterator() { return
BIG_LIST_ITERATORS.asBigListIterator(list.listIterator()); }

    @Override
    public KEY_BIG_LIST_ITERATOR KEY_GENERIC listIterator(final long index) { return
BIG_LIST_ITERATORS.asBigListIterator(list.listIterator(intIndex(index))); }

    @Override
    public boolean addAll(final long index, final Collection<? extends KEY_GENERIC_CLASS> c) { return
list.addAll(intIndex(index), c); }

    @Override
    public BIG_LIST KEY_GENERIC subList(long from, long to) { return new ListBigList
KEY_GENERIC_DIAMOND(list.subList(intIndex(from), intIndex(to))); }

    @Override
    public boolean contains(final KEY_TYPE key) { return list.contains(key); }

    @Override
    public KEY_TYPE[] TO_KEY_ARRAY() { return list.TO_KEY_ARRAY(); }

    @Override
    public void removeElements(final long from, final long to) { list.removeElements(intIndex(from), intIndex(to)); }

    #if KEYS_PRIMITIVE
    /* {@inheritDoc}
    * @deprecated Please use {@code toArray()} instead&mdash;this method is redundant and will be removed in the
future.
    */

```

```

@Deprecated
@Override
public KEY_TYPE[] TO_KEY_ARRAY(KEY_TYPE[] a) { return list.toArray(a); }

@Override
public boolean addAll(long index, COLLECTION KEY_GENERIC c) { return list.addAll(intIndex(index), c); }

@Override
public boolean addAll(COLLECTION KEY_GENERIC c) { return list.addAll(c); }

@Override
public boolean addAll(long index, BIG_LIST KEY_GENERIC c) { return list.addAll(intIndex(index), c); }

@Override
public boolean addAll(BIG_LIST KEY_GENERIC c) { return list.addAll(c); }

@Override
public boolean containsAll(COLLECTION KEY_GENERIC c) { return list.containsAll(c); }

@Override
public boolean removeAll(COLLECTION KEY_GENERIC c) { return list.removeAll(c); }

@Override
public boolean retainAll(COLLECTION KEY_GENERIC c) { return list.retainAll(c); }
#endif
@Override
public void add(long index, KEY_GENERIC_TYPE key) { list.add(intIndex(index), key); }

@Override
public boolean add(KEY_GENERIC_TYPE key) { return list.add(key); }

@Override
public KEY_GENERIC_TYPE GET_KEY(long index) { return list.GET_KEY(intIndex(index)); }

@Override
public long indexOf(KEY_TYPE k) { return list.indexOf(k); }

@Override
public long lastIndexOf(KEY_TYPE k) { return list.lastIndexOf(k); }

@Override
public KEY_GENERIC_TYPE REMOVE_KEY(long index) { return list.REMOVE_KEY(intIndex(index)); }

@Override
public KEY_GENERIC_TYPE set(long index, KEY_GENERIC_TYPE k) { return list.set(intIndex(index), k); }

@Override
public boolean isEmpty() { return list.isEmpty(); }

```

```

@Override
public <T> T[] toArray(T[] a) { return list.toArray(a); }

@Override
public boolean containsAll(Collection<?> c) { return list.containsAll(c); }

@Override
public boolean addAll(Collection<? extends KEY_GENERIC_CLASS> c) { return list.addAll(c); }

@Override
public boolean removeAll(Collection<?> c) { return list.removeAll(c); }

@Override
public boolean retainAll(Collection<?> c) { return list.retainAll(c); }

@Override
public void clear() { list.clear(); }

@Override
public int hashCode() { return list.hashCode(); }
}

/** Returns a big list backed by the specified list.
 *
 * @param list a list.
 * @return a big list backed by the specified list.
 */
public static KEY_GENERIC BIG_LIST KEY_GENERIC asBigList(final LIST KEY_GENERIC list) { return new
ListBigList KEY_GENERIC_DIAMOND(list); }

#ifdef TEST

private static KEY_TYPE genKey() {
#if KEY_CLASS_Byte || KEY_CLASS_Short || KEY_CLASS_Character
return (KEY_TYPE)(r.nextInt());
#elif KEYS_PRIMITIVE
return r.NEXT_KEY();
#elif KEY_CLASS_Object
return Integer.toBinaryString(r.nextInt());
#else
return new java.io.Serializable() {};
#endif
}
}

```

```

private static void testLists(KEY_TYPE k, BIG_LIST m, BIG_LIST t, int level) {
    int n = 100;
    int c;

    long ms;
    boolean mThrowsIllegal, tThrowsIllegal, mThrowsNoElement, tThrowsNoElement, mThrowsIndex, tThrowsIndex,
    mThrowsUnsupp, tThrowsUnsupp;
    boolean rt = false, rm = false;
    Object Rt = null, Rm = null;

    if (level == 0) return;

    /* Now we check that m and t are equal. */
    if (!m.equals(t) || !t.equals(m)) System.err.println("m: " + m + " t: " + t);

    ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) at start");
    ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) at start");

    /* Now we check that m actually holds that data. */
    for(java.util.Iterator i=t.iterator(); i.hasNext();) {
        ensure(m.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after insertion (iterating on
t)");
    }

    /* Now we check that m actually holds that data, but iterating on m. */
    for(java.util.Iterator i=m.listIterator(); i.hasNext();) {
        ensure(t.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after insertion (iterating on
m)");
    }

    /* Now we check that inquiries about random data give the same answer in m and t. For
m we use the polymorphic method. */

    for(int i=0; i<n; i++) {
        KEY_TYPE T = genKey();

        mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;

        try {
            m.contains(T);
        }
        catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
        catch (IllegalArgumentException e) { mThrowsIllegal = true; }
        catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }

        try {

```

```

    t.contains(KEY2OBJ(T));
}
catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
catch (IllegalArgumentException e) { tThrowsIllegal = true; }
catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }

ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): contains() divergence in
java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ") " + m);
ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): contains() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
ensure(mThrowsIndex == tThrowsIndex, "Error (" + level + ", " + seed + "): contains() divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex) ensure(m.contains(KEY2OBJ(T)) ==
t.contains(KEY2OBJ(T)), "Error (" + level + ", " + seed + "): divergence in keys between t and m (polymorphic
method) " + m);
}

/* Again, we check that inquiries about random data give the same answer in m and t, but
for m we use the standard method. */

for(int i=0; i<n; i++) {
    KEY_TYPE T = genKey();

    mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;

    try {
        m.contains(KEY2OBJ(T));
    }
    catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
    catch (IllegalArgumentException e) { mThrowsIllegal = true; }
    catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
    catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

    try {
        t.contains(KEY2OBJ(T));
    }
    catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
    catch (IllegalArgumentException e) { tThrowsIllegal = true; }
    catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
    catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }

    ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): contains() divergence in
java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ") " + m);
    ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): contains() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
    ensure(mThrowsIndex == tThrowsIndex, "Error (" + level + ", " + seed + "): contains() divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
}

```

```

    ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + level + ", " + seed + "): contains() divergence in
UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);
    if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex && !mThrowsUnsupp)
ensure(m.contains(KEY2OBJ(T)) == t.contains(KEY2OBJ(T)), "Error (" + level + ", " + seed + "): divergence
between t and m (standard method) " + m);
}

/* Now we add and remove random data in m and t, checking that the result is the same. */

for(int i=0; i<20*n; i++) {
    KEY_TYPE T = genKey();

    mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;

    try {
        rm = m.add(KEY2OBJ(T));
    }
    catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
    catch (IllegalArgumentException e) { mThrowsIllegal = true; }
    catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
    catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

    try {
        rt = t.add(KEY2OBJ(T));
    }
    catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
    catch (IllegalArgumentException e) { tThrowsIllegal = true; }
    catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
    catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }

    ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): add() divergence in
java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ") " + m);
    ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): add() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
    ensure(mThrowsIndex == tThrowsIndex, "Error (" + level + ", " + seed + "): add() divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
    ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + level + ", " + seed + "): add() divergence in
UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);
    if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex && !mThrowsUnsupp) ensure(rm == rt, "Error
(" + level + ", " + seed + "): divergence in add() between t and m " + m);

    T = genKey();

    mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;

    try {

```

```

    rm = m.remove(KEY2OBJ(T));
}
catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
catch (IllegalArgumentException e) { mThrowsIllegal = true; }
catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

try {
    rt = t.remove(KEY2OBJ(T));
}
catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
catch (IllegalArgumentException e) { tThrowsIllegal = true; }
catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }

if (!KEY_EQUALS(T, k) && mThrowsUnsupp && !tThrowsUnsupp) mThrowsUnsupp = true; // Stupid bug in
Collections.singleton()

    ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): remove() divergence in
java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ") " + m);
    ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): remove() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
    ensure(mThrowsIndex == tThrowsIndex, "Error (" + level + ", " + seed + "): remove() divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
    ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + level + ", " + seed + "): remove() divergence in
UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);
    if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex && !mThrowsUnsupp) ensure(rm == rt, "Error
(" + level + ", " + seed + "): divergence in remove() between t and m " + m);
}

ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after removal " + m);
ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after removal " + m);

/* Now we add and remove random data in m and t at specific positions, checking that the result is the same. */

for(int i=0; i<20*n; i++) {
    KEY_TYPE T = genKey();

    mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;

    int pos = r.nextInt(2);

    try {
        m.add(pos, KEY2OBJ(T));
    }
    catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
    catch (IllegalArgumentException e) { mThrowsIllegal = true; }

```

```

catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

try {
    t.add(pos, KEY2OBJ(T));
}
catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
catch (IllegalArgumentException e) { tThrowsIllegal = true; }
catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }

ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): add() at " + pos + "
divergence in java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement
+ ") " + m);
ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): add() at " + pos + " divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
ensure(mThrowsIndex == tThrowsIndex, "Error (" + level + ", " + seed + "): add() at " + pos + " divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + level + ", " + seed + "): add() at " + pos + " divergence in
UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);

T = genKey();

mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;

pos = r.nextInt(2);

try {
    Rm = m.remove(pos);
}
catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
catch (IllegalArgumentException e) { mThrowsIllegal = true; }
catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

try {
    Rt = t.remove(pos);
}
catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
catch (IllegalArgumentException e) { tThrowsIllegal = true; }
catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }

ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): remove() at " + pos + "
divergence in java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement
+ ") " + m);

```

```

    ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): remove() at " + pos + " divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
    ensure(mThrowsIndex == tThrowsIndex, "Error (" + level + ", " + seed + "): remove() at " + pos + " divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
    ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + level + ", " + seed + "): remove() at " + pos + " divergence
in UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);
    if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex && !mThrowsUnsupp) ensure(Rm == Rt || Rm
!= null && Rm.equals(Rt), "Error (" + level + ", " + seed + "): divergence in remove() at " + pos + " between t and
m " + m);
}

```

```

ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after removal " + m);
ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after removal " + m);

```

```

/* Now we add and remove random collections in m and t, checking that the result is the same. */

```

```

for(int i=0; i<20*n; i++) {
    KEY_TYPE T = genKey();

```

```

    mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;

```

```

    try {
        rm = m.addAll(java.util.Collections.singleton(KEY2OBJ(T)));
    }
    catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
    catch (IllegalArgumentException e) { mThrowsIllegal = true; }
    catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
    catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

```

```

    try {
        rt = t.addAll(java.util.Collections.singleton(KEY2OBJ(T)));
    }
    catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
    catch (IllegalArgumentException e) { tThrowsIllegal = true; }
    catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
    catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }

```

```

    ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): addAll() divergence in
java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ") " + m);
    ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): addAll() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
    ensure(mThrowsIndex == tThrowsIndex, "Error (" + level + ", " + seed + "): addAll() divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
    ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + level + ", " + seed + "): addAll() divergence in
UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);
    if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex && !mThrowsUnsupp) ensure(rm == rt, "Error
(" + level + ", " + seed + "): divergence in addAll() between t and m " + m);

```

```

T = genKey();

mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;

try {
    rm = m.removeAll(java.util.Collections.singleton(KEY2OBJ(T)));
}
catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
catch (IllegalArgumentException e) { mThrowsIllegal = true; }
catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

try {
    rt = t.removeAll(java.util.Collections.singleton(KEY2OBJ(T)));
}
catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
catch (IllegalArgumentException e) { tThrowsIllegal = true; }
catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }

ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): removeAll() divergence in
java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ") " + m);
ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): removeAll() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
ensure(mThrowsIndex == tThrowsIndex, "Error (" + level + ", " + seed + "): removeAll() divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + level + ", " + seed + "): removeAll() divergence in
UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);
if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex && !mThrowsUnsupp) ensure(rm == rt, "Error
(" + level + ", " + seed + "): divergence in removeAll() between t and m " + m);
}

ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after set removal " + m);
ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after set removal " + m);

/* Now we add random collections at specific positions in m and t, checking that the result is the same. */

for(int i=0; i<20*n; i++) {
    KEY_TYPE T = genKey();

    mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;

    int pos = r.nextInt(2);

```

```

try {
    rm = m.addAll(pos, java.util.Collections.singleton(KEY2OBJ(T)));
}
catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
catch (IllegalArgumentException e) { mThrowsIllegal = true; }
catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

try {
    rt = t.addAll(pos, java.util.Collections.singleton(KEY2OBJ(T)));
}
catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
catch (IllegalArgumentException e) { tThrowsIllegal = true; }
catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }

ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): addAll() at " + pos + "
divergence in java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement
+ ") " + m);
ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): addAll() at " + pos + " divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
ensure(mThrowsIndex == tThrowsIndex, "Error (" + level + ", " + seed + "): addAll() at " + pos + " divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + level + ", " + seed + "): addAll() at " + pos + " divergence
in UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);
if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex && !mThrowsUnsupp) ensure(rm == rt, "Error
(" + level + ", " + seed + "): divergence in addAll() at " + pos + " between t and m " + m);

}

ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after set removal " + m);
ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after set removal " + m);

/* Now we check that m actually holds the same data. */

for(java.util.Iterator i=t.iterator(); i.hasNext();) {
    ensure(m.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after removal (iterating on
t)");
}

/* Now we check that m actually holds that data, but iterating on m. */

for(java.util.Iterator i=m.listIterator(); i.hasNext();) {
    ensure(t.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after removal (iterating on
m)");
}

if (m instanceof Singleton) {

```

```

ensure(m.equals(((Singleton)m).clone()), "Error (" + level + ", " + seed + "): m does not equal m.clone()");
ensure(((Singleton)m).clone().equals(m), "Error (" + level + ", " + seed + "): m.clone() does not equal m");
}

int h = m.hashCode();

/* Now we save and read m. */

BIG_LIST m2 = null;

try {
    java.io.File ff = new java.io.File("it.unimi.dsi.fastutil.test");
    java.io.OutputStream os = new java.io.FileOutputStream(ff);
    java.io.ObjectOutputStream oos = new java.io.ObjectOutputStream(os);

    oos.writeObject(m);
    oos.close();

    java.io.InputStream is = new java.io.FileInputStream(ff);
    java.io.ObjectInputStream ois = new java.io.ObjectInputStream(is);

    m2 = (BIG_LIST)ois.readObject();
    ois.close();
    ff.delete();
}
catch(Exception e) {
    e.printStackTrace();
    System.exit(1);
}

#if ! KEY_CLASS_Reference

ensure(m2.hashCode() == h, "Error (" + level + ", " + seed + "): hashCode() changed after save/read");

/* Now we check that m2 actually holds that data. */

ensure(m2.equals(t), "Error (" + level + ", " + seed + "): ! m2.equals(t) after save/read");
ensure(t.equals(m2), "Error (" + level + ", " + seed + "): ! t.equals(m2) after save/read");
#endif

if (! m.isEmpty()) {
    long start = (r.nextLong() & 0x7FFFFFFFFFFFFFFFL) % m.size64();
    long end = start + (r.nextLong() & 0x7FFFFFFFFFFFFFFFL) % (m.size64() - start);
    //System.err.println("Checking subList from " + start + " to " + end + " (level=" + (level+1) + ")...");
    testLists(k, m.subList(start, end), t.subList(start, end), level - 1);

    ensure(m.equals(t), "Error (" + level + ", " + seed + m + t + "): ! m.equals(t) after subList");
    ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after subList");
}

```

```

    }

    return;
}

private static void test() {
    KEY_TYPE k = genKey();
    BIG_LIST m = new Singleton(k);
    BIG_LIST u = BIG_LISTS.unmodifiable(BIG_LISTS.asBigList(LISTS.singleton(KEY2OBJ(k))));
    testLists(k, m, u, 3);
    System.out.println("Test OK");
}

private static long seed = System.currentTimeMillis();
private static java.util.Random r = new java.util.Random(seed);

private static java.text.NumberFormat format = new java.text.DecimalFormat("#,###.00");
private static java.text.FieldPosition fp = new java.text.FieldPosition(0);

private static String format(double d) {
    StringBuffer s = new StringBuffer();
    return format.format(d, s, fp).toString();
}

private static void fatal(String msg) {
    System.out.println(msg);
    System.exit(1);
}

private static void ensure(boolean cond, String msg) {
    if (cond) return;
    fatal(msg);
}

/** This method expects as first argument a lower-cased type (e.g., "int"),
 * and as second optional argument a seed. */

public static void main(String arg[]) throws Exception {
    if (arg.length > 1) r = new java.util.Random(seed = Long.parseLong(arg[1]));

    try {
        test();
    } catch(Throwable e) {
        e.printStackTrace(System.err);
        System.err.println("seed: " + seed);
    }
}

```

```
#endif
```

```
}
```

Found in path(s):

```
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-  
a775574/drv/BigLists.drv
```

No license file was found, but licenses were detected in source scan.

```
/*
```

```
* Copyright (C) 2002-2017 Sebastiano Vigna
```

```
*
```

```
* Licensed under the Apache License, Version 2.0 (the "License");
```

```
* you may not use this file except in compliance with the License.
```

```
* You may obtain a copy of the License at
```

```
*
```

```
* http://www.apache.org/licenses/LICENSE-2.0
```

```
*
```

```
* Unless required by applicable law or agreed to in writing, software
```

```
* distributed under the License is distributed on an "AS IS" BASIS,
```

```
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
```

```
* See the License for the specific language governing permissions and
```

```
* limitations under the License.
```

```
*/
```

```
package PACKAGE;
```

```
import java.util.List;
```

```
##if ! KEY_CLASS_Reference
```

```
/** A type-specific {@link List}; provides some additional methods that use polymorphism to avoid (un)boxing.
```

```
*
```

```
* <p>Note that this type-specific interface extends {@link Comparable}: it is expected that implementing
```

```
* classes perform a lexicographical comparison using the standard operator "less than" for primitive types,
```

```
* and the usual {@link Comparable#compareTo(Object) compareTo()} method for objects.
```

```
*
```

```
* <p>Additionally, this interface strengthens {@link #listIterator()},
```

```
* {@link #listIterator(int)} and {@link #subList(int,int)}.
```

```
*
```

```
* <p>Besides polymorphic methods, this interfaces specifies methods to copy into an array or remove contiguous
```

```
* sublists. Although the abstract implementation of this interface provides simple, one-by-one implementations
```

```
* of these methods, it is expected that concrete implementation override them with optimized versions.
```

```
*
```

```
* @see List
```

```
*/
```

```

public interface LIST KEY_GENERIC extends List<KEY_GENERIC_CLASS>, Comparable<List<? extends
KEY_GENERIC_CLASS>>, COLLECTION KEY_GENERIC {
#else

/** A type-specific {@link List}; provides some additional methods that use polymorphism to avoid (un)boxing.
*
* <p>Additionally, this interface strengthens {@link #iterator()}, {@link #listIterator()},
* {@link #listIterator(int)} and {@link #subList(int,int)}. The former had been already
* strengthened upstream, but unfortunately {@link List} re-specifies it.
*
* <p>Besides polymorphic methods, this interfaces specifies methods to copy into an array or remove contiguous
* sublists. Although the abstract implementation of this interface provides simple, one-by-one implementations
* of these methods, it is expected that concrete implementation override them with optimized versions.
*
* @see List
*/

public interface LIST KEY_GENERIC extends List<KEY_GENERIC_CLASS>, COLLECTION KEY_GENERIC
{
#endif

/** Returns a type-specific iterator on the elements of this list.
*
* <p>Note that this specification strengthens the one given in {@link List#iterator()}.
* It would not be normally necessary, but {@link java.lang.Iterable#iterator()} is bizarrely re-specified
* in {@link List}.
*
* @return an iterator on the elements of this list.
*/
@Override
KEY_LIST_ITERATOR KEY_GENERIC iterator();

/** Returns a type-specific list iterator on the list.
*
* @see List#listIterator()
*/
@Override
KEY_LIST_ITERATOR KEY_GENERIC listIterator();

/** Returns a type-specific list iterator on the list starting at a given index.
*
* @see List#listIterator(int)
*/
@Override
KEY_LIST_ITERATOR KEY_GENERIC listIterator(int index);

/** Returns a type-specific view of the portion of this list from the index {@code from}, inclusive, to the index

```

```

{ @code to }, exclusive.
*
* <p>Note that this specification strengthens the one given in { @link List#subList(int,int)}.
*
* @see List#subList(int,int)
*/
@Override
LIST KEY_GENERIC subList(int from, int to);

/** Sets the size of this list.
*
* <p>If the specified size is smaller than the current size, the last elements are
* discarded. Otherwise, they are filled with 0/{ @code null}/{ @code false}.
*
* @param size the new size.
*/

void size(int size);

/** Copies (hopefully quickly) elements of this type-specific list into the given array.
*
* @param from the start index (inclusive).
* @param a the destination array.
* @param offset the offset into the destination array where to store the first element copied.
* @param length the number of elements to be copied.
*/
void getElements(int from, KEY_TYPE a[], int offset, int length);

/** Removes (hopefully quickly) elements of this type-specific list.
*
* @param from the start index (inclusive).
* @param to the end index (exclusive).
*/
void removeElements(int from, int to);

/** Add (hopefully quickly) elements to this type-specific list.
*
* @param index the index at which to add elements.
* @param a the array containing the elements.
*/
void addElements(int index, KEY_GENERIC_TYPE a[]);

/** Add (hopefully quickly) elements to this type-specific list.
*
* @param index the index at which to add elements.
* @param a the array containing the elements.
* @param offset the offset of the first element to add.
* @param length the number of elements to add.

```

```

*/
void addElements(int index, KEY_GENERIC_TYPE a[], int offset, int length);

#if KEYS_PRIMITIVE

/** Appends the specified element to the end of this list (optional operation).
 * @see List#add(Object)
 */
@Override
boolean add(KEY_TYPE key);

/** Inserts the specified element at the specified position in this list (optional operation).
 * @see List#add(int,Object)
 */
void add(int index, KEY_TYPE key);

/** { @inheritDoc }
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default void add(int index, KEY_CLASS key) {
    add(index, KEY_CLASS2TYPE(key));
}

/** Inserts all of the elements in the specified type-specific collection into this type-specific list at the specified
position (optional operation).
 * @see List#addAll(int,java.util.Collection)
 */
boolean addAll(int index, COLLECTION c);

/** Inserts all of the elements in the specified type-specific list into this type-specific list at the specified position
(optional operation).
 * @see List#add(int,Object)
 */
boolean addAll(int index, LIST c);

/** Appends all of the elements in the specified type-specific list to the end of this type-specific list (optional
operation).
 * @see List#add(int,Object)
 */
boolean addAll(LIST c);

/** Replaces the element at the specified position in this list with the specified element (optional operation).
 * @see List#set(int,Object)
 */
KEY_TYPE set(int index, KEY_TYPE k);

/** Returns the element at the specified position in this list.

```

```

* @see List#get(int)
*/
KEY_TYPE GET_KEY(int index);

/** Returns the index of the first occurrence of the specified element in this list, or -1 if this list does not contain the
element.
* @see List#indexOf(Object)
*/
int indexOf(KEY_TYPE k);

/** Returns the index of the last occurrence of the specified element in this list, or -1 if this list does not contain the
element.
* @see List#lastIndexOf(Object)
*/
int lastIndexOf(KEY_TYPE k);

/** { @inheritDoc }
* @deprecated Please use the corresponding type-specific method instead.
*/
@Deprecated
@Override
default boolean contains(final Object key) {
    return COLLECTION.super.contains(key);
}

/** { @inheritDoc }
* @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default KEY_GENERIC_CLASS get(int index) {
    return KEY2OBJ(GET_KEY(index));
}

/** { @inheritDoc }
* @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default int indexOf(Object o) {
    return indexOf(KEY_OBJ2TYPE(o));
}

/** { @inheritDoc }
* @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default int lastIndexOf(Object o) {
    return lastIndexOf(KEY_OBJ2TYPE(o));
}

```

```

/** {@inheritDoc}
 * <p>This method specification is a workaround for
 * <a href="http://bugs.java.com/bugdatabase/view_bug.do?bug_id=JDK-8177440">bug 8177440</a>.
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default boolean add(KEY_CLASS k) {
    return add(KEY_CLASS2TYPE(k));
}

/** Removes the element at the specified position in this list (optional operation).
 * @see List#remove(int)
 */
KEY_TYPE REMOVE_KEY(int index);

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead.
 */
@Deprecated
@Override
default boolean remove(final Object key) {
    return COLLECTION.super.remove(key);
}

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default KEY_GENERIC_CLASS remove(int index) {
    return KEY2OBJ(REMOVE_KEY(index));
}

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default KEY_GENERIC_CLASS set(int index, KEY_CLASS k) {
    return KEY2OBJ(set(index, KEY_CLASS2TYPE(k)));
}

#endif

}

Found in path(s):
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-
a775574/drv/List.drv

```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2002-2017 Sebastiano Vigna
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

```
package PACKAGE;
```

```
import it.unimi.dsi.fastutil.BidirectionalIterator;
#if KEYS_PRIMITIVE
import it.unimi.dsi.fastutil.objects.ObjectBidirectionalIterator;
#endif
```

```
/** A type-specific bidirectional iterator; provides an additional method to avoid (un)boxing,
 * and the possibility to skip elements backwards.
 *
 * @see BidirectionalIterator
 */
```

```
#if KEYS_PRIMITIVE
public interface KEY_BIDI_ITERATOR KEY_GENERIC extends KEY_ITERATOR KEY_GENERIC,
ObjectBidirectionalIterator<KEY_GENERIC_CLASS> {
#else
public interface KEY_BIDI_ITERATOR KEY_GENERIC extends KEY_ITERATOR KEY_GENERIC,
BidirectionalIterator<KEY_GENERIC_CLASS> {
#endif
```

```
#if KEYS_PRIMITIVE
```

```
/**
 * Returns the previous element as a primitive type.
 *
 * @return the previous element in the iteration.
 * @see java.util.ListIterator#previous()
 */
```

```

KEY_TYPE PREV_KEY());

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default KEY_GENERIC_CLASS previous() { return KEY_CLASS.valueOf(PREV_KEY()); }

#endif

/** Moves back for the given number of elements.
 *
 * <p>The effect of this call is exactly the same as that of
 * calling {@link #previous()} for {@code n} times (possibly stopping
 * if {@link #hasPrevious()} becomes false).
 *
 * @param n the number of elements to skip back.
 * @return the number of elements actually skipped.
 * @see #previous()
 */
#if KEYS_PRIMITIVE
@Override
#endif
default int back(final int n) {
    int i = n;
    while(i-- != 0 && hasPrevious()) PREV_KEY();
    return n - i - 1;
}

/** {@inheritDoc} */
@Override
default int skip(final int n) {
    return KEY_ITERATOR.super.skip(n);
}
}

```

Found in path(s):

`*/opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/drv/BidirectionalIterator.drv`

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2002-2017 Sebastiano Vigna

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

```
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

```
package PACKAGE;
```

```
/** A type-specific {@link Iterable} that further strengthens the specification of {@link Iterable#iterator()}.
*/
```

```
public interface KEY_BIDI_ITERABLE KEY_GENERIC extends KEY_ITERABLE KEY_GENERIC {
```

```
/** Returns a type-specific {@link it.unimi.dsi.fastutil.BidirectionalIterator}.
```

```
*
```

```
* @return a type-specific bidirectional iterator.
```

```
*/
```

```
@Override
```

```
KEY_BIDI_ITERATOR KEY_GENERIC iterator();
```

```
}
```

Found in path(s):

```
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-
a775574/drv/BidirectionalIterable.drv
```

No license file was found, but licenses were detected in source scan.

```
/*
```

```
* Copyright (C) 2002-2017 Sebastiano Vigna
```

```
*
```

```
* Licensed under the Apache License, Version 2.0 (the "License");
```

```
* you may not use this file except in compliance with the License.
```

```
* You may obtain a copy of the License at
```

```
*
```

```
*   http://www.apache.org/licenses/LICENSE-2.0
```

```
*
```

```
* Unless required by applicable law or agreed to in writing, software
```

```
* distributed under the License is distributed on an "AS IS" BASIS,
```

```
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
```

```
* See the License for the specific language governing permissions and
```

```
* limitations under the License.
```

```
*/
```

```
package PACKAGE;
```

```

import java.util.SortedSet;
import java.util.NoSuchElementException;
#if KEYS_REFERENCE
import java.util.Comparator;
#endif

/** A class providing static methods and objects that do useful things with type-specific sorted sets.
 *
 * @see java.util.Collections
 */

public final class SORTED_SETS {

    private SORTED_SETS() {}

    /** An immutable class representing the empty sorted set and implementing a type-specific set interface.
 *
 * <p>This class may be useful to implement your own in case you subclass
 * a type-specific sorted set.
 */

    public static class EmptySet KEY_GENERIC extends SETS.EmptySet KEY_GENERIC implements
    SORTED_SET KEY_GENERIC, java.io.Serializable, Cloneable {
        private static final long serialVersionUID = -7046029254386353129L;

        protected EmptySet() {}

        @Override
        SUPPRESS_WARNINGS_KEY_UNCHECKED
        public KEY_BIDI_ITERATOR KEY_GENERIC iterator(KEY_GENERIC_TYPE from) { return
        ITERATORS.EMPTY_ITERATOR; }

        @Override
        SUPPRESS_WARNINGS_KEY_UNCHECKED
        public SORTED_SET KEY_GENERIC subSet(KEY_GENERIC_TYPE from, KEY_GENERIC_TYPE to) { return
        EMPTY_SET; }

        @Override
        SUPPRESS_WARNINGS_KEY_UNCHECKED
        public SORTED_SET KEY_GENERIC headSet(KEY_GENERIC_TYPE from) { return EMPTY_SET; }

        @Override
        SUPPRESS_WARNINGS_KEY_UNCHECKED
        public SORTED_SET KEY_GENERIC tailSet(KEY_GENERIC_TYPE to) { return EMPTY_SET; }

        @Override

```

```

public KEY_GENERIC_TYPE FIRST() { throw new NoSuchElementException(); }

@Override
public KEY_GENERIC_TYPE LAST() { throw new NoSuchElementException(); }

@Override
public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() { return null; }

#if KEYS_PRIMITIVE
/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public SORTED_SET KEY_GENERIC subSet(KEY_GENERIC_CLASS from, KEY_GENERIC_CLASS to) {
return EMPTY_SET; }

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public SORTED_SET KEY_GENERIC headSet(KEY_GENERIC_CLASS from) { return EMPTY_SET; }

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public SORTED_SET KEY_GENERIC tailSet(KEY_GENERIC_CLASS to) { return EMPTY_SET; }

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public KEY_GENERIC_CLASS first() { throw new NoSuchElementException(); }

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public KEY_GENERIC_CLASS last() { throw new NoSuchElementException(); }
#endif

@Override
public Object clone() { return EMPTY_SET; }

private Object readResolve() { return EMPTY_SET; }
}

```

```

/** An empty sorted set (immutable). It is serializable and cloneable.
 *
 */
SUPPRESS_WARNINGS_KEY_RAWTYPES
public static final EmptySet EMPTY_SET = new EmptySet();

#if KEYS_REFERENCE
/** Returns an empty sorted set (immutable). It is serializable and cloneable.
 *
 * <p>This method provides a typesafe access to { @link #EMPTY_SET}.
 * @return an empty sorted set (immutable).
 */
@SuppressWarnings("unchecked")
public static KEY_GENERIC SET KEY_GENERIC emptySet() {
    return EMPTY_SET;
}
#endif

/** A class representing a singleton sorted set.
 *
 * <p>This class may be useful to implement your own in case you subclass
 * a type-specific sorted set.
 */

public static class Singleton KEY_GENERIC extends SETS.Singleton KEY_GENERIC implements SORTED_SET
KEY_GENERIC, java.io.Serializable, Cloneable {

    private static final long serialVersionUID = -7046029254386353129L;

    final KEY_COMPARATOR KEY_SUPER_GENERIC comparator;

    protected Singleton(final KEY_GENERIC_TYPE element, final KEY_COMPARATOR KEY_SUPER_GENERIC
comparator) {
        super(element);
        this.comparator = comparator;
    }

    private Singleton(final KEY_GENERIC_TYPE element) {
        this(element, null);
    }

    SUPPRESS_WARNINGS_KEY_UNCHECKED
    final int compare(final KEY_GENERIC_TYPE k1, final KEY_GENERIC_TYPE k2) {
        return comparator == null ? KEY_CMP(k1, k2) : comparator.compare(k1, k2);
    }

    @Override
    public KEY_BIDI_ITERATOR KEY_GENERIC iterator(KEY_GENERIC_TYPE from) {

```

```

KEY_BIDI_ITERATOR KEY_GENERIC i = iterator();
if (compare(element, from) <= 0) i.NEXT_KEY();
return i;
}

@Override
public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() { return comparator; }

@Override
SUPPRESS_WARNINGS_KEY_UNCHECKED
public SORTED_SET KEY_GENERIC subSet(final KEY_GENERIC_TYPE from, final KEY_GENERIC_TYPE
to) { if (compare(from, element) <= 0 && compare(element, to) < 0) return this; return EMPTY_SET; }

@Override
SUPPRESS_WARNINGS_KEY_UNCHECKED
public SORTED_SET KEY_GENERIC headSet(final KEY_GENERIC_TYPE to) { if (compare(element, to) < 0)
return this; return EMPTY_SET; }

@Override
SUPPRESS_WARNINGS_KEY_UNCHECKED
public SORTED_SET KEY_GENERIC tailSet(final KEY_GENERIC_TYPE from) { if (compare(from, element)
<= 0) return this; return EMPTY_SET; }

@Override
public KEY_GENERIC_TYPE FIRST() { return element; }

@Override
public KEY_GENERIC_TYPE LAST() { return element; }

#if KEYS_PRIMITIVE
/** { @inheritDoc }
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public SORTED_SET KEY_GENERIC subSet(final KEY_CLASS from, final KEY_CLASS to) { return
subSet(KEY_CLASS2TYPE(from), KEY_CLASS2TYPE(to)); }

/** { @inheritDoc }
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public SORTED_SET KEY_GENERIC headSet(final KEY_CLASS to) { return headSet(KEY_CLASS2TYPE(to));
}

/** { @inheritDoc }
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override

```

```
public SORTED_SET KEY_GENERIC tailSet(final KEY_CLASS from) { return  
tailSet(KEY_CLASS2TYPE(from)); }
```

```
/** {@inheritDoc}  
 * @deprecated Please use the corresponding type-specific method instead. */  
@Deprecated  
@Override  
public KEY_CLASS first() { return KEY2OBJ(element); }
```

```
/** {@inheritDoc}  
 * @deprecated Please use the corresponding type-specific method instead. */  
@Deprecated  
@Override  
public KEY_CLASS last() { return KEY2OBJ(element); }  
#endif  
}
```

/** Returns a type-specific immutable sorted set containing only the specified element. The returned sorted set is serializable and cloneable.

```
*  
 * @param element the only element of the returned sorted set.  
 * @return a type-specific immutable sorted set containing just { @code element }.  
*/
```

```
public static KEY_GENERIC SORTED_SET KEY_GENERIC singleton(final KEY_GENERIC_TYPE element) {  
return new Singleton KEY_GENERIC_DIAMOND(element);  
}
```

/** Returns a type-specific immutable sorted set containing only the specified element, and using a specified comparator. The returned sorted set is serializable and cloneable.

```
*  
 * @param element the only element of the returned sorted set.  
 * @param comparator the comparator to use in the returned sorted set.  
 * @return a type-specific immutable sorted set containing just { @code element }.  
*/
```

```
public static KEY_GENERIC SORTED_SET KEY_GENERIC singleton(final KEY_GENERIC_TYPE element,  
final KEY_COMPARATOR KEY_SUPER_GENERIC comparator) {  
return new Singleton KEY_GENERIC_DIAMOND(element, comparator);  
}
```

```
#if KEYS_PRIMITIVE
```

/** Returns a type-specific immutable sorted set containing only the specified element. The returned sorted set is serializable and cloneable.

```
*  
 * @param element the only element of the returned sorted set.
```

```

* @return a type-specific immutable sorted set containing just { @code element }.
*/

public static KEY_GENERIC SORTED_SET KEY_GENERIC singleton(final Object element) {
    return new Singleton(KEY_OBJ2TYPE(element));
}

/** Returns a type-specific immutable sorted set containing only the specified element, and using a specified
    comparator. The returned sorted set is serializable and cloneable.
    *
    * @param element the only element of the returned sorted set.
    * @param comparator the comparator to use in the returned sorted set.
    * @return a type-specific immutable sorted set containing just { @code element }.
    */

public static KEY_GENERIC SORTED_SET KEY_GENERIC singleton(final Object element, final
KEY_COMPARATOR KEY_SUPER_GENERIC comparator) {
    return new Singleton(KEY_OBJ2TYPE(element), comparator);
}
#endif

/** A synchronized wrapper class for sorted sets. */

public static class SynchronizedSortedSet KEY_GENERIC extends SETS.SynchronizedSet KEY_GENERIC
implements SORTED_SET KEY_GENERIC, java.io.Serializable {

    private static final long serialVersionUID = -7046029254386353129L;

    protected final SORTED_SET KEY_GENERIC sortedSet;

    protected SynchronizedSortedSet(final SORTED_SET KEY_GENERIC s, final Object sync) {
        super(s, sync);
        sortedSet = s;
    }

    protected SynchronizedSortedSet(final SORTED_SET KEY_GENERIC s) {
        super(s);
        sortedSet = s;
    }

    @Override
    public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() { synchronized(sync) { return
sortedSet.comparator(); } }

    @Override
    public SORTED_SET KEY_GENERIC subSet(final KEY_GENERIC_TYPE from, final KEY_GENERIC_TYPE
to) { return new SynchronizedSortedSet KEY_GENERIC_DIAMOND(sortedSet.subSet(from, to), sync); }

```

```

@Override
public SORTED_SET KEY_GENERIC headSet(final KEY_GENERIC_TYPE to) { return new
SynchronizedSortedSet KEY_GENERIC_DIAMOND(sortedSet.headSet(to), sync); }

@Override
public SORTED_SET KEY_GENERIC tailSet(final KEY_GENERIC_TYPE from) { return new
SynchronizedSortedSet KEY_GENERIC_DIAMOND(sortedSet.tailSet(from), sync); }

@Override
public KEY_BIDI_ITERATOR KEY_GENERIC iterator() { return sortedSet.iterator(); }

@Override
public KEY_BIDI_ITERATOR KEY_GENERIC iterator(final KEY_GENERIC_TYPE from) { return
sortedSet.iterator(from); }

@Override
public KEY_GENERIC_TYPE FIRST() { synchronized(sync) { return sortedSet.FIRST(); } }

@Override
public KEY_GENERIC_TYPE LAST() { synchronized(sync) { return sortedSet.LAST(); } }

#if KEYS_PRIMITIVE
/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public KEY_CLASS first() { synchronized(sync) { return sortedSet.first(); } }

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public KEY_CLASS last() { synchronized(sync) { return sortedSet.last(); } }

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public SORTED_SET KEY_GENERIC subSet(final KEY_CLASS from, final KEY_CLASS to) { return new
SynchronizedSortedSet(sortedSet.subSet(from, to), sync); }

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public SORTED_SET KEY_GENERIC headSet(final KEY_CLASS to) { return new
SynchronizedSortedSet(sortedSet.headSet(to), sync); }

```

```

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public SORTED_SET KEY_GENERIC tailSet(final KEY_CLASS from) { return new
SynchronizedSortedSet(sortedSet.tailSet(from), sync); }
#endif
}

/** Returns a synchronized type-specific sorted set backed by the given type-specific sorted set.
 *
 * @param s the sorted set to be wrapped in a synchronized sorted set.
 * @return a synchronized view of the specified sorted set.
 * @see java.util.Collections#synchronizedSortedSet(SortedSet)
 */
public static KEY_GENERIC SORTED_SET KEY_GENERIC synchronize(final SORTED_SET KEY_GENERIC
s) { return new SynchronizedSortedSet KEY_GENERIC_DIAMOND(s); }

/** Returns a synchronized type-specific sorted set backed by the given type-specific sorted set, using an assigned
object to synchronize.
 *
 * @param s the sorted set to be wrapped in a synchronized sorted set.
 * @param sync an object that will be used to synchronize the access to the sorted set.
 * @return a synchronized view of the specified sorted set.
 * @see java.util.Collections#synchronizedSortedSet(SortedSet)
 */

public static KEY_GENERIC SORTED_SET KEY_GENERIC synchronize(final SORTED_SET KEY_GENERIC
s, final Object sync) { return new SynchronizedSortedSet KEY_GENERIC_DIAMOND(s, sync); }

/** An unmodifiable wrapper class for sorted sets. */

public static class UnmodifiableSortedSet KEY_GENERIC extends SETS.UnmodifiableSet KEY_GENERIC
implements SORTED_SET KEY_GENERIC, java.io.Serializable {

private static final long serialVersionUID = -7046029254386353129L;

protected final SORTED_SET KEY_GENERIC sortedSet;

protected UnmodifiableSortedSet(final SORTED_SET KEY_GENERIC s) {
super(s);
sortedSet = s;
}

```

```

}

@Override
public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() { return sortedSet.comparator(); }

@Override
public SORTED_SET KEY_GENERIC subSet(final KEY_GENERIC_TYPE from, final KEY_GENERIC_TYPE
to) { return new UnmodifiableSortedSet KEY_GENERIC_DIAMOND(sortedSet.subSet(from, to)); }

@Override
public SORTED_SET KEY_GENERIC headSet(final KEY_GENERIC_TYPE to) { return new
UnmodifiableSortedSet KEY_GENERIC_DIAMOND(sortedSet.headSet(to)); }

@Override
public SORTED_SET KEY_GENERIC tailSet(final KEY_GENERIC_TYPE from) { return new
UnmodifiableSortedSet KEY_GENERIC_DIAMOND(sortedSet.tailSet(from)); }

@Override
public KEY_BIDI_ITERATOR KEY_GENERIC iterator() { return
ITERATORS.unmodifiable(sortedSet.iterator()); }

@Override
public KEY_BIDI_ITERATOR KEY_GENERIC iterator(final KEY_GENERIC_TYPE from) { return
ITERATORS.unmodifiable(sortedSet.iterator(from)); }

@Override
public KEY_GENERIC_TYPE FIRST() { return sortedSet.FIRST(); }

@Override
public KEY_GENERIC_TYPE LAST() { return sortedSet.LAST(); }

#if KEYS_PRIMITIVE
/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public KEY_CLASS first() { return sortedSet.first(); }

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public KEY_CLASS last() { return sortedSet.last(); }

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override

```

```

public SORTED_SET KEY_GENERIC subSet(final KEY_GENERIC_CLASS from, final
KEY_GENERIC_CLASS to) { return new UnmodifiableSortedSet(sortedSet.subSet(from, to)); }

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public SORTED_SET KEY_GENERIC headSet(final KEY_GENERIC_CLASS to) { return new
UnmodifiableSortedSet(sortedSet.headSet(to)); }

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public SORTED_SET KEY_GENERIC tailSet(final KEY_GENERIC_CLASS from) { return new
UnmodifiableSortedSet(sortedSet.tailSet(from)); }
#endif
}

/** Returns an unmodifiable type-specific sorted set backed by the given type-specific sorted set.
 *
 * @param s the sorted set to be wrapped in an unmodifiable sorted set.
 * @return an unmodifiable view of the specified sorted set.
 * @see java.util.Collections#unmodifiableSortedSet(SortedSet)
 */
public static KEY_GENERIC SORTED_SET KEY_GENERIC unmodifiable(final SORTED_SET
KEY_GENERIC s) { return new UnmodifiableSortedSet KEY_GENERIC_DIAMOND(s); }

#if defined(TEST) && ! KEY_CLASS_Reference

private static KEY_TYPE genKey() {
#if KEY_CLASS_Byte || KEY_CLASS_Short || KEY_CLASS_Character
return (KEY_TYPE)(r.nextInt());
#elif KEYS_PRIMITIVE
return r.NEXT_KEY();
#elif KEY_CLASS_Object
return Integer.toBinaryString(r.nextInt());
#endif
}

protected static void testSets(KEY_TYPE k, SORTED_SET m, SortedSet t, int level) {
int n = 100;
int c;

```

```

long ms;
boolean mThrowsIllegal, tThrowsIllegal, mThrowsNoElement, tThrowsNoElement, mThrowsIndex, tThrowsIndex,
mThrowsUnsupp, tThrowsUnsupp;
boolean rt = false, rm = false;

if (level == 0) return;

/* Now we check that m and t are equal. */
if (!m.equals(t) || !t.equals(m)) System.err.println("m: " + m + " t: " + t);

ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) at start");
ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) at start");

/* Now we check that m actually holds that data. */
for(java.util.Iterator i=t.iterator(); i.hasNext();) {
    ensure(m.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after insertion (iterating on
t)");
}

/* Now we check that m actually holds that data, but iterating on m. */
for(java.util.Iterator i=m.iterator(); i.hasNext();) {
    ensure(t.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after insertion (iterating on
m)");
}

/* Now we check that inquiries about random data give the same answer in m and t. For
m we use the polymorphic method. */

for(int i=0; i<n; i++) {
    KEY_TYPE T = genKey();

    mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;

    try {
        m.contains(T);
    }
    catch (NoSuchElementException e) { mThrowsNoElement = true; }
    catch (IllegalArgumentException e) { mThrowsIllegal = true; }
    catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }

    try {
        t.contains(KEY2OBJ(T));
    }
    catch (NoSuchElementException e) { tThrowsNoElement = true; }
    catch (IllegalArgumentException e) { tThrowsIllegal = true; }
    catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
}

```

```

    ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): contains() divergence in
NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ") " + m);
    ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): contains() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
    ensure(mThrowsIndex == tThrowsIndex, "Error (" + level + ", " + seed + "): contains() divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
    if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex) ensure(m.contains(KEY2OBJ(T)) ==
t.contains(KEY2OBJ(T)), "Error (" + level + ", " + seed + "): divergence in keys between t and m (polymorphic
method) " + m);
}

/* Again, we check that inquiries about random data give the same answer in m and t, but
for m we use the standard method. */

for(int i=0; i<n; i++) {
    KEY_TYPE T = genKey();

    mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;

    try {
        m.contains(KEY2OBJ(T));
    }
    catch (NoSuchElementException e) { mThrowsNoElement = true; }
    catch (IllegalArgumentException e) { mThrowsIllegal = true; }
    catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
    catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

    try {
        t.contains(KEY2OBJ(T));
    }
    catch (NoSuchElementException e) { tThrowsNoElement = true; }
    catch (IllegalArgumentException e) { tThrowsIllegal = true; }
    catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
    catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }

    ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): contains() divergence in
NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ") " + m);
    ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): contains() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
    ensure(mThrowsIndex == tThrowsIndex, "Error (" + level + ", " + seed + "): contains() divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
    ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + level + ", " + seed + "): contains() divergence in
UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);
    if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex && !mThrowsUnsupp)
ensure(m.contains(KEY2OBJ(T)) == t.contains(KEY2OBJ(T)), "Error (" + level + ", " + seed + "): divergence
between t and m (standard method) " + m);
}

```

```
/* Now we add and remove random data in m and t, checking that the result is the same. */
```

```
for(int i=0; i<20*n; i++) {  
    KEY_TYPE T = genKey();
```

```
    mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =  
    mThrowsUnsupp = tThrowsUnsupp = false;
```

```
    try {  
        rm = m.add(KEY2OBJ(T));  
    }  
    catch (NoSuchElementException e) { mThrowsNoElement = true; }  
    catch (IllegalArgumentException e) { mThrowsIllegal = true; }  
    catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }  
    catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }
```

```
    try {  
        rt = t.add(KEY2OBJ(T));  
    }  
    catch (NoSuchElementException e) { tThrowsNoElement = true; }  
    catch (IllegalArgumentException e) { tThrowsIllegal = true; }  
    catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }  
    catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }
```

```
    ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): add() divergence in  
    NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ") " + m);  
    ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): add() divergence in  
    IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);  
    ensure(mThrowsIndex == tThrowsIndex, "Error (" + level + ", " + seed + "): add() divergence in  
    IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);  
    ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + level + ", " + seed + "): add() divergence in  
    UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);  
    if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex && !mThrowsUnsupp) ensure(rm == rt, "Error  
    (" + level + ", " + seed + "): divergence in add() between t and m " + m);
```

```
    T = genKey();
```

```
    mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =  
    mThrowsUnsupp = tThrowsUnsupp = false;
```

```
    try {  
        rm = m.remove(KEY2OBJ(T));  
    }  
    catch (NoSuchElementException e) { mThrowsNoElement = true; }  
    catch (IllegalArgumentException e) { mThrowsIllegal = true; }  
    catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }  
    catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }
```

```

try {
    rt = t.remove(KEY2OBJ(T));
}
catch (NoSuchElementException e) { tThrowsNoElement = true; }
catch (IllegalArgumentException e) { tThrowsIllegal = true; }
catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }

if (! KEY_EQUALS(T, k) && ! mThrowsUnsupp && tThrowsUnsupp) mThrowsUnsupp = false; // Stupid bug in
Collections.singleton()

ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): remove() divergence in
NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ") " + m);
ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): remove() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
ensure(mThrowsIndex == tThrowsIndex, "Error (" + level + ", " + seed + "): remove() divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
// This cannot be possibly made work without a lot of fuss
//ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + level + ", " + seed + "): remove() divergence in
UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);
if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex && !mThrowsUnsupp) ensure(rm == rt, "Error
(" + level + ", " + seed + "): divergence in remove() between t and m " + m);
}

ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after removal " + m);
ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after removal " + m);

/* Now we add and remove random collections in m and t, checking that the result is the same. */

for(int i=0; i<20*n; i++) {
    KEY_TYPE T = genKey();

    mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;

    try {
        rm = m.addAll(java.util.Collections.singleton(KEY2OBJ(T)));
    }
    catch (NoSuchElementException e) { mThrowsNoElement = true; }
    catch (IllegalArgumentException e) { mThrowsIllegal = true; }
    catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
    catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

    try {
        rt = t.addAll(java.util.Collections.singleton(KEY2OBJ(T)));
    }
    catch (NoSuchElementException e) { tThrowsNoElement = true; }

```

```

catch (IllegalArgumentException e) { tThrowsIllegal = true; }
catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }

ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): addAll() divergence in
NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ") " + m);
ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): addAll() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
ensure(mThrowsIndex == tThrowsIndex, "Error (" + level + ", " + seed + "): addAll() divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + level + ", " + seed + "): addAll() divergence in
UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);
if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex && !mThrowsUnsupp) ensure(rm == rt, "Error
(" + level + ", " + seed + "): divergence in addAll() between t and m " + m);

```

```
T = genKey();
```

```
mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;
```

```

try {
    rm = m.removeAll(java.util.Collections.singleton(KEY2OBJ(T)));
}
catch (NoSuchElementException e) { mThrowsNoElement = true; }
catch (IllegalArgumentException e) { mThrowsIllegal = true; }
catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

```

```

try {
    rt = t.removeAll(java.util.Collections.singleton(KEY2OBJ(T)));
}
catch (NoSuchElementException e) { tThrowsNoElement = true; }
catch (IllegalArgumentException e) { tThrowsIllegal = true; }
catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }

```

```

ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): removeAll() divergence in
NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ") " + m);
ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): removeAll() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
ensure(mThrowsIndex == tThrowsIndex, "Error (" + level + ", " + seed + "): removeAll() divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + level + ", " + seed + "): removeAll() divergence in
UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);
if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex && !mThrowsUnsupp) ensure(rm == rt, "Error
(" + level + ", " + seed + "): divergence in removeAll() between t and m " + m);
}

```

```

ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after set removal " + m);
ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after set removal " + m);

/* Now we check that m actually holds the same data. */

for(java.util.Iterator i=t.iterator(); i.hasNext();) {
    ensure(m.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after removal (iterating on
t)");
}

/* Now we check that m actually holds that data, but iterating on m. */

for(java.util.Iterator i=m.iterator(); i.hasNext();) {
    ensure(t.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after removal (iterating on
m)");
}

if (m instanceof Singleton) {
    ensure(m.equals(((Singleton)m).clone()), "Error (" + level + ", " + seed + "): m does not equal m.clone()");
    ensure(((Singleton)m).clone().equals(m), "Error (" + level + ", " + seed + "): m.clone() does not equal m");
}

int h = m.hashCode();

/* Now we save and read m. */

SORTED_SET m2 = null;

try {
    java.io.File ff = new java.io.File("it.unimi.dsi.fastutil.test");
    java.io.OutputStream os = new java.io.FileOutputStream(ff);
    java.io.ObjectOutputStream oos = new java.io.ObjectOutputStream(os);

    oos.writeObject(m);
    oos.close();

    java.io.InputStream is = new java.io.FileInputStream(ff);
    java.io.ObjectInputStream ois = new java.io.ObjectInputStream(is);

    m2 = (SORTED_SET)ois.readObject();
    ois.close();
    ff.delete();
}
catch(Exception e) {
    e.printStackTrace();
    System.exit(1);
}

```

```

#if ! KEY_CLASS_Reference

ensure(m2.hashCode() == h, "Error (" + level + ", " + seed + "): hashCode() changed after save/read");

/* Now we check that m2 actually holds that data. */

ensure(m2.equals(t), "Error (" + level + ", " + seed + "): ! m2.equals(t) after save/read");
ensure(t.equals(m2), "Error (" + level + ", " + seed + "): ! t.equals(m2) after save/read");
#endif

/* Now we select a pair of keys and create a subset. */

if (! m.isEmpty()) {
    java.util.ListIterator i;
    Object start = m.first(), end = m.first();
    for(i = (java.util.ListIterator)m.iterator(); i.hasNext() && r.nextBoolean(); start = end = i.next());
    for(; i.hasNext() && r.nextBoolean(); end = i.next());

    //System.err.println("Checking subSet from " + start + " to " + end + " (level=" + (level+1) + ")...");
    testSets(k, (SORTED_SET)m.subSet((KEY_CLASS)start, (KEY_CLASS)end), t.subSet(start, end), level - 1);

    ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after subSet");
    ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after subSet");

    //System.err.println("Checking headSet to " + end + " (level=" + (level+1) + ")...");
    testSets(k, (SORTED_SET)m.headSet((KEY_CLASS)end), t.headSet(end), level - 1);

    ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after headSet");
    ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after headSet");

    //System.err.println("Checking tailSet from " + start + " (level=" + (level+1) + ")...");
    testSets(k, (SORTED_SET)m.tailSet((KEY_CLASS)start), t.tailSet(start), level - 1);

    ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after tailSet");
    ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after tailSet");
}

return;
}

private static void test() {
    KEY_TYPE k = genKey();
    Singleton m = new Singleton(k);
    SortedSet u = new java.util.TreeSet();
    u.add(KEY2OBJ(k));
    testSets(k, m, java.util.Collections.unmodifiableSortedSet(u), 2);
    System.out.println("Test OK");
}

```

```

}

private static long seed = System.currentTimeMillis();
private static java.util.Random r = new java.util.Random(seed);

private static java.text.NumberFormat format = new java.text.DecimalFormat("#,###.00");
private static java.text.FieldPosition fp = new java.text.FieldPosition(0);

private static String format(double d) {
    StringBuffer s = new StringBuffer();
    return format.format(d, s, fp).toString();
}

private static void fatal(String msg) {
    System.out.println(msg);
    System.exit(1);
}

private static void ensure(boolean cond, String msg) {
    if (cond) return;
    fatal(msg);
}

/** This method expects as first argument a lower-cased type (e.g., "int"),
 * and as second optional argument a seed. */

public static void main(String arg[]) throws Exception {
    if (arg.length > 1) r = new java.util.Random(seed = Long.parseLong(arg[1]));

    try {
        test();
    } catch(Throwable e) {
        e.printStackTrace(System.err);
        System.err.println("seed: " + seed);
    }
}

#endif

}

```

Found in path(s):

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/drv/SortedSets.drv

No license file was found, but licenses were detected in source scan.

/*

```

* Copyright (C) 2002-2017 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

```

```
package PACKAGE;
```

```
import java.util.ListIterator;
```

```

/** A type-specific bidirectional iterator that is also a { @link ListIterator}.
*
* <p>This interface merges the methods provided by a { @link ListIterator} and
* a type-specific { @link it.unimi.dsi.fastutil.BidirectionalIterator}. Moreover, it provides
* type-specific versions of { @link ListIterator#add(Object) add()}
* and { @link ListIterator#set(Object) set()}.
*
* @see java.util.ListIterator
* @see it.unimi.dsi.fastutil.BidirectionalIterator
*/

```

```
public interface KEY_LIST_ITERATOR KEY_GENERIC extends KEY_BIDI_ITERATOR KEY_GENERIC,
ListIterator<KEY_GENERIC_CLASS> {
```

```

/**
* Replaces the last element returned by { @link #next} or
* { @link #previous} with the specified element (optional operation).
* @param k the element used to replace the last element returned.
*
* <p>This default implementation just throws an { @link UnsupportedOperationException}.
* @see ListIterator#set(Object)
*/

```

```
#if KEYS_REFERENCE
```

```
@Override
```

```
#endif
```

```
default void set(final KEY_GENERIC_TYPE k) { throw new UnsupportedOperationException(); }
```

```

/**
 * Inserts the specified element into the list (optional operation).
 *
 * <p>This default implementation just throws an { @link UnsupportedOperationException }.
 * @param k the element to insert.
 * @see ListIterator#add(Object)
 */

#if KEYS_REFERENCE
@Override
#endif
default void add(final KEY_GENERIC_TYPE k) { throw new UnsupportedOperationException(); }

/**
 * Removes from the underlying collection the last element returned
 * by this iterator (optional operation).
 *
 * <p>This default implementation just throws an { @link UnsupportedOperationException }.
 * @see ListIterator#remove()
 */

@Override
default void remove() { throw new UnsupportedOperationException(); }

#if KEYS_PRIMITIVE
/** { @inheritDoc }
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default void set(final KEY_CLASS k) { set(k.KEY_VALUE()); }

/** { @inheritDoc }
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default void add(final KEY_CLASS k) { add(k.KEY_VALUE()); }

/** { @inheritDoc }
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default KEY_GENERIC_CLASS next() { return KEY_BIDI_ITERATOR.super.next(); }

/** { @inheritDoc }
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated

```

```

@Override
default KEY_GENERIC_CLASS previous() { return KEY_BIDI_ITERATOR.super.previous(); }
#endif

}

```

Found in path(s):

```

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-
a775574/drv/ListIterator.drv

```

No license file was found, but licenses were detected in source scan.

```

/*
* Copyright (C) 2002-2017 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

```

```

package PACKAGE;

```

```

import it.unimi.dsi.fastutil.objects.AbstractObjectList;
import it.unimi.dsi.fastutil.objects.ObjectListIterator;
import it.unimi.dsi.fastutil.longs.LongArrays;

```

```

import java.io.Serializable;
import java.util.Iterator;
import java.util.Collection;
import java.util.NoSuchElementException;
import java.util.RandomAccess;

```

```

/** Compact storage of lists of arrays using front coding.
*
* <p>This class stores immutably a list of arrays in a single large array
* using front coding (of course, the compression will be reasonable only if
* the list is sorted lexicographically&mdash;see below). It implements an
* immutable type-specific list that returns the <var>i</var>-th array when
* calling { @link #get(int) get(<var>i</var>)}. The returned array may be
* freely modified.

```

*
 * <p>Front coding is based on the idea that if the <var>i</var>-th and the
 * (<var>i</var>+1)-th array have a common prefix, we might store the length
 * of the common prefix, and then the rest of the second array.
 *
 * <p>This approach, of course, requires that once in a while an array is
 * stored entirely. The ratio of a front-coded list defines how
 * often this happens (once every { @link #ratio() } arrays). A higher ratio
 * means more compression, but means also a longer access time, as more arrays
 * have to be probed to build the result. Note that we must build an array
 * every time { @link #get(int) } is called, but this class provides also methods
 * that extract one of the stored arrays in a given array, reducing garbage
 * collection. See the documentation of the family of { @code get() }
 * methods.
 *
 * <p>By setting the ratio to 1 we actually disable front coding: however, we
 * still have a data structure storing large list of arrays with a reduced
 * overhead (just one integer per array, plus the space required for lengths).
 *
 * <p>Note that the typical usage of front-coded lists is under the form of
 * serialized objects; usually, the data that has to be compacted is processed
 * offline, and the resulting structure is stored permanently. Since the
 * pointer array is not stored, the serialized format is very small.
 *
 * <H2>Implementation Details</H2>
 *
 * <p>All arrays are stored in a { @linkplain it.unimi.dsi.fastutil.BigArrays big array }. A separate array of pointers
 * indexes arrays whose position is a multiple of the ratio: thus, a higher ratio
 * means also less pointers.
 *
 * <p>More in detail, an array whose position is a multiple of the ratio is
 * stored as the array length, followed by the elements of the array. The array
 * length is coded by a simple variable-length list of <var>k</var>-1 bit
 * blocks, where <var>k</var> is the number of bits of the underlying primitive
 * type. All other arrays are stored as follows: let { @code common } the
 * length of the maximum common prefix between the array and its predecessor.
 * Then we store the array length decremented by { @code common }, followed
 * by { @code common }, followed by the array elements whose index is
 * greater than or equal to { @code common }. For instance, if we store
 * { @code foo }, { @code foobar }, { @code football } and
 * { @code fool } in a front-coded character-array list with ratio 3, the
 * character array will contain
 *
 * <pre>
 * 3 f o o 3 3 b a r 5 3 t b a l l 4 f o o l
 * </pre>
 */

```

public class ARRAY_FRONT_CODED_LIST extends AbstractObjectList<KEY_TYPE[]> implements
Serializable, Cloneable, RandomAccess {

    private static final long serialVersionUID = 1L;

    /** The number of arrays in the list. */
    protected final int n;
    /** The ratio of this front-coded list. */
    protected final int ratio;
    /** The big array containing the compressed arrays. */
    protected final KEY_TYPE[][] array;
    /** The pointers to entire arrays in the list. */
    protected transient long[] p;

    /** Creates a new front-coded list containing the arrays returned by the given iterator.
     *
     * @param arrays an iterator returning arrays.
     * @param ratio the desired ratio.
     */

    public ARRAY_FRONT_CODED_LIST(final Iterator<KEY_TYPE[]> arrays, final int ratio) {

        if (ratio < 1) throw new IllegalArgumentException("Illegal ratio (" + ratio + ")");

        KEY_TYPE[][] array = BIG_ARRAYS.EMPTY_BIG_ARRAY;
        long[] p = LongArrays.EMPTY_ARRAY;

        KEY_TYPE[][] a = new KEY_TYPE[2][];
        long curSize = 0;
        int n = 0, b = 0, common, length, minLength;

        while(arrays.hasNext()) {
            a[b] = arrays.next();
            length = a[b].length;

            if (n % ratio == 0) {
                p = LongArrays.grow(p, n / ratio + 1);
                p[n / ratio] = curSize;

                array = BIG_ARRAYS.grow(array, curSize + count(length) + length, curSize);
                curSize += writeInt(array, length, curSize);
                BIG_ARRAYS.copyToBig(a[b], 0, array, curSize, length);
                curSize += length;
            }
            else {
                minLength = a[1 - b].length;
                if (length < minLength) minLength = length;
                for(common = 0; common < minLength; common++) if (a[0][common] != a[1][common]) break;
            }
        }
    }
}

```

```

length -= common;

array = BIG_ARRAYS.grow(array, curSize + count(length) + count(common) + length, curSize);
curSize += writeInt(array, length, curSize);
curSize += writeInt(array, common, curSize);
BIG_ARRAYS.copyToBig(a[b], common, array, curSize, length);
curSize += length;
}

b = 1 - b;
n++;
}

this.n = n;
this.ratio = ratio;
this.array = BIG_ARRAYS.trim(array, curSize);
this.p = LongArrays.trim(p, (n + ratio - 1) / ratio);

}

/** Creates a new front-coded list containing the arrays in the given collection.
 *
 * @param c a collection containing arrays.
 * @param ratio the desired ratio.
 */

public ARRAY_FRONT_CODED_LIST(final Collection<KEY_TYPE[]> c, final int ratio) {
    this(c.iterator(), ratio);
}

/* The following (rather messy) methods implements the encoding of arbitrary integers inside a big array.
 * Unfortunately, we have to specify different codes for almost every type. */

/** Reads a coded length.
 * @param a the data big array.
 * @param pos the starting position.
 * @return the length coded at { @code pos }.
 */

private static int readInt(final KEY_TYPE a[], long pos) {
#if KEY_CLASS_Integer
    return IntBigArrays.get(a, pos);
#elif KEY_CLASS_Long
    return (int)LongBigArrays.get(a, pos);
#elif KEY_CLASS_Character
    final char c0 = CharBigArrays.get(a, pos);
    return c0 < 0x8000 ? c0 : (c0 & 0x7FFF) << 16 | CharBigArrays.get(a, pos + 1);

```

```

#elif KEY_CLASS_Short
    final short s0 = ShortBigArrays.get(a, pos);
    return s0 >= 0 ? s0 : s0 << 16 | (ShortBigArrays.get(a, pos + 1) & 0xFFFF);
#else
    final byte b0 = ByteBigArrays.get(a, pos);
    if (b0 >= 0) return b0;
    final byte b1 = ByteBigArrays.get(a, pos + 1);
    if (b1 >= 0) return (- b0 - 1) << 7 | b1;
    final byte b2 = ByteBigArrays.get(a, pos + 2);
    if (b2 >= 0) return (- b0 - 1) << 14 | (- b1 - 1) << 7 | b2;
    final byte b3 = ByteBigArrays.get(a, pos + 3);
    if (b3 >= 0) return (- b0 - 1) << 21 | (- b1 - 1) << 14 | (- b2 - 1) << 7 | b3;
    return (- b0 - 1) << 28 | (- b1 - 1) << 21 | (- b2 - 1) << 14 | (- b3 - 1) << 7 | ByteBigArrays.get(a, pos + 4);
#endif
}

/** Computes the number of elements coding a given length.
 * @param length the length to be coded.
 * @return the number of elements coding {@code length}.
 */
private static int count(final int length) {
#if KEY_CLASS_Integer || KEY_CLASS_Long
    return 1;
#elif KEY_CLASS_Character || KEY_CLASS_Short
    return length < (1 << 15) ? 1 : 2;
#else
    if (length < (1 << 7)) return 1;
    if (length < (1 << 14)) return 2;
    if (length < (1 << 21)) return 3;
    if (length < (1 << 28)) return 4;
    return 5;
#endif
}

/** Writes a length.
 * @param a the data array.
 * @param length the length to be written.
 * @param pos the starting position.
 * @return the number of elements coding {@code length}.
 */
private static int writeInt(final KEY_TYPE a[], int length, long pos) {
#if KEY_CLASS_Long
    LongBigArrays.set(a, pos, length);
    return 1;
#elif KEY_CLASS_Integer
    IntBigArrays.set(a, pos, length);
    return 1;
#elif KEY_CLASS_Character

```

```

if (length < (1 << 15)) {
    CharBigArrays.set(a, pos, (char)length);
    return 1;
}
CharBigArrays.set(a, pos++, (char)(length >>> 16 | 0x8000));
CharBigArrays.set(a, pos, (char)(length & 0xFFFF));
return 2;
#eliff KEY_CLASS_Short
if (length < (1 << 15)) {
    ShortBigArrays.set(a, pos, (short)length);
    return 1;
}
ShortBigArrays.set(a, pos++, (short)(- (length >>> 16) - 1));
ShortBigArrays.set(a, pos, (short)(length & 0xFFFF));
return 2;
#else
final int count = count(length);
ByteBigArrays.set(a, pos + count - 1, (byte)(length & 0x7F));

if (count != 1) {
    int i = count - 1;
    while(i-- != 0) {
        length >>>= 7;
        ByteBigArrays.set(a, pos + i, (byte)(- (length & 0x7F) - 1));
    }
}

return count;
#endif
}

/** Returns the ratio of this list.
 *
 * @return the ratio of this list.
 */

public int ratio() {
    return ratio;
}

/** Computes the length of the array at the given index.
 *
 * <p>This private version of { @link #arrayLength(int) } does not check its argument.
 *
 * @param index an index.

```

```

* @return the length of the {@code index}-th array.
*/
private int length(final int index) {
    final KEY_TYPE[][] array = this.array;
    final int delta = index % ratio; // The index into the p array, and the delta inside the block.

    long pos = p[index / ratio]; // The position into the array of the first entire word before the index-th.
    int length = readInt(array, pos);

    if (delta == 0) return length;

    // First of all, we recover the array length and the maximum amount of copied elements.
    int common;
    pos += count(length) + length;
    length = readInt(array, pos);
    common = readInt(array, pos + count(length));

    for(int i = 0; i < delta - 1; i++) {
        pos += count(length) + count(common) + length;
        length = readInt(array, pos);
        common = readInt(array, pos + count(length));
    }

    return length + common;
}

/** Computes the length of the array at the given index.
 *
 * @param index an index.
 * @return the length of the {@code index}-th array.
 */
public int arrayLength(final int index) {
    ensureRestrictedIndex(index);
    return length(index);
}

/** Extracts the array at the given index.
 *
 * @param index an index.
 * @param a the array that will store the result (we assume that it can hold the result).
 * @param offset an offset into {@code a} where elements will be store.
 * @param length a maximum number of elements to store in {@code a}.
 * @return the length of the extracted array.
 */
private int extract(final int index, final KEY_TYPE a[], final int offset, final int length) {
    final int delta = index % ratio; // The delta inside the block.
    final long startPos = p[index / ratio]; // The position into the array of the first entire word before the index-th.

```

```

long pos, prevArrayPos;
int arrayLength = readInt(array, pos = startPos), currLen = 0, actualCommon;

if (delta == 0) {
    pos = p[index / ratio] + count(arrayLength);
    BIG_ARRAYS.copyFromBig(array, pos, a, offset, Math.min(length, arrayLength));
    return arrayLength;
}

int common = 0;

for(int i = 0; i < delta; i++) {
    prevArrayPos = pos + count(arrayLength) + (i != 0 ? count(common) : 0);
    pos = prevArrayPos + arrayLength;

    arrayLength = readInt(array, pos);
    common = readInt(array, pos + count(arrayLength));

    actualCommon = Math.min(common, length);
    if (actualCommon <= currLen) currLen = actualCommon;
    else {
        BIG_ARRAYS.copyFromBig(array, prevArrayPos, a, currLen + offset, actualCommon - currLen);
        currLen = actualCommon;
    }
}

if (currLen < length) BIG_ARRAYS.copyFromBig(array, pos + count(arrayLength) + count(common), a, currLen +
offset, Math.min(arrayLength, length - currLen));

return arrayLength + common;
}

/** {@inheritDoc}
 * <p>This implementation delegates to {@link #getArray(int)}. */
@Override
public KEY_TYPE[] get(final int index) {
    return getArray(index);
}

/** Returns an array stored in this front-coded list.
 *
 * @param index an index.
 * @return the corresponding array stored in this front-coded list.
 */
public KEY_TYPE[] getArray(final int index) {
    ensureRestrictedIndex(index);
    final int length = length(index);
    final KEY_TYPE a[] = new KEY_TYPE[length];

```

```

extract(index, a, 0, length);
return a;
}

/** Stores in the given array elements from an array stored in this front-coded list.
 *
 * @param index an index.
 * @param a the array that will store the result.
 * @param offset an offset into { @code a } where elements will be store.
 * @param length a maximum number of elements to store in { @code a }.
 * @return if { @code a } can hold the extracted elements, the number of extracted elements;
 * otherwise, the number of remaining elements with the sign changed.
 */
public int get(final int index, final KEY_TYPE[] a, final int offset, final int length) {
    ensureRestrictedIndex(index);
    ARRAYS.ensureOffsetLength(a, offset, length);

    final int arrayLength = extract(index, a, offset, length);
    if (length >= arrayLength) return arrayLength;
    return length - arrayLength;
}

/** Stores in the given array an array stored in this front-coded list.
 *
 * @param index an index.
 * @param a the array that will store the content of the result (we assume that it can hold the result).
 * @return if { @code a } can hold the extracted elements, the number of extracted elements;
 * otherwise, the number of remaining elements with the sign changed.
 */
public int get(final int index, final KEY_TYPE[] a) {
    return get(index, a, 0, a.length);
}

@Override
public int size() {
    return n;
}

@Override
public ObjectListIterator<KEY_TYPE[]> listIterator(final int start) {
    ensureIndex(start);

    return new ObjectListIterator<KEY_TYPE[]>() {
        KEY_TYPE s[] = ARRAYS.EMPTY_ARRAY;
        int i = 0;
        long pos = 0;
        boolean inSync; // Whether the current value in a is the string just before the next to be produced.
    };
}

```

```

{
  if (start != 0) {
    if (start == n) i = start; // If we start at the end, we do nothing.
    else {
      pos = p[start / ratio];
      int j = start % ratio;
      i = start - j;
      while(j-- != 0) next();
    }
  }
}

@Override
public boolean hasNext() {
  return i < n;
}

@Override
public boolean hasPrevious() {
  return i > 0;
}

@Override
public int previousIndex() {
  return i - 1;
}

@Override
public int nextIndex() {
  return i;
}

@Override
public KEY_TYPE[] next() {
  int length, common;

  if (! hasNext()) throw new NoSuchElementException();

  if (i % ratio == 0) {
    pos = p[i / ratio];
    length = readInt(array, pos);
    s = ARRAYS.ensureCapacity(s, length, 0);
    BIG_ARRAYS.copyFromBig(array, pos + count(length), s, 0, length);
    pos += length + count(length);
    inSync = true;
  }
  else {
    if (inSync) {

```

```

length = readInt(array, pos);
common = readInt(array, pos + count(length));
s = ARRAYS.ensureCapacity(s, length + common, common);
BIG_ARRAYS.copyFromBig(array, pos + count(length) + count (common), s, common, length);
pos += count(length) + count(common) + length;
length += common;
}
else {
s = ARRAYS.ensureCapacity(s, length = length(i), 0);
extract(i, s, 0, length);
}
}
i++;
return ARRAYS.copy(s, 0, length);
}

@Override
public KEY_TYPE[] previous() {
if (! hasPrevious()) throw new NoSuchElementException();
inSync = false;
return getArray(--i);
}
};
}

/** Returns a copy of this list.
 *
 * @return a copy of this list.
 */
@Override
public ARRAY_FRONT_CODED_LIST clone() {
return this;
}

@Override
public String toString() {
final StringBuffer s = new StringBuffer();
s.append("[");
for(int i = 0; i < n; i++) {
if (i != 0) s.append(", ");
s.append(ARRAY_LIST.wrap(getArray(i)).toString());
}
s.append("]");
return s.toString();
}

/** Computes the pointer array using the currently set ratio, number of elements and underlying array.

```

```

*
* @return the computed pointer array.
*/

protected long[] rebuildPointerArray() {
    final long[] p = new long[(n + ratio - 1) / ratio];
    final KEY_TYPE a[][] = array;
    int length, count;
    long pos = 0;

    for(int i = 0, j = 0, skip = ratio - 1; i < n; i++) {
        length = readInt(a, pos);
        count = count(length);
        if (++skip == ratio) {
            skip = 0;
            p[j++] = pos;
            pos += count + length;
        }
        else pos += count + count(readInt(a, pos + count)) + length;
    }

    return p;
}

private void readObject(java.io.ObjectInputStream s) throws java.io.IOException, ClassNotFoundException {
    s.defaultReadObject();

    // Rebuild pointer array
    p = rebuildPointerArray();
}

#ifdef TEST

private static long seed = System.currentTimeMillis();
private static java.util.Random r = new java.util.Random(seed);

private static KEY_TYPE genKey() {
    #if KEY_CLASS_Byte || KEY_CLASS_Short || KEY_CLASS_Character
        return (KEY_TYPE)r.nextInt();
    #elif KEYS_PRIMITIVE
        return r.NEXT_KEY();
    #elif KEY_CLASS_Object
        return Integer.toBinaryString(r.nextInt());
    #else
        return new java.io.Serializable() {};
    #endif
}

```

```

}

private static java.text.NumberFormat format = new java.text.DecimalFormat("#,###.00");
private static java.text.FieldPosition fp = new java.text.FieldPosition(0);

private static String format(double d) {
    StringBuffer s = new StringBuffer();
    return format.format(d, s, fp).toString();
}

private static void speedTest(int n, boolean comp) {
    System.out.println("There are presently no speed tests for this class.");
}

private static void fatal(String msg) {
    System.out.println(msg);
    System.exit(1);
}

private static void ensure(boolean cond, String msg) {
    if (cond) return;
    fatal(msg);
}

private static boolean contentEquals(java.util.List x, java.util.List y) {
    if (x.size() != y.size()) return false;
    for(int i = 0; i < x.size(); i++) if (! java.util.Arrays.equals((KEY_TYPE[])x.get(i), (KEY_TYPE[])y.get(i))) return
false;
    return true;
}

private static int l[];
private static KEY_TYPE[][] a;

private static void runTest(int n) {
    int c;

    l = new int[n];
    a = new KEY_TYPE[n][];

    for(int i = 0; i < n; i++) l[i] = (int)(Math.abs(r.nextGaussian())*32);
    for(int i = 0; i < n; i++) a[i] = new KEY_TYPE[l[i]];
    for(int i = 0; i < n; i++) for(int j = 0; j < l[i]; j++) a[i][j] = genKey();

    ARRAY_FRONT_CODED_LIST m = new

```

```

ARRAY_FRONT_CODED_LIST(it.unimi.dsi.fastutil.objects.ObjectIterators.wrap(a), r.nextInt(4) + 1);
it.unimi.dsi.fastutil.objects.ObjectArrayList t = new it.unimi.dsi.fastutil.objects.ObjectArrayList(a);

//System.out.println(m);
//for(i = 0; i < t.size(); i++) System.out.println(ARRAY_LIST.wrap((KEY_TYPE[])t.get(i)));

/* Now we check that m actually holds that data. */

ensure(contentEquals(m, t), "Error (" + seed + "): m does not equal t at creation");

/* Now we check cloning. */

ensure(contentEquals(m, (java.util.List)m.clone()), "Error (" + seed + "): m does not equal m.clone()");

/* Now we play with iterators. */

{
    ObjectListIterator i;
    java.util.ListIterator j;
    Object J;
    i = m.listIterator();
    j = t.listIterator();

    for(int k = 0; k < 2*n; k++) {
        ensure(i.hasNext() == j.hasNext(), "Error (" + seed + "): divergence in hasNext()");
        ensure(i.hasPrevious() == j.hasPrevious(), "Error (" + seed + "): divergence in hasPrevious()");

        if (r.nextFloat() < .8 && i.hasNext()) {
            ensure(java.util.Arrays.equals((KEY_TYPE[])i.next(), (KEY_TYPE[])j.next()), "Error (" + seed + "): divergence in
next()");

        }
        else if (r.nextFloat() < .2 && i.hasPrevious()) {
            ensure(java.util.Arrays.equals((KEY_TYPE[])i.previous(), (KEY_TYPE[])j.previous()), "Error (" + seed + "):
divergence in previous()");
        }

        ensure(i.nextIndex() == j.nextIndex(), "Error (" + seed + "): divergence in nextIndex()");
        ensure(i.previousIndex() == j.previousIndex(), "Error (" + seed + "): divergence in previousIndex()");

    }

}

{
    Object previous = null;
    Object I, J;
    int from = r.nextInt(m.size() + 1);

```

```

ObjectListIterator i;
java.util.ListIterator j;
i = m.listIterator(from);
j = t.listIterator(from);

for(int k = 0; k < 2*n; k++) {
    ensure(i.hasNext() == j.hasNext(), "Error (" + seed + "): divergence in hasNext() (iterator with starting point " +
from + ")");
    ensure(i.hasPrevious() == j.hasPrevious() , "Error (" + seed + "): divergence in hasPrevious() (iterator with starting
point " + from + ")");

    if (r.nextFloat() < .8 && i.hasNext()) {
        ensure(java.util.Arrays.equals((KEY_TYPE[])i.next(), (KEY_TYPE[])j.next()), "Error (" + seed + "): divergence in
next() (iterator with starting point " + from + ")");
        //System.err.println("Done next " + I + " " + J + " " + badPrevious);
    }
    else if (r.nextFloat() < .2 && i.hasPrevious()) {
        ensure(java.util.Arrays.equals((KEY_TYPE[])i.previous(), (KEY_TYPE[])j.previous()), "Error (" + seed + "):
divergence in previous() (iterator with starting point " + from + ")");
    }
}

}

}

try {
    java.io.File ff = new java.io.File("it.unimi.dsi.fastutil.test");
    java.io.OutputStream os = new java.io.FileOutputStream(ff);
    java.io.ObjectOutputStream oos = new java.io.ObjectOutputStream(os);

    oos.writeObject(m);
    oos.close();

    java.io.InputStream is = new java.io.FileInputStream(ff);
    java.io.ObjectInputStream ois = new java.io.ObjectInputStream(is);

    m = (ARRAY_FRONT_CODED_LIST)ois.readObject();
    ois.close();
    ff.delete();
}
catch(Exception e) {
    e.printStackTrace();
    System.exit(1);
}

```

```
ensure(contentEquals(m, t), "Error (" + seed + "): m does not equal t after save/read");
```

```
System.out.println("Test OK");  
return;  
}
```

```
public static void main(String args[]) {  
    int n = Integer.parseInt(args[1]);  
    if (args.length > 2) r = new java.util.Random(seed = Long.parseLong(args[2]));  
  
    try {  
        if ("speedTest".equals(args[0]) || "speedComp".equals(args[0])) speedTest(n, "speedComp".equals(args[0]));  
        else if ("test".equals(args[0])) runTest(n);  
    } catch(Throwable e) {  
        e.printStackTrace(System.err);  
        System.err.println("seed: " + seed);  
    }  
}  
  
#endif  
  
}
```

Found in path(s):

```
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-  
a775574/drv/ArrayFrontCodedList.drv
```

No license file was found, but licenses were detected in source scan.

```
/*  
* Copyright (C) 2003-2017 Paolo Boldi and Sebastiano Vigna  
*  
* Licensed under the Apache License, Version 2.0 (the "License");  
* you may not use this file except in compliance with the License.  
* You may obtain a copy of the License at  
*  
* http://www.apache.org/licenses/LICENSE-2.0  
*  
* Unless required by applicable law or agreed to in writing, software  
* distributed under the License is distributed on an "AS IS" BASIS,  
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
* See the License for the specific language governing permissions and  
* limitations under the License.  
*/
```

```
package PACKAGE;
```

```

#if KEY_CLASS_Object
import java.util.Comparator;
#endif

import it.unimi.dsi.fastutil.ints.IntArrays;

/** A class providing static methods and objects that do useful things with semi-indirect heaps.
 *
 * <p>A semi-indirect heap is based on a <em>reference array</em>. Elements of
 * a semi-indirect heap are integers that index the reference array (note that
 * in an <em>indirect</em> heap you can also map elements of the reference
 * array to heap positions).
 */

public final class SEMI_INDIRECT_HEAPS {

    private SEMI_INDIRECT_HEAPS() {}

    /** Moves the given element down into the semi-indirect heap until it reaches the lowest possible position.
     *
     * @param refArray the reference array.
     * @param heap the semi-indirect heap (starting at 0).
     * @param size the number of elements in the heap.
     * @param i the index in the heap of the element to be moved down.
     * @param c a type-specific comparator, or { @code null } for the natural order.
     * @return the new position in the heap of the element of heap index { @code i }.
     */

    SUPPRESS_WARNINGS_KEY_UNCHECKED
    public static KEY_GENERIC int downHeap(final KEY_GENERIC_TYPE[] refArray, final int[] heap, final int size,
    int i, final KEY_COMPARATOR KEY_GENERIC c) {
        assert i < size;

        final int e = heap[i];
        final KEY_GENERIC_TYPE E = refArray[e];
        int child;

        if (c == null)
            while ((child = (i << 1) + 1) < size) {
                int t = heap[child];
                final int right = child + 1;
                if (right < size && KEY_LESS(refArray[heap[right]], refArray[t])) t = heap[child = right];
                if (KEY_LESSEQ(E, refArray[t])) break;
                heap[i] = t;
                i = child;
            }
        else

```

```

while ((child = (i << 1) + 1) < size) {
    int t = heap[child];
    final int right = child + 1;
    if (right < size && c.compare(refArray[heap[right]], refArray[t]) < 0) t = heap[child = right];
    if (c.compare(E, refArray[t]) <= 0) break;
    heap[i] = t;
    i = child;
}

heap[i] = e;

return i;
}

/** Moves the given element up in the semi-indirect heap until it reaches the highest possible position.
 *
 * @param refArray the reference array.
 * @param heap the semi-indirect heap (starting at 0).
 * @param size the number of elements in the heap.
 * @param i the index in the heap of the element to be moved up.
 * @param c a type-specific comparator, or { @code null } for the natural order.
 * @return the new position in the heap of the element of heap index { @code i }.
 */

SUPPRESS_WARNINGS_KEY_UNCHECKED
public static KEY_GENERIC int upHeap(final KEY_GENERIC_TYPE[] refArray, final int[] heap, final int size, int
i, final KEY_COMPARATOR KEY_GENERIC c) {
    assert i < size;

    final int e = heap[i];
    final KEY_GENERIC_TYPE E = refArray[e];

    if (c == null)
        while (i != 0) {
            final int parent = (i - 1) >>> 1;
            final int t = heap[parent];
            if (KEY_LESSEQ(refArray[t], E)) break;
            heap[i] = t;
            i = parent;
        }
    else
        while (i != 0) {
            final int parent = (i - 1) >>> 1;
            final int t = heap[parent];
            if (c.compare(refArray[t], E) <= 0) break;
            heap[i] = t;
            i = parent;
        }
}

```

```

heap[i] = e;

return i;
}

/** Creates a semi-indirect heap in the given array.
 *
 * @param refArray the reference array.
 * @param offset the first element of the reference array to be put in the heap.
 * @param length the number of elements to be put in the heap.
 * @param heap the array where the heap is to be created.
 * @param c a type-specific comparator, or { @code null } for the natural order.
 */

public static KEY_GENERIC void makeHeap(final KEY_GENERIC_TYPE[] refArray, final int offset, final int
length, final int[] heap, final KEY_COMPARATOR KEY_GENERIC c) {
    ARRAYS.ensureOffsetLength(refArray, offset, length);
    if (heap.length < length) throw new IllegalArgumentException("The heap length (" + heap.length + ") is smaller
than the number of elements (" + length + ")");

    int i = length;
    while(i-- != 0) heap[i] = offset + i;

    i = length >>> 1;
    while(i-- != 0) downHeap(refArray, heap, length, i, c);
}

/** Creates a semi-indirect heap, allocating its heap array.
 *
 * @param refArray the reference array.
 * @param offset the first element of the reference array to be put in the heap.
 * @param length the number of elements to be put in the heap.
 * @param c a type-specific comparator, or { @code null } for the natural order.
 * @return the heap array.
 */

public static KEY_GENERIC int[] makeHeap(final KEY_GENERIC_TYPE[] refArray, final int offset, final int
length, final KEY_COMPARATOR KEY_GENERIC c) {
    final int[] heap = length <= 0 ? IntArrays.EMPTY_ARRAY : new int[length];
    makeHeap(refArray, offset, length, heap, c);
    return heap;
}

/** Creates a semi-indirect heap from a given index array.
 *

```

```

* @param refArray the reference array.
* @param heap an array containing indices into { @code refArray }.
* @param size the number of elements in the heap.
* @param c a type-specific comparator, or { @code null } for the natural order.
*/

public static KEY_GENERIC void makeHeap(final KEY_GENERIC_TYPE[] refArray, final int[] heap, final int
size, final KEY_COMPARATOR KEY_GENERIC c) {
    int i = size >>> 1;
    while(i-- != 0) downHeap(refArray, heap, size, i, c);
}

/** Retrieves the front of a heap in a given array.
 *
 * <p>The <em>front</em> of a semi-indirect heap is the set of indices whose associated elements in the reference
array
 * are equal to the element associated to the first index.
 *
 * <p>In several circumstances you need to know the front, and scanning linearly the entire heap is not
 * the best strategy. This method simulates (using a partial linear scan) a breadth-first visit that
 * terminates when all visited nodes are larger than the element associated
 * to the top index, which implies that no elements of the front can be found later.
 * In most cases this trick yields a significant improvement.
 *
 * @param refArray the reference array.
 * @param heap an array containing indices into { @code refArray }.
 * @param size the number of elements in the heap.
 * @param a an array large enough to hold the front (e.g., at least long as { @code refArray }).
 * @return the number of elements actually written (starting from the first position of { @code a }).
 */
SUPPRESS_WARNINGS_KEY_UNCHECKED
public static KEY_GENERIC int front(final KEY_GENERIC_TYPE[] refArray, final int[] heap, final int size, final
int[] a) {
    final KEY_GENERIC_TYPE top = refArray[heap[0]];
    int j = 0, // The current position in a
        l = 0, // The first position to visit in the next level (inclusive)
        r = 1, // The last position to visit in the next level (exclusive)
        f = 0; // The first position (in the heap array) of the next level
    for(int i = 0; i < r; i++) {
        if (i == f) { // New level
            if (l >= r) break; // If we are crossing the two bounds, we're over
            f = (f << 1) + 1; // Update the first position of the next level...
            i = l; // ...and jump directly to position l
            l = -1; // Invalidate l
        }
        if (KEY_CMP_EQ(top, refArray[heap[i]])) {
            a[j++] = heap[i];
            if (l == -1) l = i * 2 + 1; // If this is the first time in this level, set l
        }
    }
}

```

```

    r = Math.min(size, i * 2 + 3); // Update r, but do not go beyond size
}
}

return j;
}

/** Retrieves the front of a heap in a given array using a given comparator.
 *
 * <p>The <em>front</em> of a semi-indirect heap is the set of indices whose associated elements in the reference
array
 * are equal to the element associated to the first index.
 *
 * <p>In several circumstances you need to know the front, and scanning linearly the entire heap is not
 * the best strategy. This method simulates (using a partial linear scan) a breadth-first visit that
 * terminates when all visited nodes are larger than the element associated
 * to the top index, which implies that no elements of the front can be found later.
 * In most cases this trick yields a significant improvement.
 *
 * @param refArray the reference array.
 * @param heap an array containing indices into { @code refArray }.
 * @param size the number of elements in the heap.
 * @param a an array large enough to hold the front (e.g., at least long as { @code refArray }).
 * @param c a type-specific comparator.
 * @return the number of elements actually written (starting from the first position of { @code a }).
 */
public static KEY_GENERIC int front(final KEY_GENERIC_TYPE[] refArray, final int[] heap, final int size, final
int[] a, final KEY_COMPARATOR KEY_GENERIC c) {
    final KEY_GENERIC_TYPE top = refArray[heap[0]];
    int j = 0, // The current position in a
        l = 0, // The first position to visit in the next level (inclusive)
        r = 1, // The last position to visit in the next level (exclusive)
        f = 0; // The first position (in the heap array) of the next level
    for(int i = 0; i < r; i++) {
        if (i == f) { // New level
            if (l >= r) break; // If we are crossing the two bounds, we're over
            f = (f << 1) + 1; // Update the first position of the next level...
            i = l; // ...and jump directly to position l
            l = -1; // Invalidate l
        }
        if (c.compare(top, refArray[heap[i]]) == 0) {
            a[j++] = heap[i];
            if (l == -1) l = i * 2 + 1; // If this is the first time in this level, set l
            r = Math.min(size, i * 2 + 3); // Update r, but do not go beyond size
        }
    }

    return j;
}

```

```
}  
}
```

Found in path(s):

```
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/drv/SemiIndirectHeaps.drv
```

No license file was found, but licenses were detected in source scan.

```
/*
```

```
* Copyright (C) 2010-2017 Sebastiano Vigna
```

```
*
```

```
* Licensed under the Apache License, Version 2.0 (the "License");
```

```
* you may not use this file except in compliance with the License.
```

```
* You may obtain a copy of the License at
```

```
*
```

```
* http://www.apache.org/licenses/LICENSE-2.0
```

```
*
```

```
* Unless required by applicable law or agreed to in writing, software
```

```
* distributed under the License is distributed on an "AS IS" BASIS,
```

```
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
```

```
* See the License for the specific language governing permissions and
```

```
* limitations under the License.
```

```
*/
```

Found in path(s):

```
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/src/it/unimi/dsi/fastutil/Swapper.java
```

```
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/src/it/unimi/dsi/fastutil/BigList.java
```

```
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/src/it/unimi/dsi/fastutil/Size64.java
```

```
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/src/it/unimi/dsi/fastutil/BigListIterator.java
```

```
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/src/it/unimi/dsi/fastutil/BigSwapper.java
```

No license file was found, but licenses were detected in source scan.

```
/*
```

```
* Copyright (C) 2002-2017 Sebastiano Vigna
```

```
*
```

```
* Licensed under the Apache License, Version 2.0 (the "License");
```

```
* you may not use this file except in compliance with the License.
```

```
* You may obtain a copy of the License at
```

```
*
```

```
* http://www.apache.org/licenses/LICENSE-2.0
```

```
*
```

```
* Unless required by applicable law or agreed to in writing, software
```

```
* distributed under the License is distributed on an "AS IS" BASIS,
```

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

```
package PACKAGE;
```

```
import java.util.List;  
import java.util.Collection;  
import java.util.Random;  
import java.util.RandomAccess;
```

```
/** A class providing static methods and objects that do useful things with type-specific lists.  
 *  
 * @see java.util.Collections  
 */
```

```
public final class LISTS {
```

```
    private LISTS() {}
```

```
    /** Shuffles the specified list using the specified pseudorandom number generator.  
     *  
     * @param l the list to be shuffled.  
     * @param random a pseudorandom number generator.  
     * @return { @code l }.  
     */
```

```
    public static KEY_GENERIC LIST KEY_GENERIC shuffle(final LIST KEY_GENERIC l, final Random random)  
    {  
        for(int i = l.size(); i-- != 0;) {  
            final int p = random.nextInt(i + 1);  
            final KEY_GENERIC_TYPE t = l.GET_KEY(i);  
            l.set(i, l.GET_KEY(p));  
            l.set(p, t);  
        }  
        return l;  
    }  
}
```

```
    /** An immutable class representing an empty type-specific list.  
     *  
     * <p>This class may be useful to implement your own in case you subclass  
     * a type-specific list.  
     */
```

```
    public static class EmptyList KEY_GENERIC extends COLLECTIONS.EmptyCollection KEY_GENERIC  
    implements LIST KEY_GENERIC, RandomAccess, java.io.Serializable, Cloneable {
```

```

private static final long serialVersionUID = -7046029254386353129L;

protected EmptyList() {}

@Override
public KEY_GENERIC_TYPE GET_KEY(int i) { throw new IndexOutOfBoundsException(); }

@Override
public boolean REMOVE(KEY_TYPE k) { throw new UnsupportedOperationException(); }

@Override
public KEY_GENERIC_TYPE REMOVE_KEY(int i) { throw new UnsupportedOperationException(); }

@Override
public void add(final int index, final KEY_GENERIC_TYPE k) { throw new UnsupportedOperationException(); }

@Override
public KEY_GENERIC_TYPE set(final int index, final KEY_GENERIC_TYPE k) { throw new
UnsupportedOperationException(); }

@Override
public int indexOf(KEY_TYPE k) { return -1; }

@Override
public int lastIndexOf(KEY_TYPE k) { return -1; }

@Override
public boolean addAll(int i, Collection<? extends KEY_GENERIC_CLASS> c) { throw new
UnsupportedOperationException(); }

#if KEYS_PRIMITIVE
@Override
public boolean addAll(LIST c) { throw new UnsupportedOperationException(); }

@Override
public boolean addAll(int i, COLLECTION c) { throw new UnsupportedOperationException(); }

@Override
public boolean addAll(int i, LIST c) { throw new UnsupportedOperationException(); }

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@SuppressWarnings("deprecation")
@Deprecated
@Override
public void add(final int index, final KEY_GENERIC_CLASS k) { throw new UnsupportedOperationException(); }

/** {@inheritDoc}

```

```

    * @deprecated Please use the corresponding type-specific method instead. */
    @SuppressWarnings("deprecation")
    @Deprecated
    @Override
    public KEY_CLASS get(final int index) { throw new UnsupportedOperationException(); }

    /** { @inheritDoc }
    * @deprecated Please use the corresponding type-specific method instead. */
    @SuppressWarnings("deprecation")
    @Deprecated
    @Override
    public boolean add(final KEY_GENERIC_CLASS k) { throw new UnsupportedOperationException(); }

    /** { @inheritDoc }
    * @deprecated Please use the corresponding type-specific method instead. */
    @SuppressWarnings("deprecation")
    @Deprecated
    @Override
    public KEY_GENERIC_CLASS set(final int index, final KEY_GENERIC_CLASS k) { throw new
    UnsupportedOperationException(); }

    /** { @inheritDoc }
    * @deprecated Please use the corresponding type-specific method instead. */
    @SuppressWarnings("deprecation")
    @Deprecated
    @Override
    public KEY_GENERIC_CLASS remove(int k) { throw new UnsupportedOperationException(); }

    /** { @inheritDoc }
    * @deprecated Please use the corresponding type-specific method instead. */
    @SuppressWarnings("deprecation")
    @Deprecated
    @Override
    public int indexOf(Object k) { return -1; }

    /** { @inheritDoc }
    * @deprecated Please use the corresponding type-specific method instead. */
    @SuppressWarnings("deprecation")
    @Deprecated
    @Override
    public int lastIndexOf(Object k) { return -1; }
#endif
    SUPPRESS_WARNINGS_KEY_UNCHECKED
    @Override
    public KEY_LIST_ITERATOR KEY_GENERIC listIterator() { return ITERATORS.EMPTY_ITERATOR; }

    SUPPRESS_WARNINGS_KEY_UNCHECKED
    @Override

```

```

public KEY_LIST_ITERATOR KEY_GENERIC iterator() { return ITERATORS.EMPTY_ITERATOR; }

SUPPRESS_WARNINGS_KEY_UNCHECKED
@Override
public KEY_LIST_ITERATOR KEY_GENERIC listIterator(int i) { if (i == 0) return
ITERATORS.EMPTY_ITERATOR; throw new IndexOutOfBoundsException(String.valueOf(i)); }

@Override
public LIST KEY_GENERIC subList(int from, int to) { if (from == 0 && to == 0) return this; throw new
IndexOutOfBoundsException(); }

@Override
public void getElements(int from, KEY_TYPE[] a, int offset, int length) { if (from == 0 && length == 0 && offset
>= 0 && offset <= a.length) return; throw new IndexOutOfBoundsException(); }

@Override
public void removeElements(int from, int to) { throw new UnsupportedOperationException(); }

@Override
public void addElements(int index, final KEY_GENERIC_TYPE a[], int offset, int length) { throw new
UnsupportedOperationException(); }

@Override
public void addElements(int index, final KEY_GENERIC_TYPE a[]) { throw new
UnsupportedOperationException(); }

@Override
public void size(int s) { throw new UnsupportedOperationException(); }

#if ! KEY_CLASS_Reference
@Override
public int compareTo(final List<? extends KEY_GENERIC_CLASS> o) {
    if (o == this) return 0;
    return ((List<?>)o).isEmpty() ? 0 : -1;
}
#endif

@Override
public Object clone() { return EMPTY_LIST; }

@Override
public int hashCode() { return 1; }

@Override
@Override
@SuppressWarnings("rawtypes")
public boolean equals(Object o) { return o instanceof List && ((List)o).isEmpty(); }

@Override

```

```

public String toString() { return "[]"; }

private Object readResolve() { return EMPTY_LIST; }
}

/** An empty list (immutable). It is serializable and cloneable.
 */
SUPPRESS_WARNINGS_KEY_RAWTYPES
public static final EmptyList EMPTY_LIST = new EmptyList();

#if KEYS_REFERENCE
/** Returns an empty list (immutable). It is serializable and cloneable.
 *
 * <p>This method provides a typesafe access to {@link #EMPTY_LIST}.
 * @return an empty list (immutable).
 */
@SuppressWarnings("unchecked")
public static KEY_GENERIC LIST KEY_GENERIC emptyList() {
    return EMPTY_LIST;
}
#endif

/** An immutable class representing a type-specific singleton list.
 *
 * <p>This class may be useful to implement your own in case you subclass
 * a type-specific list.
 */

public static class Singleton KEY_GENERIC extends ABSTRACT_LIST KEY_GENERIC implements
RandomAccess, java.io.Serializable, Cloneable {

    private static final long serialVersionUID = -7046029254386353129L;

    private final KEY_GENERIC_TYPE element;

    protected Singleton(final KEY_GENERIC_TYPE element) { this.element = element; }

    @Override
    public KEY_GENERIC_TYPE GET_KEY(final int i) { if (i == 0) return element; throw new
IndexOutOfBoundsException(); }

    @Override
    public boolean REMOVE(KEY_TYPE k) { throw new UnsupportedOperationException(); }

    @Override
    public KEY_GENERIC_TYPE REMOVE_KEY(final int i) { throw new UnsupportedOperationException(); }
}

```

```

@Override
public boolean contains(final KEY_TYPE k) { return KEY_EQUALS(k, element); }

/* Slightly optimized w.r.t. the one in ABSTRACT_SET. */

@Override
public KEY_TYPE[] TO_KEY_ARRAY() {
    KEY_TYPE a[] = new KEY_TYPE[1];
    a[0] = element;
    return a;
}

@Override
public KEY_LIST_ITERATOR KEY_GENERIC listIterator() { return ITERATORS.singleton(element); }

@Override
public KEY_LIST_ITERATOR KEY_GENERIC iterator() { return listIterator(); }

@Override
public KEY_LIST_ITERATOR KEY_GENERIC listIterator(final int i) {
    if (i > 1 || i < 0) throw new IndexOutOfBoundsException();
    final KEY_LIST_ITERATOR KEY_GENERIC l = listIterator();
    if (i == 1) l.NEXT_KEY();
    return l;
}

@Override
SUPPRESS_WARNINGS_KEY_UNCHECKED
public LIST KEY_GENERIC subList(final int from, final int to) {
    ensureIndex(from);
    ensureIndex(to);
    if (from > to) throw new IndexOutOfBoundsException("Start index (" + from + ") is greater than end index (" + to +
    ")");

    if (from != 0 || to != 1) return EMPTY_LIST;
    return this;
}

@Override
public boolean addAll(int i, Collection<? extends KEY_GENERIC_CLASS> c) { throw new
UnsupportedOperationException(); }

@Override
public boolean addAll(final Collection<? extends KEY_GENERIC_CLASS> c) { throw new
UnsupportedOperationException(); }

@Override
public boolean removeAll(final Collection<?> c) { throw new UnsupportedOperationException(); }

```

```

@Override
public boolean retainAll(final Collection<?> c) { throw new UnsupportedOperationException(); }

#if KEYS_PRIMITIVE

@Override
public boolean addAll(LIST c) { throw new UnsupportedOperationException(); }

@Override
public boolean addAll(int i, LIST c) { throw new UnsupportedOperationException(); }

@Override
public boolean addAll(int i, COLLECTION c) { throw new UnsupportedOperationException(); }

@Override
public boolean addAll(final COLLECTION c) { throw new UnsupportedOperationException(); }

@Override
public boolean removeAll(final COLLECTION c) { throw new UnsupportedOperationException(); }

@Override
public boolean retainAll(final COLLECTION c) { throw new UnsupportedOperationException(); }

#endif

@Override
public int size() { return 1; }

@Override
public void size(final int size) { throw new UnsupportedOperationException(); }

@Override
public void clear() { throw new UnsupportedOperationException(); }

@Override
public Object clone() { return this; }
}

/** Returns a type-specific immutable list containing only the specified element. The returned list is serializable and
cloneable.
*
* @param element the only element of the returned list.
* @return a type-specific immutable list containing just { @code element}.
*/

public static KEY_GENERIC LIST KEY_GENERIC singleton(final KEY_GENERIC_TYPE element) { return new
Singleton KEY_GENERIC_DIAMOND(element); }

```

```
#if KEYS_PRIMITIVE
```

```
/** Returns a type-specific immutable list containing only the specified element. The returned list is serializable and cloneable.
```

```
*
```

```
* @param element the only element of the returned list.
```

```
* @return a type-specific immutable list containing just { @code element}.
```

```
*/
```

```
public static KEY_GENERIC LIST KEY_GENERIC singleton(final Object element) { return new Singleton  
KEY_GENERIC_DIAMOND(KEY_OBJ2TYPE(element)); }
```

```
#endif
```

```
/** A synchronized wrapper class for lists. */
```

```
public static class SynchronizedList KEY_GENERIC extends COLLECTIONS.SynchronizedCollection  
KEY_GENERIC implements LIST KEY_GENERIC, java.io.Serializable {
```

```
private static final long serialVersionUID = -7046029254386353129L;
```

```
protected final LIST KEY_GENERIC list; // Due to the large number of methods that are not in COLLECTION,  
this is worth caching.
```

```
protected SynchronizedList(final LIST KEY_GENERIC l, final Object sync) {  
super(l, sync);  
this.list = l;  
}
```

```
protected SynchronizedList(final LIST KEY_GENERIC l) {  
super(l);  
this.list = l;  
}
```

```
@Override
```

```
public KEY_GENERIC_TYPE GET_KEY(final int i) { synchronized(sync) { return list.GET_KEY(i); } }
```

```
@Override
```

```
public KEY_GENERIC_TYPE set(final int i, final KEY_GENERIC_TYPE k) { synchronized(sync) { return  
list.set(i, k); } }
```

```
@Override
```

```
public void add(final int i, final KEY_GENERIC_TYPE k) { synchronized(sync) { list.add(i, k); } }
```

```
@Override
```

```
public KEY_GENERIC_TYPE REMOVE_KEY(final int i) { synchronized(sync) { return list.REMOVE_KEY(i); }  
}
```

```

@Override
public int indexOf(final KEY_TYPE k) { synchronized(sync) { return list.indexOf(k); } }

@Override
public int lastIndexOf(final KEY_TYPE k) { synchronized(sync) { return list.lastIndexOf(k); } }

@Override
public boolean addAll(final int index, final Collection<? extends KEY_GENERIC_CLASS> c) {
synchronized(sync) { return list.addAll(index, c); } }

@Override
public void getElements(final int from, final KEY_TYPE a[], final int offset, final int length) { synchronized(sync)
{ list.getElements(from, a, offset, length); } }

@Override
public void removeElements(final int from, final int to) { synchronized(sync) { list.removeElements(from, to); } }

@Override
public void addElements(int index, final KEY_GENERIC_TYPE a[], int offset, int length) { synchronized(sync) {
list.addElements(index, a, offset, length); } }

@Override
public void addElements(int index, final KEY_GENERIC_TYPE a[]) { synchronized(sync) {
list.addElements(index, a); } }

@Override
public void size(final int size) { synchronized(sync) { list.size(size); } }

@Override
public KEY_LIST_ITERATOR KEY_GENERIC listIterator() { return list.listIterator(); }

@Override
public KEY_LIST_ITERATOR KEY_GENERIC iterator() { return listIterator(); }

@Override
public KEY_LIST_ITERATOR KEY_GENERIC listIterator(final int i) { return list.listIterator(i); }

@Override
public LIST KEY_GENERIC subList(final int from, final int to) { synchronized(sync) { return new
SynchronizedList KEY_GENERIC_DIAMOND(list.subList(from, to), sync); } }

@Override
public boolean equals(final Object o) { if (o == this) return true; synchronized(sync) { return collection.equals(o); }
}

@Override
public int hashCode() { synchronized(sync) { return collection.hashCode(); } }

```

```

#if ! KEY_CLASS_Reference
    @Override
    public int compareTo(final List<? extends KEY_GENERIC_CLASS> o) { synchronized(sync) { return
list.compareTo(o); } }
#endif

#if KEYS_PRIMITIVE
    @Override
    public boolean addAll(final int index, final COLLECTION c) { synchronized(sync) { return list.addAll(index, c); }
}

    @Override
    public boolean addAll(final int index, LIST l) { synchronized(sync) { return list.addAll(index, l); } }

    @Override
    public boolean addAll(LIST l) { synchronized(sync) { return list.addAll(l); } }

    /** {@inheritDoc}
    * @deprecated Please use the corresponding type-specific method instead. */
    @Deprecated
    @Override
    public KEY_GENERIC_CLASS get(final int i) { synchronized(sync) { return list.get(i); } }

    /** {@inheritDoc}
    * @deprecated Please use the corresponding type-specific method instead. */
    @Deprecated
    @Override
    public void add(final int i, KEY_GENERIC_CLASS k) { synchronized(sync) { list.add(i, k); } }

    /** {@inheritDoc}
    * @deprecated Please use the corresponding type-specific method instead. */
    @Deprecated
    @Override
    public KEY_GENERIC_CLASS set(final int index, KEY_GENERIC_CLASS k) { synchronized(sync) { return
list.set(index, k); } }

    /** {@inheritDoc}
    * @deprecated Please use the corresponding type-specific method instead. */
    @Deprecated
    @Override
    public KEY_GENERIC_CLASS remove(final int i) { synchronized(sync) { return list.remove(i); } }

    /** {@inheritDoc}
    * @deprecated Please use the corresponding type-specific method instead. */
    @Deprecated
    @Override
    public int indexOf(final Object o) { synchronized(sync) { return list.indexOf(o); } }

```

```

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
public int lastIndexOf(final Object o) { synchronized(sync) { return list.lastIndexOf(o); } }
#endif

private void writeObject(java.io.ObjectOutputStream s) throws java.io.IOException {
    synchronized(sync) { s.defaultWriteObject(); }
}

}

/** A synchronized wrapper class for random-access lists. */

public static class SynchronizedRandomAccessList KEY_GENERIC extends SynchronizedList KEY_GENERIC
implements RandomAccess, java.io.Serializable {
    private static final long serialVersionUID = 0L;

    protected SynchronizedRandomAccessList(final LIST KEY_GENERIC l, final Object sync) {
        super(l, sync);
    }

    protected SynchronizedRandomAccessList(final LIST KEY_GENERIC l) {
        super(l);
    }

    @Override
    public LIST KEY_GENERIC subList(final int from, final int to) { synchronized(sync) { return new
SynchronizedRandomAccessList KEY_GENERIC_DIAMOND(list.subList(from, to), sync); } }
}

/** Returns a synchronized type-specific list backed by the given type-specific list.
 *
 * @param l the list to be wrapped in a synchronized list.
 * @return a synchronized view of the specified list.
 * @see java.util.Collections#synchronizedList(List)
 */
public static KEY_GENERIC LIST KEY_GENERIC synchronize(final LIST KEY_GENERIC l) {
    return l instanceof RandomAccess ? new SynchronizedRandomAccessList KEY_GENERIC_DIAMOND(l) : new
SynchronizedList KEY_GENERIC_DIAMOND(l);
}

/** Returns a synchronized type-specific list backed by the given type-specific list, using an assigned object to
synchronize.
 *

```

```

* @param l the list to be wrapped in a synchronized list.
* @param sync an object that will be used to synchronize the access to the list.
* @return a synchronized view of the specified list.
* @see java.util.Collections#synchronizedList(List)
*/

public static KEY_GENERIC LIST KEY_GENERIC synchronize(final LIST KEY_GENERIC l, final Object sync)
{
    return l instanceof RandomAccess ? new SynchronizedRandomAccessList KEY_GENERIC_DIAMOND(l, sync) :
    new SynchronizedList KEY_GENERIC_DIAMOND(l, sync);
}

/** An unmodifiable wrapper class for lists. */

public static class UnmodifiableList KEY_GENERIC extends COLLECTIONS.UnmodifiableCollection
KEY_GENERIC implements LIST KEY_GENERIC, java.io.Serializable {

    private static final long serialVersionUID = -7046029254386353129L;

    protected final LIST KEY_GENERIC list; // Due to the large number of methods that are not in COLLECTION,
this is worth caching.

    protected UnmodifiableList(final LIST KEY_GENERIC l) {
        super(l);
        this.list = l;
    }

    @Override
    public KEY_GENERIC_TYPE GET_KEY(final int i) { return list.GET_KEY(i); }

    @Override
    public KEY_GENERIC_TYPE set(final int i, final KEY_GENERIC_TYPE k) { throw new
UnsupportedOperationException(); }

    @Override
    public void add(final int i, final KEY_GENERIC_TYPE k) { throw new UnsupportedOperationException(); }

    @Override
    public KEY_GENERIC_TYPE REMOVE_KEY(final int i) { throw new UnsupportedOperationException(); }

    @Override
    public int indexOf(final KEY_TYPE k) { return list.indexOf(k); }

    @Override
    public int lastIndexOf(final KEY_TYPE k) { return list.lastIndexOf(k); }

```

```

@Override
public boolean addAll(final int index, final Collection<? extends KEY_GENERIC_CLASS> c) { throw new
UnsupportedOperationException(); }

@Override
public void getElements(final int from, final KEY_TYPE a[], final int offset, final int length) {
list.getElements(from, a, offset, length); }

@Override
public void removeElements(final int from, final int to) { throw new UnsupportedOperationException(); }

@Override
public void addElements(int index, final KEY_GENERIC_TYPE a[], int offset, int length) { throw new
UnsupportedOperationException(); }

@Override
public void addElements(int index, final KEY_GENERIC_TYPE a[]) { throw new
UnsupportedOperationException(); }

@Override
public void size(final int size) { list.size(size); }

@Override
public KEY_LIST_ITERATOR KEY_GENERIC listIterator() { return
ITERATORS.unmodifiable(list.listIterator()); }

@Override
public KEY_LIST_ITERATOR KEY_GENERIC iterator() { return listIterator(); }

@Override
public KEY_LIST_ITERATOR KEY_GENERIC listIterator(final int i) { return
ITERATORS.unmodifiable(list.listIterator(i)); }

@Override
public LIST KEY_GENERIC subList(final int from, final int to) { return new UnmodifiableList
KEY_GENERIC_DIAMOND(list.subList(from, to)); }

@Override
public boolean equals(final Object o) { if (o == this) return true; return collection.equals(o); }

@Override
public int hashCode() { return collection.hashCode(); }

#if ! KEY_CLASS_Reference
@Override
public int compareTo(final List<? extends KEY_GENERIC_CLASS> o) { return list.compareTo(o); }
#endif

```

```

#if KEYS_PRIMITIVE
    @Override
    public boolean addAll(final int index, final COLLECTION c) { throw new UnsupportedOperationException(); }

    @Override
    public boolean addAll(final LIST l) { throw new UnsupportedOperationException(); }

    @Override
    public boolean addAll(final int index, final LIST l) { throw new UnsupportedOperationException(); }

    /** {@inheritDoc}
     * @deprecated Please use the corresponding type-specific method instead. */
    @Deprecated
    @Override
    public KEY_GENERIC_CLASS get(final int i) { return list.get(i); }

    /** {@inheritDoc}
     * @deprecated Please use the corresponding type-specific method instead. */
    @Deprecated
    @Override
    public void add(final int i, KEY_GENERIC_CLASS k) { throw new UnsupportedOperationException(); }

    /** {@inheritDoc}
     * @deprecated Please use the corresponding type-specific method instead. */
    @Deprecated
    @Override
    public KEY_GENERIC_CLASS set(final int index, KEY_GENERIC_CLASS k) { throw new
UnsupportedOperationException(); }

    /** {@inheritDoc}
     * @deprecated Please use the corresponding type-specific method instead. */
    @Deprecated
    @Override
    public KEY_GENERIC_CLASS remove(final int i) { throw new UnsupportedOperationException(); }

    /** {@inheritDoc}
     * @deprecated Please use the corresponding type-specific method instead. */
    @Deprecated
    @Override
    public int indexOf(final Object o) { return list.indexOf(o); }

    /** {@inheritDoc}
     * @deprecated Please use the corresponding type-specific method instead. */
    @Deprecated
    @Override
    public int lastIndexOf(final Object o) { return list.lastIndexOf(o); }
#endif
}

```

```

/** An unmodifiable wrapper class for random-access lists. */

public static class UnmodifiableRandomAccessList KEY_GENERIC extends UnmodifiableList KEY_GENERIC
implements RandomAccess, java.io.Serializable {

    private static final long serialVersionUID = 0L;

    protected UnmodifiableRandomAccessList(final LIST KEY_GENERIC l) {
        super(l);
    }

    @Override
    public LIST KEY_GENERIC subList(final int from, final int to) { return new UnmodifiableRandomAccessList
KEY_GENERIC_DIAMOND(list.subList(from, to)); }
}

/** Returns an unmodifiable type-specific list backed by the given type-specific list.
 *
 * @param l the list to be wrapped in an unmodifiable list.
 * @return an unmodifiable view of the specified list.
 * @see java.util.Collections#unmodifiableList(List)
 */
public static KEY_GENERIC LIST KEY_GENERIC unmodifiable(final LIST KEY_GENERIC l) {
    return l instanceof RandomAccess ? new UnmodifiableRandomAccessList KEY_GENERIC_DIAMOND(l) : new
UnmodifiableList KEY_GENERIC_DIAMOND(l);
}

#ifdef TEST

    private static KEY_TYPE genKey() {
    #if KEY_CLASS_Byte || KEY_CLASS_Short || KEY_CLASS_Character
        return (KEY_TYPE)(r.nextInt());
    #elif KEYS_PRIMITIVE
        return r.NEXT_KEY();
    #elif KEY_CLASS_Object
        return Integer.toBinaryString(r.nextInt());
    #else
        return new java.io.Serializable() {};
    #endif
    }
}

```

```

private static void testLists(KEY_TYPE k, LIST m, List t, int level) {
    int n = 100;
    int c;

    long ms;
    boolean mThrowsIllegal, tThrowsIllegal, mThrowsNoElement, tThrowsNoElement, mThrowsIndex, tThrowsIndex,
    mThrowsUnsupp, tThrowsUnsupp;
    boolean rt = false, rm = false;
    Object Rt = null, Rm = null;

    if (level == 0) return;

    /* Now we check that m and t are equal. */
    if (!m.equals(t) || !t.equals(m)) System.err.println("m: " + m + " t: " + t);

    ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) at start");
    ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) at start");

    /* Now we check that m actually holds that data. */
    for(java.util.Iterator i=t.iterator(); i.hasNext();) {
        ensure(m.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after insertion (iterating on
t)");
    }

    /* Now we check that m actually holds that data, but iterating on m. */
    for(java.util.Iterator i=m.listIterator(); i.hasNext();) {
        ensure(t.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after insertion (iterating on
m)");
    }

    /* Now we check that inquiries about random data give the same answer in m and t. For
m we use the polymorphic method. */

    for(int i=0; i<n; i++) {
        KEY_TYPE T = genKey();

        mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;

        try {
            m.contains(T);
        }
        catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
        catch (IllegalArgumentException e) { mThrowsIllegal = true; }
        catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }

        try {

```

```

    t.contains(KEY2OBJ(T));
}
catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
catch (IllegalArgumentException e) { tThrowsIllegal = true; }
catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }

    ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): contains() divergence in
java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ") " + m);
    ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): contains() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
    ensure(mThrowsIndex == tThrowsIndex, "Error (" + level + ", " + seed + "): contains() divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
    if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex) ensure(m.contains(KEY2OBJ(T)) ==
t.contains(KEY2OBJ(T)), "Error (" + level + ", " + seed + "): divergence in keys between t and m (polymorphic
method) " + m);
}

/* Again, we check that inquiries about random data give the same answer in m and t, but
for m we use the standard method. */

for(int i=0; i<n; i++) {
    KEY_TYPE T = genKey();

    mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;

    try {
        m.contains(KEY2OBJ(T));
    }
    catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
    catch (IllegalArgumentException e) { mThrowsIllegal = true; }
    catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
    catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

    try {
        t.contains(KEY2OBJ(T));
    }
    catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
    catch (IllegalArgumentException e) { tThrowsIllegal = true; }
    catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
    catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }

    ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): contains() divergence in
java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ") " + m);
    ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): contains() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
    ensure(mThrowsIndex == tThrowsIndex, "Error (" + level + ", " + seed + "): contains() divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);

```

```

    ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + level + ", " + seed + "): contains() divergence in
UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);
    if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex && !mThrowsUnsupp)
ensure(m.contains(KEY2OBJ(T)) == t.contains(KEY2OBJ(T)), "Error (" + level + ", " + seed + "): divergence
between t and m (standard method) " + m);
}

/* Now we add and remove random data in m and t, checking that the result is the same. */

for(int i=0; i<20*n; i++) {
    KEY_TYPE T = genKey();

    mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;

    try {
        rm = m.add(KEY2OBJ(T));
    }
    catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
    catch (IllegalArgumentException e) { mThrowsIllegal = true; }
    catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
    catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

    try {
        rt = t.add(KEY2OBJ(T));
    }
    catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
    catch (IllegalArgumentException e) { tThrowsIllegal = true; }
    catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
    catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }

    ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): add() divergence in
java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ") " + m);
    ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): add() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
    ensure(mThrowsIndex == tThrowsIndex, "Error (" + level + ", " + seed + "): add() divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
    ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + level + ", " + seed + "): add() divergence in
UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);
    if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex && !mThrowsUnsupp) ensure(rm == rt, "Error
(" + level + ", " + seed + "): divergence in add() between t and m " + m);

    T = genKey();

    mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;

    try {

```

```

    rm = m.remove(KEY2OBJ(T));
}
catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
catch (IllegalArgumentException e) { mThrowsIllegal = true; }
catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

try {
    rt = t.remove(KEY2OBJ(T));
}
catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
catch (IllegalArgumentException e) { tThrowsIllegal = true; }
catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }

if (!KEY_EQUALS(T, k) && mThrowsUnsupp && !tThrowsUnsupp) mThrowsUnsupp = true; // Stupid bug in
Collections.singleton()

    ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): remove() divergence in
java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ") " + m);
    ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): remove() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
    ensure(mThrowsIndex == tThrowsIndex, "Error (" + level + ", " + seed + "): remove() divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
    ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + level + ", " + seed + "): remove() divergence in
UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);
    if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex && !mThrowsUnsupp) ensure(rm == rt, "Error
(" + level + ", " + seed + "): divergence in remove() between t and m " + m);
}

ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after removal " + m);
ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after removal " + m);

/* Now we add and remove random data in m and t at specific positions, checking that the result is the same. */

for(int i=0; i<20*n; i++) {
    KEY_TYPE T = genKey();

    mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;

    int pos = r.nextInt(2);

    try {
        m.add(pos, KEY2OBJ(T));
    }
    catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
    catch (IllegalArgumentException e) { mThrowsIllegal = true; }

```

```

catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

try {
    t.add(pos, KEY2OBJ(T));
}
catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
catch (IllegalArgumentException e) { tThrowsIllegal = true; }
catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }

ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): add() at " + pos + "
divergence in java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement
+ ") " + m);
ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): add() at " + pos + " divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
ensure(mThrowsIndex == tThrowsIndex, "Error (" + level + ", " + seed + "): add() at " + pos + " divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + level + ", " + seed + "): add() at " + pos + " divergence in
UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);

T = genKey();

mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;

pos = r.nextInt(2);

try {
    Rm = m.remove(pos);
}
catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
catch (IllegalArgumentException e) { mThrowsIllegal = true; }
catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

try {
    Rt = t.remove(pos);
}
catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
catch (IllegalArgumentException e) { tThrowsIllegal = true; }
catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }

ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): remove() at " + pos + "
divergence in java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement
+ ") " + m);

```

```

    ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): remove() at " + pos + " divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
    ensure(mThrowsIndex == tThrowsIndex, "Error (" + level + ", " + seed + "): remove() at " + pos + " divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
    ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + level + ", " + seed + "): remove() at " + pos + " divergence
in UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);
    if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex && !mThrowsUnsupp) ensure(Rm == Rt || Rm
!= null && Rm.equals(Rt), "Error (" + level + ", " + seed + "): divergence in remove() at " + pos + " between t and
m " + m);
}

```

```

ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after removal " + m);
ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after removal " + m);

```

```

/* Now we add and remove random collections in m and t, checking that the result is the same. */

```

```

for(int i=0; i<20*n; i++) {
    KEY_TYPE T = genKey();

```

```

    mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;

```

```

    try {
        rm = m.addAll(java.util.Collections.singleton(KEY2OBJ(T)));
    }
    catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
    catch (IllegalArgumentException e) { mThrowsIllegal = true; }
    catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
    catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

```

```

    try {
        rt = t.addAll(java.util.Collections.singleton(KEY2OBJ(T)));
    }
    catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
    catch (IllegalArgumentException e) { tThrowsIllegal = true; }
    catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
    catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }

```

```

    ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): addAll() divergence in
java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ") " + m);
    ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): addAll() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
    ensure(mThrowsIndex == tThrowsIndex, "Error (" + level + ", " + seed + "): addAll() divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
    ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + level + ", " + seed + "): addAll() divergence in
UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);
    if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex && !mThrowsUnsupp) ensure(rm == rt, "Error
(" + level + ", " + seed + "): divergence in addAll() between t and m " + m);

```

```

T = genKey();

mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;

try {
    rm = m.removeAll(java.util.Collections.singleton(KEY2OBJ(T)));
}
catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
catch (IllegalArgumentException e) { mThrowsIllegal = true; }
catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

try {
    rt = t.removeAll(java.util.Collections.singleton(KEY2OBJ(T)));
}
catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
catch (IllegalArgumentException e) { tThrowsIllegal = true; }
catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }

ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): removeAll() divergence in
java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ") " + m);
ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): removeAll() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
ensure(mThrowsIndex == tThrowsIndex, "Error (" + level + ", " + seed + "): removeAll() divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + level + ", " + seed + "): removeAll() divergence in
UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);
if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex && !mThrowsUnsupp) ensure(rm == rt, "Error
(" + level + ", " + seed + "): divergence in removeAll() between t and m " + m);
}

ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after set removal " + m);
ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after set removal " + m);

/* Now we add random collections at specific positions in m and t, checking that the result is the same. */

for(int i=0; i<20*n; i++) {
    KEY_TYPE T = genKey();

    mThrowsIndex = tThrowsIndex = mThrowsNoElement = tThrowsNoElement = mThrowsIllegal = tThrowsIllegal =
mThrowsUnsupp = tThrowsUnsupp = false;

    int pos = r.nextInt(2);

```

```

try {
    rm = m.addAll(pos, java.util.Collections.singleton(KEY2OBJ(T)));
}
catch (java.util.NoSuchElementException e) { mThrowsNoElement = true; }
catch (IllegalArgumentException e) { mThrowsIllegal = true; }
catch (IndexOutOfBoundsException e) { mThrowsIndex = true; }
catch (UnsupportedOperationException e) { mThrowsUnsupp = true; }

try {
    rt = t.addAll(pos, java.util.Collections.singleton(KEY2OBJ(T)));
}
catch (java.util.NoSuchElementException e) { tThrowsNoElement = true; }
catch (IllegalArgumentException e) { tThrowsIllegal = true; }
catch (IndexOutOfBoundsException e) { tThrowsIndex = true; }
catch (UnsupportedOperationException e) { tThrowsUnsupp = true; }

ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): addAll() at " + pos + "
divergence in java.util.NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement
+ ") " + m);
ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): addAll() at " + pos + " divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ") " + m);
ensure(mThrowsIndex == tThrowsIndex, "Error (" + level + ", " + seed + "): addAll() at " + pos + " divergence in
IndexOutOfBoundsException for " + T + " (" + mThrowsIndex + ", " + tThrowsIndex + ") " + m);
ensure(mThrowsUnsupp == tThrowsUnsupp, "Error (" + level + ", " + seed + "): addAll() at " + pos + " divergence
in UnsupportedOperationException for " + T + " (" + mThrowsUnsupp + ", " + tThrowsUnsupp + ") " + m);
if (!mThrowsNoElement && !mThrowsIllegal && !mThrowsIndex && !mThrowsUnsupp) ensure(rm == rt, "Error
(" + level + ", " + seed + "): divergence in addAll() at " + pos + " between t and m " + m);

}

ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after set removal " + m);
ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after set removal " + m);

/* Now we check that m actually holds the same data. */

for(java.util.Iterator i=t.iterator(); i.hasNext();) {
    ensure(m.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after removal (iterating on
t)");
}

/* Now we check that m actually holds that data, but iterating on m. */

for(java.util.Iterator i=m.listIterator(); i.hasNext();) {
    ensure(t.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after removal (iterating on
m)");
}

if (m instanceof Singleton) {

```

```

ensure(m.equals(((Singleton)m).clone()), "Error (" + level + ", " + seed + "): m does not equal m.clone()");
ensure(((Singleton)m).clone().equals(m), "Error (" + level + ", " + seed + "): m.clone() does not equal m");
}

int h = m.hashCode();

/* Now we save and read m. */

LIST m2 = null;

try {
java.io.File ff = new java.io.File("it.unimi.dsi.fastutil.test");
java.io.OutputStream os = new java.io.FileOutputStream(ff);
java.io.ObjectOutputStream oos = new java.io.ObjectOutputStream(os);

oos.writeObject(m);
oos.close();

java.io.InputStream is = new java.io.FileInputStream(ff);
java.io.ObjectInputStream ois = new java.io.ObjectInputStream(is);

m2 = (LIST)ois.readObject();
ois.close();
ff.delete();
}
catch(Exception e) {
e.printStackTrace();
System.exit(1);
}

#if ! KEY_CLASS_Reference

ensure(m2.hashCode() == h, "Error (" + level + ", " + seed + "): hashCode() changed after save/read");

/* Now we check that m2 actually holds that data. */

ensure(m2.equals(t), "Error (" + level + ", " + seed + "): ! m2.equals(t) after save/read");
ensure(t.equals(m2), "Error (" + level + ", " + seed + "): ! t.equals(m2) after save/read");
#endif

if (! m.isEmpty()) {
int start = r.nextInt(m.size());
int end = start + r.nextInt(m.size() - start);
//System.err.println("Checking subList from " + start + " to " + end + " (level=" + (level+1) + ")...");
testLists(k, m.subList(start, end), t.subList(start, end), level - 1);

ensure(m.equals(t), "Error (" + level + ", " + seed + m + t + "): ! m.equals(t) after subList");
ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after subList");
}

```

```

    }

    return;
}

private static void test() {
    KEY_TYPE k = genKey();
    LIST m = new Singleton(k);
    List u = java.util.Collections.singletonList(KEY2OBJ(k));
    testLists(k, m, java.util.Collections.unmodifiableList(u), 3);
    System.out.println("Test OK");
}

private static long seed = System.currentTimeMillis();
private static java.util.Random r = new java.util.Random(seed);

private static java.text.NumberFormat format = new java.text.DecimalFormat("#,###.00");
private static java.text.FieldPosition fp = new java.text.FieldPosition(0);

private static String format(double d) {
    StringBuffer s = new StringBuffer();
    return format.format(d, s, fp).toString();
}

private static void fatal(String msg) {
    System.out.println(msg);
    System.exit(1);
}

private static void ensure(boolean cond, String msg) {
    if (cond) return;
    fatal(msg);
}

/** This method expects as first argument a lower-cased type (e.g., "int"),
 * and as second optional argument a seed. */

public static void main(String arg[]) throws Exception {
    if (arg.length > 1) r = new java.util.Random(seed = Long.parseLong(arg[1]));

    try {
        test();
    } catch(Throwable e) {
        e.printStackTrace(System.err);
        System.err.println("seed: " + seed);
    }
}

```

#endif

}

Found in path(s):

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/drv/Lists.drv

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2005-2017 Sebastiano Vigna

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/src/it/unimi/dsi/fastutil/io/FastBufferedInputStream.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/src/it/unimi/dsi/fastutil/io/FastMultiByteArrayInputStream.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/src/it/unimi/dsi/fastutil/io/MeasurableOutputStream.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/src/it/unimi/dsi/fastutil/io/MeasurableStream.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/src/it/unimi/dsi/fastutil/io/FastByteArrayInputStream.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/drv/TextIO.drv

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/src/it/unimi/dsi/fastutil/io/MeasurableInputStream.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/src/it/unimi/dsi/fastutil/io/RepositionableStream.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/drv/BinIO.drv

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/src/it/unimi/dsi/fastutil/io/InspectableFileCachedInputStream.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-

a775574/src/it/unimi/dsi/fastutil/io/FastByteArrayOutputStream.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-

a775574/src/it/unimi/dsi/fastutil/io/FastBufferedOutputStream.java

No license file was found, but licenses were detected in source scan.

```
<project>
  <modelVersion>4.0.0</modelVersion>
  <groupId>it.unimi.dsi</groupId>
  <artifactId>fastutil</artifactId>
  <packaging>jar</packaging>
  <name>fastutil</name>
  <version>VERSION</version>
  <description>fastutil extends the Java Collections Framework by providing type-specific maps, sets, lists and
  priority queues with a small memory footprint and fast access and insertion; provides also big (64-bit) arrays, sets
  and lists, and fast, practical I/O classes for binary and text files.</description>
  <url>http://fastutil.di.unimi.it/</url>
  <licenses>
    <license>
      <name>Apache License, Version 2.0</name>
      <url>http://www.apache.org/licenses/LICENSE-2.0.html</url>
      <distribution>repo</distribution>
    </license>
  </licenses>
  <scm>
    <connection>scm:git://github.com/vigna/fastutil.git</connection>
    <url>https://github.com/vigna/fastutil</url>
  </scm>
  <developers>
    <developer>
      <id>vigna</id>
      <name>Sebastiano Vigna</name>
      <email>vigna@di.unimi.it</email>
    </developer>
  </developers>
  <properties>
    <maven.compiler.source>1.8</maven.compiler.source>
    <maven.compiler.target>1.8</maven.compiler.target>
  </properties>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.12</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>
```

Found in path(s):

```
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-  
a775574/pom-model.xml
```

No license file was found, but licenses were detected in source scan.

```
/*
```

```
* Copyright (C) 2002-2017 Sebastiano Vigna
```

```
*
```

```
* Licensed under the Apache License, Version 2.0 (the "License");
```

```
* you may not use this file except in compliance with the License.
```

```
* You may obtain a copy of the License at
```

```
*
```

```
* http://www.apache.org/licenses/LICENSE-2.0
```

```
*
```

```
* Unless required by applicable law or agreed to in writing, software
```

```
* distributed under the License is distributed on an "AS IS" BASIS,
```

```
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
```

```
* See the License for the specific language governing permissions and
```

```
* limitations under the License.
```

```
*/
```

```
package PACKAGE;
```

```
import it.unimi.dsi.fastutil.objects.Object2IntOpenHashMap;
```

```
import java.io.Serializable;
```

```
import java.util.concurrent.locks.ReentrantReadWriteLock;
```

```
import java.util.concurrent.locks.ReentrantReadWriteLock.ReadLock;
```

```
import java.util.concurrent.locks.ReentrantReadWriteLock.WriteLock;
```

```
/** A concurrent counting map. The map is made by a number of stripes (instances of {@link  
Object2IntOpenHashMap})
```

```
* which are accessed independently
```

```
* using a {@link ReentrantReadWriteLock}. Only one thread can write in a stripe at a time, but different stripes
```

```
* can be modified independently and read access can happen concurrently on each stripe.
```

```
*
```

```
* @param <K> the type of keys.
```

```
*/
```

```
public class STRIPED_OPEN_HASH_MAP_KEY_VALUE_GENERIC extends ABSTRACT_MAP
```

```
KEY_VALUE_GENERIC implements java.io.Serializable, Cloneable {
```

```
    private static final long serialVersionUID = 1L;
```

```
    /** The stripes. Keys are distributed among them using the lower bits of their {@link Object#hashCode()}. */
```

```
    private final OPEN_HASH_MAP_KEY_VALUE_GENERIC[] map;
```

```
    /** An array of locks parallel to {@link #map}, protecting each stripe. */
```

```
    private final transient ReentrantReadWriteLock[] lock;
```

```
    /** {@link #map.length} - 1, cached. */
```

```

private final int mask;

/** Creates a new concurrent counting map with concurrency level equal to {@link
Runtime#availableProcessors()}. */
public STRIPED_OPEN_HASH_MAP() {
    this(Runtime.getRuntime().availableProcessors());
}

/** Creates a new concurrent counting map.
 *
 * @param concurrencyLevel the number of stripes (it will be {@linkplain Integer#highestOneBit(int) forced to be a
power of two}); ideally, as large as the number of threads that will ever access
 * this map, but higher values require more space.
 */
SUPPRESS_WARNINGS_KEY_UNCHECKED
public STRIPED_OPEN_HASH_MAP(final int concurrencyLevel) {
    map = new OPEN_HASH_MAP[Integer.highestOneBit(concurrencyLevel)];
    lock = new ReentrantReadWriteLock[map.length];
    for(int i = map.length; i-- != 0;) {
        map[i] = new OPEN_HASH_MAP KEY_VALUE_GENERIC_DIAMOND();
        lock[i] = new ReentrantReadWriteLock();
    }
    mask = map.length - 1;
}

#if KEYS_PRIMITIVE

public VALUE_GENERIC_CLASS get(final KEY_CLASS k) {
    final int stripe = KEY2INTHASH(k) & mask;
    final ReadLock readLock = lock[stripe].readLock();
    try {
        readLock.lock();
        return map[stripe].get(k);
    }
    finally {
        readLock.unlock();
    }
}

#endif

SUPPRESS_WARNINGS_KEY_UNCHECKED
public VALUE_GENERIC_TYPE GET_VALUE(final KEY_TYPE k) {
    final int stripe = KEY2INTHASH(k) & mask;
    final ReadLock readLock = lock[stripe].readLock();
    try {
        readLock.lock();
        return map[stripe].GET_VALUE(k);
    }
}

```

```

    }
    finally {
        readLock.unlock();
    }
}

public VALUE_GENERIC_TYPE put(final KEY_GENERIC_TYPE k, final VALUE_GENERIC_TYPE v) {
    final int stripe = KEY2INTHASH(k) & mask;
    final WriteLock writeLock = lock[stripe].writeLock();
    try {
        writeLock.lock();
        return map[stripe].put(k, v);
    }
    finally {
        writeLock.unlock();
    }
}

public VALUE_GENERIC_TYPE putIfAbsent(final KEY_GENERIC_TYPE k, final VALUE_GENERIC_TYPE
v) {
    final int stripe = KEY2INTHASH(k) & mask;
    final WriteLock writeLock = lock[stripe].writeLock();
    try {
        writeLock.lock();
        if (map[stripe].containsKey(k)) return map[stripe].get(k);
        return map[stripe].put(k, v);
    }
    finally {
        writeLock.unlock();
    }
}

#if VALUES_PRIMITIVE || KEYS_PRIMITIVE

public VALUE_GENERIC_CLASS put(final KEY_GENERIC_CLASS ok, final VALUE_GENERIC_CLASS ov)
{
    final int stripe = KEY2INTHASH(ok) & mask;
    final WriteLock writeLock = lock[stripe].writeLock();
    try {
        writeLock.lock();
        return map[stripe].put(ok, ov);
    }
    finally {
        writeLock.unlock();
    }
}

```

```

public VALUE_GENERIC_CLASS putIfAbsent(final KEY_GENERIC_CLASS ok, final
VALUE_GENERIC_CLASS ov) {
    final int stripe = KEY2INTHASH(ok) & mask;
    final WriteLock writeLock = lock[stripe].writeLock();
    try {
        writeLock.lock();
        if (map[stripe].containsKey(ok)) return map[stripe].get(ok);
        return map[stripe].put(ok, ov);
    }
    finally {
        writeLock.unlock();
    }
}

#endif

public int size() {
    int size = 0;
    for(int stripe = lock.length; stripe-- != 0;) {
        final ReadLock readLock = lock[stripe].readLock();
        try {
            readLock.lock();
            size += map[stripe].size();
        }
        finally {
            readLock.unlock();
        }
    }

    return size;
}

public FastEntrySet KEY_VALUE_GENERIC ENTRYSET() {
    throw new UnsupportedOperationException();
}
}

```

Found in path(s):

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/drv/StripedOpenHashMap.drv

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2002-2017 Sebastiano Vigna

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

```
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

```
package PACKAGE;
```

```
import java.util.Collection;
```

```
/** A type-specific {@link Collection}; provides some additional methods
* that use polymorphism to avoid (un)boxing.
*
* <p>Additionally, this class defines strengthens (again) {@link #iterator()}.
*
* @see Collection
*/
```

```
public interface COLLECTION KEY_GENERIC extends Collection<KEY_GENERIC_CLASS>,
KEY_ITERABLE KEY_GENERIC {
```

```
/** Returns a type-specific iterator on the elements of this collection.
*
* <p>Note that this specification strengthens the one given in
* {@link java.lang.Iterable#iterator()}, which was already
* strengthened in the corresponding type-specific class,
* but was weakened by the fact that this interface extends {@link Collection}.
*
* @return a type-specific iterator on the elements of this collection.
*/
```

```
@Override
```

```
KEY_ITERATOR KEY_GENERIC iterator();
```

```
#if KEYS_PRIMITIVE
```

```
/** Ensures that this collection contains the specified element (optional operation).
* @see Collection#add(Object)
*/
```

```
boolean add(KEY_TYPE key);
```

```
/** Returns {@code true} if this collection contains the specified element.
* @see Collection#contains(Object)
*/
```

```

boolean contains(KEY_TYPE key);

/** Removes a single instance of the specified element from this
 * collection, if it is present (optional operation).
 *
 * <p>Note that this method should be called { @link java.util.Collection#remove(Object) remove()}, but the clash
 * with the similarly named index-based method in the { @link java.util.List} interface
 * forces us to use a distinguished name. For simplicity, the set interfaces reinstates
 * { @code remove()}.
 *
 * @see Collection#remove(Object)
 */
boolean rem(KEY_TYPE key);

/** { @inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead.
 */
@Deprecated
@Override
default boolean add(final KEY_GENERIC_CLASS key) {
    return add(KEY_CLASS2TYPE(key));
}

/** { @inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead.
 */
@Deprecated
@Override
default boolean contains(final Object key) {
    if (key == null) return false;
    return contains(KEY_OBJ2TYPE(key));
}

/** { @inheritDoc}
 * @deprecated Please use (and implement) the { @code rem()} method instead.
 */
@Deprecated
@Override
default boolean remove(final Object key) {
    if (key == null) return false;
    return rem(KEY_OBJ2TYPE(key));
}

/** Returns a primitive type array containing the items of this collection.
 * @return a primitive type array containing the items of this collection.
 * @see Collection#toArray()
 */
KEY_TYPE[] TO_KEY_ARRAY();

```

```
/** Returns a primitive type array containing the items of this collection.
 *
 * <p>Note that, contrarily to { @link Collection#toArray(Object[])}, this
 * methods just writes all elements of this collection: no special
 * value will be added after the last one.
 *
 * @param a if this array is big enough, it will be used to store this collection.
 * @return a primitive type array containing the items of this collection.
 * @see Collection#toArray(Object[])
 * @deprecated Please use { @code toArray()} instead&mdash;this method is redundant and will be removed in the
future.
 */
```

@Deprecated

```
KEY_TYPE[] TO_KEY_ARRAY(KEY_TYPE a[]);
```

```
/** Returns an array containing all of the elements in this collection; the runtime type of the returned array is that of
the specified array.
 *
 * <p>Note that, contrarily to { @link Collection#toArray(Object[])}, this
 * methods just writes all elements of this collection: no special
 * value will be added after the last one.
 *
 * @param a if this array is big enough, it will be used to store this collection.
 * @return a primitive type array containing the items of this collection.
 * @see Collection#toArray(Object[])
 */
```

```
KEY_TYPE[] toArray(KEY_TYPE a[]);
```

```
/** Adds all elements of the given type-specific collection to this collection.
 *
 * @param c a type-specific collection.
 * @see Collection#addAll(Collection)
 * @return { @code true} if this collection changed as a result of the call.
 */
```

```
boolean addAll(COLLECTION c);
```

```
/** Checks whether this collection contains all elements from the given type-specific collection.
 *
 * @param c a type-specific collection.
 * @see Collection#containsAll(Collection)
 * @return { @code true} if this collection contains all elements of the argument.
 */
```

```
boolean containsAll(COLLECTION c);
```

```
/** Remove from this collection all elements in the given type-specific collection.
 *
 * @param c a type-specific collection.
```

```

* @see Collection#removeAll(Collection)
* @return { @code true } if this collection changed as a result of the call.
*/
boolean removeAll(COLLECTION c);

#ifdef JDK_PRIMITIVE_PREDICATE
/** { @inheritDoc}
* @deprecated Please use the corresponding type-specific method instead.
*/
@Deprecated
@Override
default boolean removeIf(final java.util.function.Predicate<? super KEY_GENERIC_CLASS> filter) {
return removeIf((JDK_PRIMITIVE_PREDICATE) key -> filter.test(KEY2OBJ(KEY_NARROWING(key))));
}

/** Remove from this collection all elements which satisfy the given predicate.
*
* @param filter a predicate which returns { @code true } for elements to be
* removed.
* @see Collection#removeIf(java.util.function.Predicate)
* @return { @code true } if any elements were removed.
*/
@SuppressWarnings("overloads")
default boolean removeIf(final JDK_PRIMITIVE_PREDICATE filter) {
java.util.Objects.requireNonNull(filter);
boolean removed = false;
final KEY_ITERATOR each = iterator();
while (each.hasNext()) {
if (filter.test(each.NEXT_KEY())) {
each.remove();
removed = true;
}
}
return removed;
}
#endif

/** Retains in this collection only elements from the given type-specific collection.
*
* @param c a type-specific collection.
* @see Collection#retainAll(Collection)
* @return { @code true } if this collection changed as a result of the call.
*/
boolean retainAll(COLLECTION c);

#endif
}

```

Found in path(s):

```
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-
a775574/drv/Collection.drv
```

No license file was found, but licenses were detected in source scan.

```
/*
```

```
* Copyright (C) 2002-2017 Sebastiano Vigna
```

```
*
```

```
* Licensed under the Apache License, Version 2.0 (the "License");
```

```
* you may not use this file except in compliance with the License.
```

```
* You may obtain a copy of the License at
```

```
*
```

```
* http://www.apache.org/licenses/LICENSE-2.0
```

```
*
```

```
* Unless required by applicable law or agreed to in writing, software
```

```
* distributed under the License is distributed on an "AS IS" BASIS,
```

```
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
```

```
* See the License for the specific language governing permissions and
```

```
* limitations under the License.
```

```
*/
```

```
package PACKAGE;
```

```
import it.unimi.dsi.fastutil.objects.AbstractObjectSortedSet;
```

```
import it.unimi.dsi.fastutil.objects.ObjectBidirectionalIterator;
```

```
import it.unimi.dsi.fastutil.objects.ObjectListIterator;
```

```
import it.unimi.dsi.fastutil.objects.ObjectSortedSet;
```

```
import VALUE_PACKAGE.VALUE_COLLECTION;
```

```
import VALUE_PACKAGE.VALUE_ABSTRACT_COLLECTION;
```

```
import VALUE_PACKAGE.VALUE_ITERATOR;
```

```
import java.util.Comparator;
```

```
import java.util.Iterator;
```

```
import java.util.Map;
```

```
import java.util.SortedMap;
```

```
import java.util.NoSuchElementException;
```

```
#if VALUES_PRIMITIVE
```

```
import VALUE_PACKAGE.VALUE_LIST_ITERATOR;
```

```
#endif
```

```
/** A type-specific red-black tree map with a fast, small-footprint implementation.
```

```
*
```

```
* <p>The iterators provided by the views of this class are type-specific { @linkplain
```

```
* it.unimi.dsi.fastutil.BidirectionalIterator bidirectional iterators}.
```

```

* Moreover, the iterator returned by {@code iterator()} can be safely cast
* to a type-specific {@linkplain java.util.ListIterator list iterator}.
*
*/

public class RB_TREE_MAP KEY_VALUE_GENERIC extends ABSTRACT_SORTED_MAP
KEY_VALUE_GENERIC implements java.io.Serializable, Cloneable {

    /** A reference to the root entry. */
    protected transient Entry KEY_VALUE_GENERIC tree;

    /** Number of entries in this map. */
    protected int count;

    /** The first key in this map. */
    protected transient Entry KEY_VALUE_GENERIC firstEntry;

    /** The last key in this map. */
    protected transient Entry KEY_VALUE_GENERIC lastEntry;

    /** Cached set of entries. */
    protected transient ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> entries;

    /** Cached set of keys. */
    protected transient SORTED_SET KEY_GENERIC keys;

    /** Cached collection of values. */
    protected transient VALUE_COLLECTION VALUE_GENERIC values;

    /** The value of this variable remembers, after a {@code put()}
    * or a {@code remove()}, whether the <em>domain</em> of the map
    * has been modified. */
    protected transient boolean modified;

    /** This map's comparator, as provided in the constructor. */
    protected Comparator<? super KEY_GENERIC_CLASS> storedComparator;

    /** This map's actual comparator; it may differ from {@link #storedComparator} because it is
    always a type-specific comparator, so it could be derived from the former by wrapping. */
    protected transient KEY_COMPARATOR KEY_SUPER_GENERIC actualComparator;

    private static final long serialVersionUID = -7046029254386353129L;

    {
        allocatePaths();
    }

    /** Creates a new empty tree map.

```

```

*/

public RB_TREE_MAP() {
    tree = null;
    count = 0;
}

/** Generates the comparator that will be actually used.
 *
 * <p>When a given {@link Comparator} is specified and stored in {@link
 * #storedComparator}, we must check whether it is type-specific. If it is
 * so, we can use it directly, and we store it in {@link #actualComparator}. Otherwise,
 * we adapt it using a helper static method.
 */
private void setActualComparator() {
#ifdef KEY_CLASS_Object
    actualComparator = storedComparator;
#else
    actualComparator = COMPARATORS.AS_KEY_COMPARATOR(storedComparator);
#endif
}

/** Creates a new empty tree map with the given comparator.
 *
 * @param c a (possibly type-specific) comparator.
 */
public RB_TREE_MAP(final Comparator<? super KEY_GENERIC_CLASS> c) {
    this();
    storedComparator = c;
    setActualComparator();
}

/** Creates a new tree map copying a given map.
 *
 * @param m a {@link Map} to be copied into the new tree map.
 */
public RB_TREE_MAP(final Map<? extends KEY_GENERIC_CLASS, ? extends VALUE_GENERIC_CLASS>
m) {
    this();
    putAll(m);
}

/** Creates a new tree map copying a given sorted map (and its {@link Comparator}).
 *

```

```

* @param m a {@link SortedMap} to be copied into the new tree map.
*/

public RB_TREE_MAP(final SortedMap<KEY_GENERIC_CLASS,VALUE_GENERIC_CLASS> m) {
    this(m.comparator());
    putAll(m);
}

/** Creates a new tree map copying a given map.
 *
 * @param m a type-specific map to be copied into the new tree map.
 */

public RB_TREE_MAP(final MAP KEY_VALUE_EXTENDS_GENERIC m) {
    this();
    putAll(m);
}

/** Creates a new tree map copying a given sorted map (and its {@link Comparator}).
 *
 * @param m a type-specific sorted map to be copied into the new tree map.
 */

public RB_TREE_MAP(final SORTED_MAP KEY_VALUE_GENERIC m) {
    this(m.comparator());
    putAll(m);
}

/** Creates a new tree map using the elements of two parallel arrays and the given comparator.
 *
 * @param k the array of keys of the new tree map.
 * @param v the array of corresponding values in the new tree map.
 * @param c a (possibly type-specific) comparator.
 * @throws IllegalArgumentException if {@code k} and {@code v} have different lengths.
 */

public RB_TREE_MAP(final KEY_GENERIC_TYPE[] k, final VALUE_GENERIC_TYPE v[], final
Comparator<? super KEY_GENERIC_CLASS> c) {
    this(c);
    if (k.length != v.length) throw new IllegalArgumentException("The key array and the value array have different
lengths (" + k.length + " and " + v.length + ")");
    for(int i = 0; i < k.length; i++) this.put(k[i], v[i]);
}

/** Creates a new tree map using the elements of two parallel arrays.
 *
 * @param k the array of keys of the new tree map.
 * @param v the array of corresponding values in the new tree map.

```

```

* @throws IllegalArgumentException if {@code k} and {@code v} have different lengths.
*/

public RB_TREE_MAP(final KEY_GENERIC_TYPE[] k, final VALUE_GENERIC_TYPE v[]) {
    this(k, v, null);
}

/*
* The following methods implements some basic building blocks used by
* all accessors. They are (and should be maintained) identical to those used in RBTreeSet.drv.
*
* The put()/remove() code is derived from Ben Pfaff's GNU libavl
* (http://www.msu.edu/~pfaffben/avl/). If you want to understand what's
* going on, you should have a look at the literate code contained therein
* first.
*/

/** Compares two keys in the right way.
*
* <p>This method uses the {@link #actualComparator} if it is non-{@code null}.
* Otherwise, it resorts to primitive type comparisons or to {@link Comparable#compareTo(Object) compareTo()}.
*
* @param k1 the first key.
* @param k2 the second key.
* @return a number smaller than, equal to or greater than 0, as usual
* (i.e., when  $k1 < k2$ ,  $k1 = k2$  or  $k1 > k2$ , respectively).
*/

SUPPRESS_WARNINGS_KEY_UNCHECKED
final int compare(final KEY_GENERIC_TYPE k1, final KEY_GENERIC_TYPE k2) {
    return actualComparator == null ? KEY_CMP(k1, k2) : actualComparator.compare(k1, k2);
}

/** Returns the entry corresponding to the given key, if it is in the tree; {@code null}, otherwise.
*
* @param k the key to search for.
* @return the corresponding entry, or {@code null} if no entry with the given key exists.
*/

final Entry KEY_VALUE_GENERIC findKey(final KEY_GENERIC_TYPE k) {
    Entry KEY_VALUE_GENERIC e = tree;
    int cmp;

    while (e != null && (cmp = compare(k, e.key)) != 0) e = cmp < 0 ? e.left() : e.right();

    return e;
}

```

```

/** Locates a key.
 *
 * @param k a key.
 * @return the last entry on a search for the given key; this will be
 * the given key, if it present; otherwise, it will be either the smallest greater key or the greatest smaller key.
 */

final Entry KEY_VALUE_GENERIC locateKey(final KEY_GENERIC_TYPE k) {
    Entry KEY_VALUE_GENERIC e = tree, last = tree;
    int cmp = 0;

    while (e != null && (cmp = compare(k, e.key)) != 0) {
        last = e;
        e = cmp < 0 ? e.left() : e.right();
    }

    return cmp == 0 ? e : last;
}

/** This vector remembers the path and the direction followed during the
 * current insertion. It suffices for about  $2^{32}$  entries. */
private transient boolean dirPath[];
private transient Entry KEY_VALUE_GENERIC nodePath[];

SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED_RAWTYPES
private void allocatePaths() {
    dirPath = new boolean[64];
    nodePath = new Entry[64];
}

#if VALUES_PRIMITIVE && !VALUE_CLASS_Boolean
/** Adds an increment to value currently associated with a key.
 *
 * <p>Note that this method respects the { @linkplain #defaultReturnValue() default return value } semantics: when
 * called with a key that does not currently appears in the map, the key
 * will be associated with the default return value plus
 * the given increment.
 *
 * @param k the key.
 * @param incr the increment.
 * @return the old value, or the { @linkplain #defaultReturnValue() default return value } if no value was present for
the given key.
 */
public VALUE_GENERIC_TYPE addTo(final KEY_GENERIC_TYPE k, final VALUE_GENERIC_TYPE incr) {
    Entry KEY_VALUE_GENERIC e = add(k);
    final VALUE_GENERIC_TYPE oldValue = e.value;
    e.value += incr;
}

```

```

return oldValue;
}
#endif

@Override
public VALUE_GENERIC_TYPE put(final KEY_GENERIC_TYPE k, final VALUE_GENERIC_TYPE v) {
    Entry KEY_VALUE_GENERIC e = add(k);
    final VALUE_GENERIC_TYPE oldValue = e.value;
    e.value = v;
    return oldValue;
}

/** Returns a node with key k in the balanced tree, creating one with defRetValue if necessary.
 *
 * @param k the key
 * @return a node with key k. If a node with key k already exists, then that node is returned,
 * otherwise a new node with defRetValue is created ensuring that the tree is balanced
 * after creation of the node.
 */
private Entry KEY_VALUE_GENERIC add(final KEY_GENERIC_TYPE k) {
    /* After execution of this method, modified is true iff a new entry has been inserted. */
    modified = false;
    int maxDepth = 0;

    Entry KEY_VALUE_GENERIC e;

    if (tree == null) { // The case of the empty tree is treated separately.
        count++;
        e = tree = lastEntry = firstEntry = new Entry KEY_VALUE_GENERIC_DIAMOND(k, defRetValue);
    }
    else {
        Entry KEY_VALUE_GENERIC p = tree;
        int cmp, i = 0;

        while(true) {
            if ((cmp = compare(k, p.key)) == 0) {
                // We clean up the node path, or we could have stale references later.
                while(i-- != 0) nodePath[i] = null;
                return p;
            }

            nodePath[i] = p;

            if (dirPath[i++] = cmp > 0) {
                if (p.succ()) {
                    count++;
                    e = new Entry KEY_VALUE_GENERIC_DIAMOND(k, defRetValue);

```

```

if (p.right == null) lastEntry = e;

e.left = p;
e.right = p.right;

p.right(e);

break;
}

p = p.right;
}
else {
if (p.pred()) {
count++;
e = new Entry KEY_VALUE_GENERIC_DIAMOND(k, defRetVal);

if (p.left == null) firstEntry = e;

e.right = p;
e.left = p.left;

p.left(e);

break;
}

p = p.left;
}
}

modified = true;
maxDepth = i--;

while(i > 0 && ! nodePath[i].black()) {
if (! dirPath[i - 1]) {
Entry KEY_VALUE_GENERIC y = nodePath[i - 1].right;

if (! nodePath[i - 1].succ() && ! y.black()) {
nodePath[i].black(true);
y.black(true);
nodePath[i - 1].black(false);
i -= 2;
}
else {
Entry KEY_VALUE_GENERIC x;

if (! dirPath[i]) y = nodePath[i];

```

```

else {
    x = nodePath[i];
    y = x.right;
    x.right = y.left;
    y.left = x;
    nodePath[i - 1].left = y;

    if (y.pred()) {
        y.pred(false);
        x.succ(y);
    }
}

x = nodePath[i - 1];
x.black(false);
y.black(true);

x.left = y.right;
y.right = x;
if (i < 2) tree = y;
else {
    if (dirPath[i - 2]) nodePath[i - 2].right = y;
    else nodePath[i - 2].left = y;
}

if (y.succ()) {
    y.succ(false);
    x.pred(y);
}
break;
}
}
else {
    Entry KEY_VALUE_GENERIC y = nodePath[i - 1].left;

    if (! nodePath[i - 1].pred() && ! y.black()) {
        nodePath[i].black(true);
        y.black(true);
        nodePath[i - 1].black(false);
        i -= 2;
    }
    else {
        Entry KEY_VALUE_GENERIC x;

        if (dirPath[i]) y = nodePath[i];
        else {
            x = nodePath[i];
            y = x.left;

```

```

x.left = y.right;
y.right = x;
nodePath[i - 1].right = y;

if (y.succ()) {
  y.succ(false);
  x.pred(y);
}

}

x = nodePath[i - 1];
x.black(false);
y.black(true);

x.right = y.left;
y.left = x;
if (i < 2) tree = y;
else {
  if (dirPath[i - 2]) nodePath[i - 2].right = y;
  else nodePath[i - 2].left = y;
}

if (y.pred()){
  y.pred(false);
  x.succ(y);
}

break;
}
}
}
}
tree.black(true);
// We clean up the node path, or we could have stale references later.
while(maxDepth-- != 0) nodePath[maxDepth] = null;
return e;
}

/* After execution of this method, { @link #modified } is true iff an entry
has been deleted. */

SUPPRESS_WARNINGS_KEY_UNCHECKED
@Override
public VALUE_GENERIC_TYPE REMOVE_VALUE(final KEY_TYPE k) {
  modified = false;

```

```

if (tree == null) return defRetVal;

Entry KEY_VALUE_GENERIC p = tree;
int cmp;
int i = 0;
final KEY_GENERIC_TYPE kk = KEY_GENERIC_CAST k;

while(true) {
if ((cmp = compare(kk, p.key)) == 0) break;

dirPath[i] = cmp > 0;
nodePath[i] = p;

if (dirPath[i++] {
if ((p = p.right()) == null) {
// We clean up the node path, or we could have stale references later.
while(i-- != 0) nodePath[i] = null;
return defRetVal;
}
}
else {
if ((p = p.left()) == null) {
// We clean up the node path, or we could have stale references later.
while(i-- != 0) nodePath[i] = null;
return defRetVal;
}
}
}

if (p.left == null) firstEntry = p.next();
if (p.right == null) lastEntry = p.prev();

if (p.succ()) {
if (p.pred()) {
if (i == 0) tree = p.left;
else {
if (dirPath[i - 1]) nodePath[i - 1].succ(p.right);
else nodePath[i - 1].pred(p.left);
}
}
else {
p.prev().right = p.right;

if (i == 0) tree = p.left;
else {
if (dirPath[i - 1]) nodePath[i - 1].right = p.left;
else nodePath[i - 1].left = p.left;
}
}
}
}

```

```

    }
    }
    }
else {
    boolean color;
    Entry KEY_VALUE_GENERIC r = p.right;

    if (r.pred()) {
        r.left = p.left;
        r.pred(p.pred());
        if (! r.pred()) r.prev().right = r;
        if (i == 0) tree = r;
        else {
            if (dirPath[i - 1]) nodePath[i - 1].right = r;
            else nodePath[i - 1].left = r;
        }

        color = r.black();
        r.black(p.black());
        p.black(color);
        dirPath[i] = true;
        nodePath[i++] = r;
    }
    else {
        Entry KEY_VALUE_GENERIC s;
        int j = i++;

        while(true) {
            dirPath[i] = false;
            nodePath[i++] = r;
            s = r.left;
            if (s.pred()) break;
            r = s;
        }

        dirPath[j] = true;
        nodePath[j] = s;

        if (s.succ()) r.pred(s);
        else r.left = s.right;

        s.left = p.left;

        if (! p.pred()) {
            p.prev().right = s;
            s.pred(false);
        }
    }
}

```

```

s.right(p.right);

color = s.black();
s.black(p.black());
p.black(color);

if (j == 0) tree = s;
else {
    if (dirPath[j - 1]) nodePath[j - 1].right = s;
    else nodePath[j - 1].left = s;
}
}
}

int maxDepth = i;

if (p.black()) {
    for(; i > 0; i--) {
        if (dirPath[i - 1] && ! nodePath[i - 1].succ() ||
            ! dirPath[i - 1] && ! nodePath[i - 1].pred()) {
            Entry KEY_VALUE_GENERIC x = dirPath[i - 1] ? nodePath[i - 1].right : nodePath[i - 1].left;

            if (! x.black()) {
                x.black(true);
                break;
            }
        }

        if (! dirPath[i - 1]) {
            Entry KEY_VALUE_GENERIC w = nodePath[i - 1].right;

            if (! w.black()) {
                w.black(true);
                nodePath[i - 1].black(false);

                nodePath[i - 1].right = w.left;
                w.left = nodePath[i - 1];

                if (i < 2) tree = w;
                else {
                    if (dirPath[i - 2]) nodePath[i - 2].right = w;
                    else nodePath[i - 2].left = w;
                }

                nodePath[i] = nodePath[i - 1];
                dirPath[i] = false;
                nodePath[i - 1] = w;
                if (maxDepth == i++) maxDepth++;
            }
        }
    }
}

```

```

w = nodePath[i - 1].right;
}

if ((w.pred() || w.left.black()) &&
    (w.succ() || w.right.black())) {
    w.black(false);
}
else {
    if (w.succ() || w.right.black()) {
        Entry KEY_VALUE_GENERIC y = w.left;

        y.black(true);
        w.black(false);
        w.left = y.right;
        y.right = w;
        w = nodePath[i - 1].right = y;

        if (w.succ()) {
            w.succ(false);
            w.right.pred(w);
        }
    }

    w.black(nodePath[i - 1].black());
    nodePath[i - 1].black(true);
    w.right.black(true);

    nodePath[i - 1].right = w.left;
    w.left = nodePath[i - 1];

    if (i < 2) tree = w;
    else {
        if (dirPath[i - 2]) nodePath[i - 2].right = w;
        else nodePath[i - 2].left = w;
    }

    if (w.pred()) {
        w.pred(false);
        nodePath[i - 1].succ(w);
    }
    break;
}
else {
    Entry KEY_VALUE_GENERIC w = nodePath[i - 1].left;

    if (! w.black()) {

```

```

w.black (true);
nodePath[i - 1].black(false);

nodePath[i - 1].left = w.right;
w.right = nodePath[i - 1];

if (i < 2) tree = w;
else {
    if (dirPath[i - 2]) nodePath[i - 2].right = w;
    else nodePath[i - 2].left = w;
}

nodePath[i] = nodePath[i - 1];
dirPath[i] = true;
nodePath[i - 1] = w;
if (maxDepth == i++) maxDepth++;

w = nodePath[i - 1].left;
}

if ((w.pred() || w.left.black()) &&
    (w.succ() || w.right.black())) {
    w.black(false);
}
else {
    if (w.pred() || w.left.black()) {
        Entry KEY_VALUE_GENERIC y = w.right;

        y.black(true);
        w.black (false);
        w.right = y.left;
        y.left = w;
        w = nodePath[i - 1].left = y;

        if (w.pred()) {
            w.pred(false);
            w.left.succ(w);
        }
    }
}

w.black(nodePath[i - 1].black());
nodePath[i - 1].black(true);
w.left.black(true);

nodePath[i - 1].left = w.right;
w.right = nodePath[i - 1];

if (i < 2) tree = w;

```

```

else {
    if (dirPath[i - 2]) nodePath[i - 2].right = w;
    else nodePath[i - 2].left = w;
}

if (w.succ()) {
    w.succ(false);
    nodePath[i - 1].pred(w);
}
break;
}
}
}

if (tree != null) tree.black(true);
}

modified = true;
count--;
// We clean up the node path, or we could have stale references later.
while(maxDepth-- != 0) nodePath[maxDepth] = null;
return p.value;
}

```

```

@Override
public boolean containsValue(final VALUE_TYPE v) {
    final ValueIterator i = new ValueIterator();
    VALUE_TYPE ev;

    int j = count;
    while(j-- != 0) {
        ev = i.NEXT_VALUE();
        if (VALUE_EQUALS(ev, v)) return true;
    }

    return false;
}

```

```

@Override
public void clear() {
    count = 0;
    tree = null;
    entries = null;
    values = null;
    keys = null;
    firstEntry = lastEntry = null;
}

```

```

}

/** This class represent an entry in a tree map.
 *
 * <p>We use the only "metadata", i.e., {@link Entry#info}, to store
 * information about color, predecessor status and successor status.
 *
 * <p>Note that since the class is recursive, it can be
 * considered equivalently a tree.
 */

private static final class Entry KEY_VALUE_GENERIC extends ABSTRACT_MAP.BasicEntry
KEY_VALUE_GENERIC implements Cloneable {
    /** The the bit in this mask is true, the node is black. */
    private static final int BLACK_MASK = 1;
    /** If the bit in this mask is true, {@link #right} points to a successor. */
    private static final int SUCC_MASK = 1 << 31;
    /** If the bit in this mask is true, {@link #left} points to a predecessor. */
    private static final int PRED_MASK = 1 << 30;
    /** The pointers to the left and right subtrees. */
    Entry KEY_VALUE_GENERIC left, right;
    /** This integers holds different information in different bits (see {@link #SUCC_MASK} and {@link
    #PRED_MASK}). */
    int info;

    Entry() {
        super(KEY_NULL, VALUE_NULL);
    }

    /** Creates a new entry with the given key and value.
     *
     * @param k a key.
     * @param v a value.
     */
    Entry(final KEY_GENERIC_TYPE k, final VALUE_GENERIC_TYPE v) {
        super(k, v);
        info = SUCC_MASK | PRED_MASK;
    }

    /** Returns the left subtree.
     *
     * @return the left subtree ({@code null} if the left
     * subtree is empty).
     */
    Entry KEY_VALUE_GENERIC left() {
        return (info & PRED_MASK) != 0 ? null : left;
    }
}

```

```

/** Returns the right subtree.
 *
 * @return the right subtree ({ @code null} if the right
 * subtree is empty).
 */
Entry KEY_VALUE_GENERIC right() {
    return (info & SUCC_MASK) != 0 ? null : right;
}

/** Checks whether the left pointer is really a predecessor.
 * @return true if the left pointer is a predecessor.
 */
boolean pred() {
    return (info & PRED_MASK) != 0;
}

/** Checks whether the right pointer is really a successor.
 * @return true if the right pointer is a successor.
 */
boolean succ() {
    return (info & SUCC_MASK) != 0;
}

/** Sets whether the left pointer is really a predecessor.
 * @param pred if true then the left pointer will be considered a predecessor.
 */
void pred(final boolean pred) {
    if (pred) info |= PRED_MASK;
    else info &= ~PRED_MASK;
}

/** Sets whether the right pointer is really a successor.
 * @param succ if true then the right pointer will be considered a successor.
 */
void succ(final boolean succ) {
    if (succ) info |= SUCC_MASK;
    else info &= ~SUCC_MASK;
}

/** Sets the left pointer to a predecessor.
 * @param pred the predecessr.
 */
void pred(final Entry KEY_VALUE_GENERIC pred) {
    info |= PRED_MASK;
    left = pred;
}

```

```

/** Sets the right pointer to a successor.
 * @param succ the successor.
 */
void succ(final Entry KEY_VALUE_GENERIC succ) {
    info |= SUCC_MASK;
    right = succ;
}

/** Sets the left pointer to the given subtree.
 * @param left the new left subtree.
 */
void left(final Entry KEY_VALUE_GENERIC left) {
    info &= ~PRED_MASK;
    this.left = left;
}

/** Sets the right pointer to the given subtree.
 * @param right the new right subtree.
 */
void right(final Entry KEY_VALUE_GENERIC right) {
    info &= ~SUCC_MASK;
    this.right = right;
}

/** Returns whether this node is black.
 * @return true iff this node is black.
 */
boolean black() {
    return (info & BLACK_MASK) != 0;
}

/** Sets whether this node is black.
 * @param black if true, then this node becomes black; otherwise, it becomes red.
 */
void black(final boolean black) {
    if (black) info |= BLACK_MASK;
    else info &= ~BLACK_MASK;
}

/** Computes the next entry in the set order.
 *
 * @return the next entry ({ @code null}) if this is the last entry).
 */
Entry KEY_VALUE_GENERIC next() {
    Entry KEY_VALUE_GENERIC next = this.right;
    if ((info & SUCC_MASK) == 0) while ((next.info & PRED_MASK) == 0) next = next.left;
}

```

```

return next;
}

/** Computes the previous entry in the set order.
 *
 * @return the previous entry ({@code null}) if this is the first entry).
 */

Entry KEY_VALUE_GENERIC prev() {
    Entry KEY_VALUE_GENERIC prev = this.left;
    if ((info & PRED_MASK) == 0) while ((prev.info & SUCC_MASK) == 0) prev = prev.right;
    return prev;
}

@Override
public VALUE_GENERIC_TYPE setValue(final VALUE_GENERIC_TYPE value) {
    final VALUE_GENERIC_TYPE oldValue = this.value;
    this.value = value;
    return oldValue;
}

@Override
SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
public Entry KEY_VALUE_GENERIC clone() {
    Entry KEY_VALUE_GENERIC c;
    try {
        c = (Entry KEY_VALUE_GENERIC)super.clone();
    }
    catch(CloneNotSupportedException cantHappen) {
        throw new InternalError();
    }

    c.key = key;
    c.value = value;
    c.info = info;

    return c;
}

@Override
@Override
@SuppressWarnings("unchecked")
public boolean equals(final Object o) {
    if (!(o instanceof Map.Entry)) return false;
    Map.Entry <KEY_GENERIC_CLASS, VALUE_GENERIC_CLASS> e = (Map.Entry <KEY_GENERIC_CLASS,
    VALUE_GENERIC_CLASS>)o;

    return KEY_EQUALS(key, KEY_CLASS2TYPE(e.getKey())) && VALUE_EQUALS(value,
    VALUE_CLASS2TYPE(e.getValue()));
}

```

```

}

@Override
public int hashCode() {
    return KEY2JAVAHASH_NOT_NULL(key) ^ VALUE2JAVAHASH(value);
}

@Override
public String toString() {
    return key + "=>" + value;
}

/*
    public void prettyPrint() {
        prettyPrint(0);
    }

    public void prettyPrint(int level) {
        if (pred()) {
            for (int i = 0; i < level; i++)
                System.err.print(" ");
            System.err.println("pred: " + left);
        }
        else if (left != null)
            left.prettyPrint(level + 1);
        for (int i = 0; i < level; i++)
            System.err.print(" ");
        System.err.println(key + "=" + value + " (" + balance() + ")");
        if (succ()) {
            for (int i = 0; i < level; i++)
                System.err.print(" ");
            System.err.println("succ: " + right);
        }
        else if (right != null)
            right.prettyPrint(level + 1);
    }
}

/*
    public void prettyPrint() {
        System.err.println("size: " + count);
        if (tree != null) tree.prettyPrint();
    }
}

SUPPRESS_WARNINGS_KEY_UNCHECKED
@Override
public boolean containsKey(final KEY_TYPE k) {

```

```
return findKey(KEY_GENERIC_CAST k) != null;
}
```

```
@Override
public int size() {
    return count;
}
```

```
@Override
public boolean isEmpty() {
    return count == 0;
}
```

```
SUPPRESS_WARNINGS_KEY_UNCHECKED
```

```
@Override
public VALUE_GENERIC_TYPE GET_VALUE(final KEY_TYPE k) {
    final Entry KEY_VALUE_GENERIC e = findKey(KEY_GENERIC_CAST k);
    return e == null ? defRetValue : e.value;
}
```

```
@Override
public KEY_GENERIC_TYPE FIRST_KEY() {
    if (tree == null) throw new NoSuchElementException();
    return firstEntry.key;
}
```

```
@Override
public KEY_GENERIC_TYPE LAST_KEY() {
    if (tree == null) throw new NoSuchElementException();
    return lastEntry.key;
}
```

```
/** An abstract iterator on the whole range.
 *
 * <p>This class can iterate in both directions on a threaded tree.
 */
```

```
private class TreeIterator {
    /** The entry that will be returned by the next call to {@link java.util.ListIterator#previous()} (or {@code null} if
    no previous entry exists). */
    Entry KEY_VALUE_GENERIC prev;
    /** The entry that will be returned by the next call to {@link java.util.ListIterator#next()} (or {@code null} if no
    next entry exists). */
    Entry KEY_VALUE_GENERIC next;
    /** The last entry that was returned (or {@code null} if we did not iterate or used {@link #remove()}). */
    Entry KEY_VALUE_GENERIC curr;
    /** The current index (in the sense of a {@link java.util.ListIterator}). Note that this value is not meaningful when
```

this {@link TreeIterator} has been created using the nonempty constructor.*/

```
int index = 0;
```

```
TreeIterator() {  
    next = firstEntry;  
}
```

```
TreeIterator(final KEY_GENERIC_TYPE k) {  
    if ((next = locateKey(k)) != null) {  
        if (compare(next.key, k) <= 0) {  
            prev = next;  
            next = next.next();  
        }  
        else prev = next.prev();  
    }  
}
```

```
public boolean hasNext() { return next != null; }  
public boolean hasPrevious() { return prev != null; }
```

```
void updateNext() {  
    next = next.next();  
}
```

```
Entry KEY_VALUE_GENERIC nextEntry() {  
    if (! hasNext()) throw new NoSuchElementException();  
    curr = prev = next;  
    index++;  
    updateNext();  
    return curr;  
}
```

```
void updatePrevious() {  
    prev = prev.prev();  
}
```

```
Entry KEY_VALUE_GENERIC previousEntry() {  
    if (! hasPrevious()) throw new NoSuchElementException();  
    curr = next = prev;  
    index--;  
    updatePrevious();  
    return curr;  
}
```

```
public int nextIndex() {  
    return index;  
}
```

```

public int previousIndex() {
    return index - 1;
}

public void remove() {
    if (curr == null) throw new IllegalStateException();
    /* If the last operation was a next(), we are removing an entry that precedes
       the current index, and thus we must decrement it. */
    if (curr == prev) index--;
    next = prev = curr;
    updatePrevious();
    updateNext();
    RB_TREE_MAP.this.REMOVE_VALUE(curr.key);
    curr = null;
}

public int skip(final int n) {
    int i = n;
    while(i-- != 0 && hasNext()) nextEntry();
    return n - i - 1;
}

public int back(final int n) {
    int i = n;
    while(i-- != 0 && hasPrevious()) previousEntry();
    return n - i - 1;
}
}

/** An iterator on the whole range.
 *
 * <p>This class can iterate in both directions on a threaded tree.
 */

private class EntryIterator extends TreeIterator implements ObjectListIterator<MAP.Entry
KEY_VALUE_GENERIC> {
    EntryIterator() {}

    EntryIterator(final KEY_GENERIC_TYPE k) {
        super(k);
    }

    @Override
    public MAP.Entry KEY_VALUE_GENERIC next() { return nextEntry(); }
    @Override
    public MAP.Entry KEY_VALUE_GENERIC previous() { return previousEntry(); }
}

```

```

@Override
public ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> ENTRYSET() {
    if (entries == null) entries = new AbstractObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC>() {
        final Comparator<? super MAP.Entry KEY_VALUE_GENERIC> comparator = (Comparator<MAP.Entry
KEY_VALUE_GENERIC>) (x, y) -> RB_TREE_MAP.this.actualComparator.compare(x.ENTRY_GET_KEY(),
y.ENTRY_GET_KEY());

@Override
public Comparator<? super MAP.Entry KEY_VALUE_GENERIC> comparator() { return comparator; }

@Override
public ObjectBidirectionalIterator<MAP.Entry KEY_VALUE_GENERIC> iterator() { return new EntryIterator(); }

@Override
public ObjectBidirectionalIterator<MAP.Entry KEY_VALUE_GENERIC> iterator(final MAP.Entry
KEY_VALUE_GENERIC from) { return new EntryIterator(from.ENTRY_GET_KEY()); }

@Override
SUPPRESS_WARNINGS_KEY_UNCHECKED
public boolean contains(final Object o) {
    if (!(o instanceof Map.Entry)) return false;
    final Map.Entry<?,?> e = (Map.Entry<?,?>)o;
#if KEYS_PRIMITIVE
    if (e.getKey() == null || !(e.getKey() instanceof KEY_CLASS)) return false;
#endif
#if VALUES_PRIMITIVE
    if (e.getValue() == null || !(e.getValue() instanceof VALUE_CLASS)) return false;
#endif
    final Entry KEY_VALUE_GENERIC f = findKey(KEY_OBJ2TYPE(KEY_GENERIC_CAST e.getKey()));
    return e.equals(f);
}

@Override
SUPPRESS_WARNINGS_KEY_UNCHECKED
public boolean remove(final Object o) {
    if (!(o instanceof Map.Entry)) return false;
    final Map.Entry<?,?> e = (Map.Entry<?,?>)o;
#if KEYS_PRIMITIVE
    if (e.getKey() == null || !(e.getKey() instanceof KEY_CLASS)) return false;
#endif
#if VALUES_PRIMITIVE
    if (e.getValue() == null || !(e.getValue() instanceof VALUE_CLASS)) return false;
#endif
    final Entry KEY_VALUE_GENERIC f = findKey(KEY_OBJ2TYPE(KEY_GENERIC_CAST e.getKey()));
    if (f == null || !VALUE_EQUALS(f.ENTRY_GET_VALUE(), VALUE_OBJ2TYPE(e.getValue()))) return false;
    RB_TREE_MAP.this.REMOVE_VALUE(f.key);
}

```

```

    return true;
}

@Override
public int size() { return count; }

@Override
public void clear() { RB_TREE_MAP.this.clear(); }

@Override
public MAP.Entry KEY_VALUE_GENERIC first() { return firstEntry; }

@Override
public MAP.Entry KEY_VALUE_GENERIC last() { return lastEntry; }

@Override
public ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> subSet(MAP.Entry KEY_VALUE_GENERIC
from, MAP.Entry KEY_VALUE_GENERIC to) { return subMap(from.ENTRY_GET_KEY(),
to.ENTRY_GET_KEY()).ENTRYSET(); }

@Override
public ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> headSet(MAP.Entry KEY_VALUE_GENERIC
to) { return headMap(to.ENTRY_GET_KEY()).ENTRYSET(); }

@Override
public ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> tailSet(MAP.Entry KEY_VALUE_GENERIC
from) { return tailMap(from.ENTRY_GET_KEY()).ENTRYSET(); }
};

return entries;
}

/** An iterator on the whole range of keys.
 *
 * <p>This class can iterate in both directions on the keys of a threaded tree. We
 * simply override the {@link java.util.ListIterator#next()}/{@link java.util.ListIterator#previous()} methods (and
 possibly
 * their type-specific counterparts) so that they return keys instead of entries.
 */
private final class KeyIterator extends TreeIterator implements KEY_LIST_ITERATOR KEY_GENERIC {
    public KeyIterator() {}
    public KeyIterator(final KEY_GENERIC_TYPE k) { super(k); }

    @Override
    public KEY_GENERIC_TYPE NEXT_KEY() { return nextEntry().key; }

    @Override
    public KEY_GENERIC_TYPE PREV_KEY() { return previousEntry().key; }
}

```

```

};

/** A keyset implementation using a more direct implementation for iterators. */
private class KeySet extends ABSTRACT_SORTED_MAP KEY_VALUE_GENERIC.KeySet {
    @Override
    public KEY_BIDI_ITERATOR KEY_GENERIC iterator() { return new KeyIterator(); }
    @Override
    public KEY_BIDI_ITERATOR KEY_GENERIC iterator(final KEY_GENERIC_TYPE from) { return new
KeyIterator(from); }
}

/** Returns a type-specific sorted set view of the keys contained in this map.
 *
 * <p>In addition to the semantics of { @link java.util.Map#keySet() }, you can
 * safely cast the set returned by this call to a type-specific sorted
 * set interface.
 *
 * @return a type-specific sorted set view of the keys contained in this map.
 */
@Override
public SORTED_SET KEY_GENERIC keySet() {
    if (keys == null) keys = new KeySet();
    return keys;
}

/** An iterator on the whole range of values.
 *
 * <p>This class can iterate in both directions on the values of a threaded tree. We
 * simply override the { @link java.util.ListIterator#next()}/{ @link java.util.ListIterator#previous() } methods (and
possibly
 * their type-specific counterparts) so that they return values instead of entries.
 */
private final class ValueIterator extends TreeIterator implements VALUE_LIST_ITERATOR VALUE_GENERIC {
    @Override
    public VALUE_GENERIC_TYPE NEXT_VALUE() { return nextEntry().value; }

    @Override
    public VALUE_GENERIC_TYPE PREV_VALUE() { return previousEntry().value; }
};

/** Returns a type-specific collection view of the values contained in this map.
 *
 * <p>In addition to the semantics of { @link java.util.Map#values() }, you can
 * safely cast the collection returned by this call to a type-specific collection
 * interface.
 *
 * @return a type-specific collection view of the values contained in this map.

```

```

*/
@Override
public VALUE_COLLECTION VALUE_GENERIC values() {
    if (values == null) values = new VALUE_ABSTRACT_COLLECTION VALUE_GENERIC() {
        @Override
        public VALUE_ITERATOR VALUE_GENERIC iterator() { return new ValueIterator(); }
        @Override
        public boolean contains(final VALUE_TYPE k) { return containsValue(k); }
        @Override
        public int size() { return count; }
        @Override
        public void clear() { RB_TREE_MAP.this.clear(); }
    };

    return values;
}

@Override
public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() { return actualComparator; }

@Override
public SORTED_MAP KEY_VALUE_GENERIC headMap(KEY_GENERIC_TYPE to) { return new
Submap(KEY_NULL, true, to, false); }

@Override
public SORTED_MAP KEY_VALUE_GENERIC tailMap(KEY_GENERIC_TYPE from) { return new
Submap(from, false, KEY_NULL, true); }

@Override
public SORTED_MAP KEY_VALUE_GENERIC subMap(KEY_GENERIC_TYPE from, KEY_GENERIC_TYPE
to) { return new Submap(from, false, to, false); }

/** A submap with given range.
 *
 * <p>This class represents a submap. One has to specify the left/right
 * limits (which can be set to  $-\infty$ ; or  $\infty$ ). Since the submap is a
 * view on the map, at a given moment it could happen that the limits of
 * the range are not any longer in the main map. Thus, things such as
 * {@link java.util.SortedMap#firstKey()} or {@link java.util.Collection#size()} must be always computed
 * on-the-fly.
 */
private final class Submap extends ABSTRACT_SORTED_MAP KEY_VALUE_GENERIC implements
java.io.Serializable {
    private static final long serialVersionUID = -7046029254386353129L;

    /** The start of the submap range, unless {@link #bottom} is true. */
    KEY_GENERIC_TYPE from;
    /** The end of the submap range, unless {@link #top} is true. */

```

```

KEY_GENERIC_TYPE to;
/** If true, the submap range starts from -&infin;. */
boolean bottom;
/** If true, the submap range goes to &infin;. */
boolean top;
/** Cached set of entries. */
protected transient ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> entries;
/** Cached set of keys. */
protected transient SORTED_SET KEY_GENERIC keys;
/** Cached collection of values. */
protected transient VALUE_COLLECTION VALUE_GENERIC values;

/** Creates a new submap with given key range.
 *
 * @param from from the start of the submap range.
 * @param bottom if true, the first parameter is ignored and the range starts from -&infin;.
 * @param to to the end of the submap range.
 * @param top if true, the third parameter is ignored and the range goes to &infin;.
 */
public Submap(final KEY_GENERIC_TYPE from, final boolean bottom, final KEY_GENERIC_TYPE to, final
boolean top) {
    if (! bottom && ! top && RB_TREE_MAP.this.compare(from, to) > 0) throw new
IllegalArgumentException("Start key (" + from + ") is larger than end key (" + to + ")");

    this.from = from;
    this.bottom = bottom;
    this.to = to;
    this.top = top;
    this.defRetValue = RB_TREE_MAP.this.defRetValue;
}

@Override
public void clear() {
    final SubmapIterator i = new SubmapIterator();
    while(i.hasNext()) {
        i.nextEntry();
        i.remove();
    }
}

/** Checks whether a key is in the submap range.
 * @param k a key.
 * @return true if is the key is in the submap range.
 */
final boolean in(final KEY_GENERIC_TYPE k) {
    return (bottom || RB_TREE_MAP.this.compare(k, from) >= 0) &&
        (top || RB_TREE_MAP.this.compare(k, to) < 0);
}

```

```

@Override
public ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> ENTRYSET() {
    if (entries == null) entries = new AbstractObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC>() {
        @Override
        public ObjectBidirectionalIterator<MAP.Entry KEY_VALUE_GENERIC> iterator() {
            return new SubmapEntryIterator();
        }
    };

    @Override
    public ObjectBidirectionalIterator<MAP.Entry KEY_VALUE_GENERIC> iterator(final MAP.Entry
    KEY_VALUE_GENERIC from) {
        return new SubmapEntryIterator(from.ENTRY_GET_KEY());
    }

    @Override
    public Comparator<? super MAP.Entry KEY_VALUE_GENERIC> comparator() { return
    RB_TREE_MAP.this.ENTRYSET().comparator(); }

    @Override
    SUPPRESS_WARNINGS_KEY_UNCHECKED
    public boolean contains(final Object o) {
        if (!(o instanceof Map.Entry)) return false;
        final Map.Entry<?,?> e = (Map.Entry<?,?>)o;
    #if KEYS_PRIMITIVE
        if (e.getKey() == null || !(e.getKey() instanceof KEY_CLASS)) return false;
    #endif
    #if VALUES_PRIMITIVE
        if (e.getValue() == null || !(e.getValue() instanceof VALUE_CLASS)) return false;
    #endif
        final RB_TREE_MAP.Entry KEY_VALUE_GENERIC f = findKey(KEY_OBJ2TYPE(KEY_GENERIC_CAST
    e.getKey()));
        return f != null && in(f.key) && e.equals(f);
    }

    @Override
    SUPPRESS_WARNINGS_KEY_UNCHECKED
    public boolean remove(final Object o) {
        if (!(o instanceof Map.Entry)) return false;
        final Map.Entry<?,?> e = (Map.Entry<?,?>)o;
    #if KEYS_PRIMITIVE
        if (e.getKey() == null || !(e.getKey() instanceof KEY_CLASS)) return false;
    #endif
    #if VALUES_PRIMITIVE
        if (e.getValue() == null || !(e.getValue() instanceof VALUE_CLASS)) return false;
    #endif
        final RB_TREE_MAP.Entry KEY_VALUE_GENERIC f = findKey(KEY_OBJ2TYPE(KEY_GENERIC_CAST
    e.getKey()));

```

```

    if (f != null && in(f.key)) Submap.this.REMOVE_VALUE(f.key);
    return f != null;
}

@Override
public int size() {
    int c = 0;
    for(Iterator<?> i = iterator(); i.hasNext(); i.next()) c++;
    return c;
}

@Override
public boolean isEmpty() { return ! new SubmapIterator().hasNext(); }

@Override
public void clear() { Submap.this.clear(); }

@Override
public MAP.Entry KEY_VALUE_GENERIC first() { return firstEntry(); }

@Override
public MAP.Entry KEY_VALUE_GENERIC last() { return lastEntry(); }

@Override
public ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> subSet(MAP.Entry KEY_VALUE_GENERIC
from, MAP.Entry KEY_VALUE_GENERIC to) { return subMap(from.ENTRY_GET_KEY(),
to.ENTRY_GET_KEY()).ENTRYSET(); }

@Override
public ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> headSet(MAP.Entry KEY_VALUE_GENERIC
to) { return headMap(to.ENTRY_GET_KEY()).ENTRYSET(); }

@Override
public ObjectSortedSet<MAP.Entry KEY_VALUE_GENERIC> tailSet(MAP.Entry KEY_VALUE_GENERIC
from) { return tailMap(from.ENTRY_GET_KEY()).ENTRYSET(); }
};

return entries;
}

private class KeySet extends ABSTRACT_SORTED_MAP KEY_VALUE_GENERIC.KeySet {
    @Override
    public KEY_BIDI_ITERATOR KEY_GENERIC iterator() { return new SubmapKeyIterator(); }
    @Override
    public KEY_BIDI_ITERATOR KEY_GENERIC iterator(final KEY_GENERIC_TYPE from) { return new
SubmapKeyIterator(from); }
}

```

```

@Override
public SORTED_SET KEY_GENERIC keySet() {
    if (keys == null) keys = new KeySet();
    return keys;
}

@Override
public VALUE_COLLECTION VALUE_GENERIC values() {
    if (values == null) values = new VALUE_ABSTRACT_COLLECTION VALUE_GENERIC() {
        @Override
        public VALUE_ITERATOR VALUE_GENERIC iterator() { return new SubmapValueIterator(); }
        @Override
        public boolean contains(final VALUE_TYPE k) { return containsValue(k); }
        @Override
        public int size() { return Submap.this.size(); }
        @Override
        public void clear() { Submap.this.clear(); }
    };

    return values;
}

@Override
SUPPRESS_WARNINGS_KEY_UNCHECKED
public boolean containsKey(final KEY_TYPE k) { return in(KEY_GENERIC_CAST k) &&
RB_TREE_MAP.this.containsKey(k); }

@Override
public boolean containsValue(final VALUE_TYPE v) {
    final SubmapIterator i = new SubmapIterator();
    VALUE_TYPE ev;

    while(i.hasNext()) {
        ev = i.nextEntry().value;
        if (VALUE_EQUALS(ev, v)) return true;
    }

    return false;
}

@Override
SUPPRESS_WARNINGS_KEY_UNCHECKED
public VALUE_GENERIC_TYPE GET_VALUE(final KEY_TYPE k) {
    final RB_TREE_MAP.Entry KEY_VALUE_GENERIC e;
    final KEY_GENERIC_TYPE kk = KEY_GENERIC_CAST k;
    return in(kk) && (e = findKey(kk)) != null ? e.value : this.defRetVal;
}

```

```

@Override
public VALUE_GENERIC_TYPE put(final KEY_GENERIC_TYPE k, final VALUE_GENERIC_TYPE v) {
    modified = false;
    if (! in(k)) throw new IllegalArgumentException("Key (" + k + ") out of range [" + (bottom ? "-" :
String.valueOf(from)) + ", " + (top ? "-" : String.valueOf(to)) + ")");
    final VALUE_GENERIC_TYPE oldValue = RB_TREE_MAP.this.put(k, v);
    return modified ? this.defRetValue : oldValue;
}

```

```

@Override
SUPPRESS_WARNINGS_KEY_UNCHECKED
public VALUE_GENERIC_TYPE REMOVE_VALUE(final KEY_TYPE k) {
    modified = false;
    if (! in(KEY_GENERIC_CAST k)) return this.defRetValue;
    final VALUE_GENERIC_TYPE oldValue = RB_TREE_MAP.this.REMOVE_VALUE(k);
    return modified ? oldValue : this.defRetValue;
}

```

```

@Override
public int size() {
    final SubmapIterator i = new SubmapIterator();
    int n = 0;

    while(i.hasNext()) {
        n++;
        i.nextEntry();
    }

    return n;
}

```

```

@Override
public boolean isEmpty() { return ! new SubmapIterator().hasNext(); }

```

```

@Override
public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() { return actualComparator; }

```

```

@Override
public SORTED_MAP KEY_VALUE_GENERIC headMap(final KEY_GENERIC_TYPE to) {
    if (top) return new Submap(from, bottom, to, false);
    return compare(to, this.to) < 0 ? new Submap(from, bottom, to, false) : this;
}

```

```

@Override
public SORTED_MAP KEY_VALUE_GENERIC tailMap(final KEY_GENERIC_TYPE from) {
    if (bottom) return new Submap(from, false, to, top);
    return compare(from, this.from) > 0 ? new Submap(from, false, to, top) : this;
}

```

```

@Override
public SORTED_MAP KEY_VALUE_GENERIC subMap(KEY_GENERIC_TYPE from, KEY_GENERIC_TYPE
to) {
    if (top && bottom) return new Submap(from, false, to, false);
    if (! top) to = compare(to, this.to) < 0 ? to : this.to;
    if (! bottom) from = compare(from, this.from) > 0 ? from : this.from;
    if (! top && ! bottom && from == this.from && to == this.to) return this;
    return new Submap(from, false, to, false);
}

/** Locates the first entry.
 *
 * @return the first entry of this submap, or { @code null } if the submap is empty.
 */
public RB_TREE_MAP.Entry KEY_VALUE_GENERIC firstEntry() {
    if (tree == null) return null;
    // If this submap goes to -infinity, we return the main map first entry; otherwise, we locate the start of the map.
    RB_TREE_MAP.Entry KEY_VALUE_GENERIC e;
    if (bottom) e = firstEntry();
    else {
        e = locateKey(from);
        // If we find either the start or something greater we're OK.
        if (compare(e.key, from) < 0) e = e.next();
    }
    // Finally, if this submap doesn't go to infinity, we check that the resulting key isn't greater than the end.
    if (e == null || ! top && compare(e.key, to) >= 0) return null;
    return e;
}

/** Locates the last entry.
 *
 * @return the last entry of this submap, or { @code null } if the submap is empty.
 */
public RB_TREE_MAP.Entry KEY_VALUE_GENERIC lastEntry() {
    if (tree == null) return null;
    // If this submap goes to infinity, we return the main map last entry; otherwise, we locate the end of the map.
    RB_TREE_MAP.Entry KEY_VALUE_GENERIC e;
    if (top) e = lastEntry();
    else {
        e = locateKey(to);
        // If we find something smaller than the end we're OK.
        if (compare(e.key, to) >= 0) e = e.prev();
    }
    // Finally, if this submap doesn't go to -infinity, we check that the resulting key isn't smaller than the start.
    if (e == null || ! bottom && compare(e.key, from) < 0) return null;
    return e;
}

```

```

@Override
public KEY_GENERIC_TYPE FIRST_KEY() {
    RB_TREE_MAP.Entry KEY_VALUE_GENERIC e = firstEntry();
    if (e == null) throw new NoSuchElementException();
    return e.key;
}

@Override
public KEY_GENERIC_TYPE LAST_KEY() {
    RB_TREE_MAP.Entry KEY_VALUE_GENERIC e = lastEntry();
    if (e == null) throw new NoSuchElementException();
    return e.key;
}

/** An iterator for subranges.
 *
 * <p>This class inherits from { @link TreeIterator}, but overrides the methods that
 * update the pointer after a { @link java.util.ListIterator#next()} or { @link java.util.ListIterator#previous()}. If we
 would
 * move out of the range of the submap we just overwrite the next or previous
 * entry with { @code null}.
 */
private class SubmapIterator extends TreeIterator {
    SubmapIterator() {
        next = firstEntry();
    }

    SubmapIterator(final KEY_GENERIC_TYPE k) {
        this();

        if (next != null) {
            if (! bottom && compare(k, next.key) < 0) prev = null;
            else if (! top && compare(k, (prev = lastEntry()).key) >= 0) next = null;
            else {
                next = locateKey(k);

                if (compare(next.key, k) <= 0) {
                    prev = next;
                    next = next.next();
                }
                else prev = next.prev();
            }
        }
    }

    @Override
    void updatePrevious() {

```

```

    prev = prev.prev();
    if (! bottom && prev != null && RB_TREE_MAP.this.compare(prev.key, from) < 0) prev = null;
}

@Override
void updateNext() {
    next = next.next();
    if (! top && next != null && RB_TREE_MAP.this.compare(next.key, to) >= 0) next = null;
}
}

private class SubmapEntryIterator extends SubmapIterator implements ObjectListIterator<MAP.Entry
KEY_VALUE_GENERIC> {
    SubmapEntryIterator() {}

    SubmapEntryIterator(final KEY_GENERIC_TYPE k) {
        super(k);
    }

    @Override
    public MAP.Entry KEY_VALUE_GENERIC next() { return nextEntry(); }
    @Override
    public MAP.Entry KEY_VALUE_GENERIC previous() { return previousEntry(); }
}

/** An iterator on a subrange of keys.
 *
 * <p>This class can iterate in both directions on a subrange of the
 * keys of a threaded tree. We simply override the {@link
 * java.util.ListIterator#next()}/{@link java.util.ListIterator#previous()} methods (and possibly their
 * type-specific counterparts) so that they return keys instead of
 * entries.
 */
private final class SubmapKeyIterator extends SubmapIterator implements KEY_LIST_ITERATOR
KEY_GENERIC {
    public SubmapKeyIterator() { super(); }
    public SubmapKeyIterator(KEY_GENERIC_TYPE from) { super(from); }

    @Override
    public KEY_GENERIC_TYPE NEXT_KEY() { return nextEntry().key; }
    @Override
    public KEY_GENERIC_TYPE PREV_KEY() { return previousEntry().key; }
};

/** An iterator on a subrange of values.
 *
 * <p>This class can iterate in both directions on the values of a

```

```

* subrange of the keys of a threaded tree. We simply override the
* {@link java.util.ListIterator#next()}/{@link java.util.ListIterator#previous()} methods (and possibly their
* type-specific counterparts) so that they return values instead of
* entries.
*/
private final class SubmapValueIterator extends SubmapIterator implements VALUE_LIST_ITERATOR
VALUE_GENERIC {
    @Override
    public VALUE_GENERIC_TYPE NEXT_VALUE() { return nextEntry().value; }
    @Override
    public VALUE_GENERIC_TYPE PREV_VALUE() { return previousEntry().value; }
};
}

/** Returns a deep copy of this tree map.
*
* <p>This method performs a deep copy of this tree map; the data stored in the
* set, however, is not cloned. Note that this makes a difference only for object keys.
*
* @return a deep copy of this tree map.
*/
@Override
SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
public RB_TREE_MAP KEY_VALUE_GENERIC clone() {
    RB_TREE_MAP KEY_VALUE_GENERIC c;
    try {
        c = (RB_TREE_MAP KEY_VALUE_GENERIC)super.clone();
    }
    catch(CloneNotSupportedException cantHappen) {
        throw new InternalError();
    }

    c.keys = null;
    c.values = null;
    c.entries = null;
    c.allocatePaths();

    if (count != 0) {
        // Also this apparently unfathomable code is derived from GNU libavl.
        Entry KEY_VALUE_GENERIC e, p, q, rp = new Entry KEY_VALUE_GENERIC_DIAMOND(), rq = new Entry
KEY_VALUE_GENERIC_DIAMOND();

        p = rp;
        rp.left(tree);

        q = rq;
        rq.pred(null);
    }
}

```

```

while(true) {
    if (! p.pred()) {
        e = p.left.clone();
        e.pred(q.left);
        e.succ(q);
        q.left(e);

        p = p.left;
        q = q.left;
    }
    else {
        while(p.succ()) {
            p = p.right;

            if (p == null) {
                q.right = null;
                c.tree = rq.left;

                c.firstEntry = c.tree;
                while(c.firstEntry.left != null) c.firstEntry = c.firstEntry.left;
                c.lastEntry = c.tree;
                while(c.lastEntry.right != null) c.lastEntry = c.lastEntry.right;

                return c;
            }
            q = q.right;
        }

        p = p.right;
        q = q.right;
    }

    if (! p.succ()) {
        e = p.right.clone();
        e.succ(q.right);
        e.pred(q);
        q.right(e);
    }
}

return c;
}

private void writeObject(java.io.ObjectOutputStream s) throws java.io.IOException {
    int n = count;

```

```

EntryIterator i = new EntryIterator();
Entry KEY_VALUE_GENERIC e;

s.defaultWriteObject();

while(n-- != 0) {
    e = i.nextEntry();
    s.WRITE_KEY(e.key);
    s.WRITE_VALUE(e.value);
}
}

/** Reads the given number of entries from the input stream, returning the corresponding tree.
 *
 * @param s the input stream.
 * @param n the (positive) number of entries to read.
 * @param pred the entry containing the key that preceeds the first key in the tree.
 * @param succ the entry containing the key that follows the last key in the tree.
 */
SUPPRESS_WARNINGS_KEY_VALUE_UNCHECKED
private Entry KEY_VALUE_GENERIC readTree(final java.io.ObjectInputStream s, final int n, final Entry
KEY_VALUE_GENERIC pred, final Entry KEY_VALUE_GENERIC succ) throws java.io.IOException,
ClassNotFoundException {
    if (n == 1) {
        final Entry KEY_VALUE_GENERIC top = new Entry
KEY_VALUE_GENERIC_DIAMOND(KEY_GENERIC_CAST s.READ_KEY(), VALUE_GENERIC_CAST
s.READ_VALUE());
        top.pred(pred);
        top.succ(succ);
        top.black(true);

        return top;
    }

    if (n == 2) {
        /* We handle separately this case so that recursion will
        *always* be on nonempty subtrees. */
        final Entry KEY_VALUE_GENERIC top = new Entry
KEY_VALUE_GENERIC_DIAMOND(KEY_GENERIC_CAST s.READ_KEY(), VALUE_GENERIC_CAST
s.READ_VALUE());
        top.black(true);
        top.right(new Entry KEY_VALUE_GENERIC_DIAMOND(KEY_GENERIC_CAST s.READ_KEY(),
VALUE_GENERIC_CAST s.READ_VALUE()));
        top.right.pred(top);
        top.pred(pred);
        top.right.succ(succ);
    }
}

```

```

return top;
}

// The right subtree is the largest one.
final int rightN = n / 2, leftN = n - rightN - 1;

final Entry KEY_VALUE_GENERIC top = new Entry KEY_VALUE_GENERIC_DIAMOND();

top.left(readTree(s, leftN, pred, top));

top.key = KEY_GENERIC_CAST s.READ_KEY();
top.value = VALUE_GENERIC_CAST s.READ_VALUE();
top.black(true);

top.right(readTree(s, rightN, top, succ));

if (n + 2 == ((n + 2) & -(n + 2))) top.right.black(false); // Quick test for determining whether n + 2 is a power of 2.

return top;
}

private void readObject(java.io.ObjectInputStream s) throws java.io.IOException, ClassNotFoundException {
    s.defaultReadObject();
    /* The storedComparator is now correctly set, but we must restore
       on-the-fly the actualComparator. */
    setActualComparator();
    allocatePaths();

    if (count != 0) {
        tree = readTree(s, count, null, null);
        Entry KEY_VALUE_GENERIC e;

        e = tree;
        while(e.left() != null) e = e.left();
        firstEntry = e;

        e = tree;
        while(e.right() != null) e = e.right();
        lastEntry = e;
    }
}

#ifdef ASSERTS_CODE
private void checkNodePath() {
    for(int i = nodePath.length; i-- != 0;) assert nodePath[i] == null : i;
}

private static KEY_VALUE_GENERIC int checkTree(Entry KEY_VALUE_GENERIC e, int d, int D) {

```

```

if (e == null) return 0;
if (e.black()) d++;
if (e.left() != null) D = checkTree(e.left(), d, D);
if (e.right() != null) D = checkTree(e.right(), d, D);
if (e.left() == null && e.right() == null) {
    if (D == -1) D = d;
    else if (D != d) throw new AssertionError("Mismatch between number of black nodes (" + D + " and " + d + ")");
}
return D;
}
#endif

```

```

#ifdef TEST

```

```

private static long seed = System.currentTimeMillis();
private static java.util.Random r = new java.util.Random(seed);

```

```

private static KEY_TYPE genKey() {
#ifdef KEY_CLASS_Byte || KEY_CLASS_Short || KEY_CLASS_Character
    return (KEY_TYPE)r.nextInt();
#elif KEYS_PRIMITIVE
    return r.NEXT_KEY();
#else
    return Integer.toBinaryString(r.nextInt());
#endif
}

```

```

private static VALUE_TYPE genValue() {
#ifdef VALUE_CLASS_Byte || VALUE_CLASS_Short || VALUE_CLASS_Character
    return (VALUE_TYPE)r.nextInt();
#elif VALUES_PRIMITIVE
    return r.NEXT_VALUE();
#elif !VALUE_CLASS_Reference || KEY_CLASS_Reference
    return Integer.toBinaryString(r.nextInt());
#else
    return new java.io.Serializable() {};
#endif
}

```

```

private static java.text.NumberFormat format = new java.text.DecimalFormat("#,###.00");
private static java.text.FieldPosition p = new java.text.FieldPosition(0);

```

```

private static String format(double d) {
    StringBuffer s = new StringBuffer();
    return format.format(d, s, p).toString();
}

```

```

}

private static void speedTest(int n, boolean comp) {
    int i, j;
    RB_TREE_MAP m;
    java.util.TreeMap t;
    KEY_TYPE k[] = new KEY_TYPE[n];
    KEY_TYPE nk[] = new KEY_TYPE[n];
    VALUE_TYPE v[] = new VALUE_TYPE[n];
    long ms;

    for(i = 0; i < n; i++) {
        k[i] = genKey();
        nk[i] = genKey();
        v[i] = genValue();
    }

    double totPut = 0, totYes = 0, totNo = 0, totAddTo = 0, totIterFor = 0, totIterBack = 0, totRemYes = 0, d, dd, ddd;

    if (comp) { for(j = 0; j < 20; j++) {

        t = new java.util.TreeMap();

        /* We first add all pairs to t. */
        for(i = 0; i < n; i++) t.put(KEY2OBJ(k[i]), VALUE2OBJ(v[i]));

        /* Then we remove the first half and put it back. */
        for(i = 0; i < n/2; i++) t.remove(KEY2OBJ(k[i]));

        ms = System.currentTimeMillis();
        for(i = 0; i < n/2; i++) t.put(KEY2OBJ(k[i]), VALUE2OBJ(v[i]));
        d = System.currentTimeMillis() - ms;

        /* Then we remove the other half and put it back again. */
        ms = System.currentTimeMillis();
        for(i = n/2; i < n; i++) t.remove(KEY2OBJ(k[i]));
        dd = System.currentTimeMillis() - ms ;

        ms = System.currentTimeMillis();
        for(i = n/2; i < n; i++) t.put(KEY2OBJ(k[i]), VALUE2OBJ(v[i]));
        d += System.currentTimeMillis() - ms;
        if (j > 2) totPut += n/d;
        System.out.print("Add: " + format(n/d) + " K/s ");

        /* Then we remove again the first half. */
        ms = System.currentTimeMillis();
        for(i = 0; i < n/2; i++) t.remove(KEY2OBJ(k[i]));
    }
}

```

```

dd += System.currentTimeMillis() - ms ;
if (j > 2) totRemYes += n/dd;
System.out.print("RemYes: " + format(n/dd) + " K/s ");

/* And then we put it back. */
for(i = 0; i < n/2; i++) t.put(KEY2OBJ(k[i]), VALUE2OBJ(v[i]));

#if VALUES_PRIMITIVE && !VALUE_CLASS_Boolean
/* we perform n/2 addTo() operations with get then put */
ms = System.currentTimeMillis();
for(i = 0; i < n/2; i++) t.put(KEY2OBJ(k[i]), (VALUE_TYPE)((VALUE_CLASS) t.get(KEY2OBJ(k[i])) + i));
ddd = System.currentTimeMillis() - ms;
if (j > 2) totAddTo += n/ddd;
System.out.print("AddTo: " + format(n/ddd) + " K/s ");
#endif

/* We check for pairs in t. */
ms = System.currentTimeMillis();
for(i = 0; i < n; i++) t.containsKey(KEY2OBJ(k[i]));
d = 1.0 * n / (System.currentTimeMillis() - ms);
if (j > 2) totYes += d;
System.out.print("Yes: " + format(d) + " K/s ");

/* We check for pairs not in t. */
ms = System.currentTimeMillis();
for(i = 0; i < n; i++) t.containsKey(KEY2OBJ(nk[i]));
d = 1.0 * n / (System.currentTimeMillis() - ms);
if (j > 2) totNo += d;
System.out.print("No: " + format(d) + " K/s ");

/* We iterate on t. */
ms = System.currentTimeMillis();
for(Iterator it = t.entrySet().iterator(); it.hasNext(); it.next());
d = 1.0 * n / (System.currentTimeMillis() - ms);
if (j > 2) totIterFor += d;
System.out.print("IterFor: " + format(d) + " K/s ");

System.out.println();
}

System.out.println();
System.out.println("java.util Put: " + format(totPut/(j-3)) + " K/s RemYes: " + format(totRemYes/(j-3)) + " K/s Yes:
" + format(totYes/(j-3)) + " K/s No: " + format(totNo/(j-3))+ "K/s AddTo: " + format(totAddTo/(j-3)) + " K/s
IterFor: " + format(totIterFor/(j-3)) + " K/s");

System.out.println();

t = null;

```

```
totPut = totYes = totNo = totIterFor = totIterBack = totRemYes = 0;
```

```
}
```

```
for(j = 0; j < 20; j++) {
```

```
    m = new RB_TREE_MAP();
```

```
    /* We first add all pairs to m. */
```

```
    for(i = 0; i < n; i++) m.put(k[i], v[i]);
```

```
    /* Then we remove the first half and put it back. */
```

```
    for(i = 0; i < n/2; i++) m.remove(k[i]);
```

```
    ms = System.currentTimeMillis();
```

```
    for(i = 0; i < n/2; i++) m.put(k[i], v[i]);
```

```
    d = System.currentTimeMillis() - ms;
```

```
    /* Then we remove the other half and put it back again. */
```

```
    ms = System.currentTimeMillis();
```

```
    for(i = n/2; i < n; i++) m.remove(k[i]);
```

```
    dd = System.currentTimeMillis() - ms ;
```

```
    ms = System.currentTimeMillis();
```

```
    for(i = n/2; i < n; i++) m.put(k[i], v[i]);
```

```
    d += System.currentTimeMillis() - ms;
```

```
    if (j > 2) totPut += n/d;
```

```
    System.out.print("Add: " + format(n/d) + " K/s ");
```

```
    /* Then we remove again the first half. */
```

```
    ms = System.currentTimeMillis();
```

```
    for(i = 0; i < n/2; i++) m.remove(k[i]);
```

```
    dd += System.currentTimeMillis() - ms ;
```

```
    if (j > 2) totRemYes += n/dd;
```

```
    System.out.print("RemYes: " + format(n/dd) + " K/s ");
```

```
    /* And then we put it back. */
```

```
    for(i = 0; i < n/2; i++) m.put(k[i], v[i]);
```

```
#if VALUES_PRIMITIVE && !VALUE_CLASS_Boolean
```

```
/* we perform n/2 addTo() operations with get then put */
```

```
ms = System.currentTimeMillis();
```

```
for(i = 0; i < n/2; i++) m.addTo(k[i], (VALUE_TYPE) i);
```

```
ddd = System.currentTimeMillis() - ms;
```

```
if (j > 2) totAddTo += n/ddd;
```

```
System.out.print("AddTo: " + format(n/ddd) + " K/s ");
```

```
#endif
```

```

/* We check for pairs in m. */
ms = System.currentTimeMillis();
for(i = 0; i < n; i++) m.containsKey(k[i]);
d = 1.0 * n / (System.currentTimeMillis() - ms);
if (j > 2) totYes += d;
System.out.print("Yes: " + format(d) + " K/s ");

/* We check for pairs not in m. */
ms = System.currentTimeMillis();
for(i = 0; i < n; i++) m.containsKey(nk[i]);
d = 1.0 * n / (System.currentTimeMillis() - ms);
if (j > 2) totNo += d;
System.out.print("No: " + format(d) + " K/s ");

/* We iterate on m. */
java.util.ListIterator it = (java.util.ListIterator)m.entrySet().iterator();
ms = System.currentTimeMillis();
for(; it.hasNext(); it.next());
d = 1.0 * n / (System.currentTimeMillis() - ms);
if (j > 2) totIterFor += d;
System.out.print("IterFor: " + format(d) + " K/s ");

/* We iterate back on m. */
ms = System.currentTimeMillis();
for(; it.hasPrevious(); it.previous());
d = 1.0 * n / (System.currentTimeMillis() - ms);
if (j > 2) totIterBack += d;
System.out.print("IterBack: " + format(d) + " K/s ");

System.out.println();
}

System.out.println();
System.out.println("fastutil Put: " + format(totPut/(j-3)) + " K/s RemYes: " + format(totRemYes/(j-3)) + " K/s Yes:
" + format(totYes/(j-3)) + " K/s No: " + format(totNo/(j-3))+ "K/s AddTo: " + format(totAddTo/(j-3)) + " K/s
IterFor: " + format(totIterFor/(j-3)) + " K/s");

System.out.println();

}

private static boolean valEquals(Object o1, Object o2) {
return o1 == null ? o2 == null : o1.equals(o2);
}

```

```

private static void fatal(String msg) {
    System.out.println(msg);
    System.exit(1);
}

private static void ensure(boolean cond, String msg) {
    if (cond) return;
    fatal(msg);
}

private static Object[] k, v, nk;
private static KEY_TYPE kt[];
private static KEY_TYPE nkt[];
private static VALUE_TYPE vt[];
private static RB_TREE_MAP topMap;

protected static void testMaps(SORTED_MAP m, SortedMap t, int n, int level) {
    long ms;
    boolean mThrowsIllegal, tThrowsIllegal, mThrowsNoElement, tThrowsNoElement;
    Object rt = null, rm = null;

    if (level > 4) return;

    /* Now we check that both maps agree on first/last keys. */

    mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

    try {
        m.firstKey();
    }
    catch (NoSuchElementException e) { mThrowsNoElement = true; }
    try {
        t.firstKey();
    }
    catch (NoSuchElementException e) { tThrowsNoElement = true; }

    ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): firstKey() divergence at
start in NoSuchElementException (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");
    if (! mThrowsNoElement) ensure(t.firstKey().equals(m.firstKey()), "Error (" + level + ", " + seed + "): m and t differ
at start on their first key (" + m.firstKey() + ", " + t.firstKey() + ")");

    mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

    try {
        m.lastKey();
    }
    catch (NoSuchElementException e) { mThrowsNoElement = true; }

```

```

try {
    t.lastKey();
}
catch (NoSuchElementException e) { tThrowsNoElement = true; }

ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): lastKey() divergence at start
in NoSuchElementException (" + mThrowsNoElement + ", " + tThrowsNoElement + "));

if (! mThrowsNoElement) ensure(t.lastKey().equals(m.lastKey()), "Error (" + level + ", " + seed + "): m and t differ
at start on their last key (" + m.lastKey() + ", " + t.lastKey() + "));

/* Now we check that m and t are equal. */
if (!m.equals(t) || ! t.equals(m)) System.err.println("m: " + m + " t: " + t);

ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) at start");
ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) at start");

/* Now we check that m actually holds that data. */
for(Iterator i=t.entrySet().iterator(); i.hasNext();) {
    java.util.Map.Entry e = (java.util.Map.Entry)i.next();
    ensure(valEquals(e.getValue(), m.get(e.getKey())), "Error (" + level + ", " + seed + "): m and t differ on an entry
("+e+") after insertion (iterating on t)");
}

/* Now we check that m actually holds that data, but iterating on m. */
for(Iterator i=m.entrySet().iterator(); i.hasNext();) {
    Entry e = (Entry)i.next();
    ensure(valEquals(e.getValue(), t.get(e.getKey())), "Error (" + level + ", " + seed + "): m and t differ on an entry
("+e+") after insertion (iterating on m)");
}

/* Now we check that m actually holds the same keys. */
for(Iterator i=t.keySet().iterator(); i.hasNext();) {
    Object o = i.next();
    ensure(m.containsKey(o), "Error (" + level + ", " + seed + "): m and t differ on a key (" + o + ") after insertion
(iterating on t)");
    ensure(m.keySet().contains(o), "Error (" + level + ", " + seed + "): m and t differ on a key (" + o + ", in keySet()) after
insertion (iterating on t)");
}

/* Now we check that m actually holds the same keys, but iterating on m. */
for(Iterator i=m.keySet().iterator(); i.hasNext();) {
    Object o = i.next();
    ensure(t.containsKey(o), "Error (" + level + ", " + seed + "): m and t differ on a key after insertion (iterating on m)");
}

```

```

    ensure(t.keySet().contains(o), "Error (" + level + ", " + seed + "): m and t differ on a key (in keySet()) after insertion
(iterating on m)");
}

/* Now we check that m actually hold the same values. */
for(Iterator i=t.values().iterator(); i.hasNext(); ) {
    Object o = i.next();
    ensure(m.containsValue(o), "Error (" + level + ", " + seed + "): m and t differ on a value after insertion (iterating on
t)");
    ensure(m.values().contains(o), "Error (" + level + ", " + seed + "): m and t differ on a value (in values()) after
insertion (iterating on t)");
}

/* Now we check that m actually hold the same values, but iterating on m. */
for(Iterator i=m.values().iterator(); i.hasNext(); ) {
    Object o = i.next();
    ensure(t.containsValue(o), "Error (" + level + ", " + seed + "): m and t differ on a value after insertion (iterating on
m)");
    ensure(t.values().contains(o), "Error (" + level + ", " + seed + "): m and t differ on a value (in values()) after
insertion (iterating on m)");
}

/* Now we check that inquiries about random data give the same answer in m and t. For
m we use the polymorphic method. */

for(int i=0; i<n; i++) {
    KEY_TYPE T = genKey();

    mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

    try {
        m.containsKey(KEY2OBJ(T));
    }
    catch (NoSuchElementException e) { mThrowsNoElement = true; }
    catch (IllegalArgumentException e) { mThrowsIllegal = true; }

    try {
        t.containsKey(KEY2OBJ(T));
    }
    catch (NoSuchElementException e) { tThrowsNoElement = true; }
    catch (IllegalArgumentException e) { tThrowsIllegal = true; }

    ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): containsKey() divergence in
NoSuchElementException (" + mThrowsNoElement + ", " + tThrowsNoElement + "));
    ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): containsKey() divergence in
IllegalArgumentException (" + mThrowsIllegal + ", " + tThrowsIllegal + "));
    if (!mThrowsNoElement && !mThrowsIllegal) {

```

```

    ensure(m.containsKey(KEY2OBJ(T)) == t.containsKey(KEY2OBJ(T)), "Error (" + level + ", " + seed + "):
divergence in keys between t and m (polymorphic method)");

#if KEY_CLASS_Object && ! (VALUES_REFERENCE)
    if ((m.GET_VALUE(T) != VALUE_NULL) != ((t.get(KEY2OBJ(T)) == null ? VALUE_NULL :
VALUE_OBJ2TYPE(t.get(KEY2OBJ(T)))) != VALUE_NULL) ||
        t.get(KEY2OBJ(T)) != null &&
        ! VALUE2OBJ(m.GET_VALUE(T)).equals(t.get(KEY2OBJ(T))))
#else
    if ((m.get(T) != VALUE_NULL) != ((t.get(KEY2OBJ(T)) == null ? VALUE_NULL :
VALUE_OBJ2TYPE(t.get(KEY2OBJ(T)))) != VALUE_NULL) ||
        t.get(KEY2OBJ(T)) != null &&
        ! m.get(KEY2OBJ(T)).equals(t.get(KEY2OBJ(T))))
#endif
    {
        System.out.println("Error (" + level + ", " + seed + "): divergence between t and m (polymorphic method)");
        System.exit(1);
    }
}

/* Again, we check that inquiries about random data give the same answer in m and t, but
for m we use the standard method. */

for(int i=0; i<n; i++) {
    KEY_TYPE T = genKey();

    mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

    try {
        m.get(KEY2OBJ(T));
    }
    catch (NoSuchElementException e) { mThrowsNoElement = true; }
    catch (IllegalArgumentException e) { mThrowsIllegal = true; }

    try {
        t.get(KEY2OBJ(T));
    }
    catch (NoSuchElementException e) { tThrowsNoElement = true; }
    catch (IllegalArgumentException e) { tThrowsIllegal = true; }

    ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): get() divergence in
NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");
    ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): get() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ")");
    if (!mThrowsNoElement && !mThrowsIllegal) ensure(valEquals(m.get(KEY2OBJ(T)), t.get(KEY2OBJ(T))),
"Error (" + level + ", " + seed + "): divergence between t and m (standard method)");
}

```

```
/* Now we put and remove random data in m and t, checking that the result is the same. */
```

```
for(int i=0; i<20*n; i++) {  
    KEY_TYPE T = genKey();  
    VALUE_TYPE U = genValue();
```

```
    mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;
```

```
    try {  
        rm = m.put(KEY2OBJ(T), VALUE2OBJ(U));  
    }  
    catch (NoSuchElementException e) { mThrowsNoElement = true; }  
    catch (IllegalArgumentException e) { mThrowsIllegal = true; }
```

```
    try {  
        rt = t.put(KEY2OBJ(T), VALUE2OBJ(U));  
    }  
    catch (NoSuchElementException e) { tThrowsNoElement = true; }  
    catch (IllegalArgumentException e) { tThrowsIllegal = true; }
```

```
    ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): put() divergence in  
    NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + "));  
    ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): put() divergence in  
    IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + "));  
    if (!mThrowsNoElement && !mThrowsIllegal) ensure(valEquals(rm, rt), "Error (" + level + ", " + seed + "):  
    divergence in put() between t and m (" + rt + ", " + rm + "));
```

```
    T = genKey();
```

```
    mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;
```

```
    try {  
        rm = m.remove(KEY2OBJ(T));  
    }  
    catch (NoSuchElementException e) { mThrowsNoElement = true; }  
    catch (IllegalArgumentException e) { mThrowsIllegal = true; }
```

```
    try {  
        rt = t.remove(KEY2OBJ(T));  
    }  
    catch (NoSuchElementException e) { tThrowsNoElement = true; }  
    catch (IllegalArgumentException e) { tThrowsIllegal = true; }
```

```
    ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): remove() divergence in  
    NoSuchElementException for " + T + " (" + mThrowsNoElement + ", " + tThrowsNoElement + "));
```

```

    ensure(mThrowsIllegal == tThrowsIllegal, "Error (" + level + ", " + seed + "): remove() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ")");
    if (!mThrowsNoElement && !mThrowsIllegal) ensure(valEquals(rm, rt), "Error (" + level + ", " + seed + "):
divergence in remove() between t and m (" + rt + ", " + rm + ")");
}

ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after removal");
ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after removal");

/* Now we check that m actually holds the same data. */

for(Iterator i=t.entrySet().iterator(); i.hasNext();) {
    java.util.Map.Entry e = (java.util.Map.Entry)i.next();
    ensure(valEquals(e.getValue(), m.get(e.getKey())), "Error (" + level + ", " + seed + "): m and t differ on an entry
("+e+") after removal (iterating on t)");
}

/* Now we check that m actually holds that data, but iterating on m. */

for(Iterator i=m.entrySet().iterator(); i.hasNext();) {
    Entry e = (Entry)i.next();
    ensure(valEquals(e.getValue(), t.get(e.getKey())), "Error (" + level + ", " + seed + "): m and t differ on an entry
("+e+") after removal (iterating on m)");
}

/* Now we check that m actually holds the same keys. */

for(Iterator i=t.keySet().iterator(); i.hasNext();) {
    Object o = i.next();
    ensure(m.containsKey(o), "Error (" + level + ", " + seed + "): m and t differ on a key (" +o+") after removal (iterating
on t)");
    ensure(m.keySet().contains(o), "Error (" + level + ", " + seed + "): m and t differ on a key (" +o+", in keySet()) after
removal (iterating on t)");
}

/* Now we check that m actually holds the same keys, but iterating on m. */

for(Iterator i=m.keySet().iterator(); i.hasNext();) {
    Object o = i.next();
    ensure(t.containsKey(o), "Error (" + level + ", " + seed + "): m and t differ on a key after removal (iterating on m)");
    ensure(t.keySet().contains(o), "Error (" + level + ", " + seed + "): m and t differ on a key (in keySet()) after removal
(iterating on m)");
}

/* Now we check that m actually hold the same values. */

for(Iterator i=t.values().iterator(); i.hasNext();) {

```

```

    Object o = i.next();
    ensure(m.containsValue(o), "Error (" + level + ", " + seed + "): m and t differ on a value after removal (iterating on
t)");
    ensure(m.values().contains(o), "Error (" + level + ", " + seed + "): m and t differ on a value (in values()) after
removal (iterating on t)");
}

/* Now we check that m actually hold the same values, but iterating on m. */

for(Iterator i=m.values().iterator(); i.hasNext(); ) {
    Object o = i.next();
    ensure(t.containsValue(o), "Error (" + level + ", " + seed + "): m and t differ on a value after removal (iterating on
m)");
    ensure(t.values().contains(o), "Error (" + level + ", " + seed + "): m and t differ on a value (in values()) after removal
(iterating on m)");
}

/* Now we check that both maps agree on first/last keys. */

mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

try {
    m.firstKey();
}
catch (NoSuchElementException e) { mThrowsNoElement = true; }
try {
    t.firstKey();
}
catch (NoSuchElementException e) { tThrowsNoElement = true; }

ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): firstKey() divergence in
NoSuchElementException (" + mThrowsNoElement + ", " + tThrowsNoElement + "));
if (! mThrowsNoElement) ensure(t.firstKey().equals(m.firstKey()), "Error (" + level + ", " + seed + "): m and t differ
on their first key (" + m.firstKey() + ", " + t.firstKey() + "));

mThrowsNoElement = mThrowsIllegal = tThrowsNoElement = tThrowsIllegal = false;

try {
    m.lastKey();
}
catch (NoSuchElementException e) { mThrowsNoElement = true; }
try {
    t.lastKey();
}
catch (NoSuchElementException e) { tThrowsNoElement = true; }

ensure(mThrowsNoElement == tThrowsNoElement, "Error (" + level + ", " + seed + "): lastKey() divergence in
NoSuchElementException (" + mThrowsNoElement + ", " + tThrowsNoElement + "));

```

```
if (! mThrowsNoElement) ensure(t.lastKey().equals(m.lastKey()), "Error (" + level + ", " + seed + "): m and t differ on their last key (" + m.lastKey() + ", " + t.lastKey() + "));
```

```
/* Now we check cloning. */
```

```
if (level == 0) {  
    ensure(m.equals(((RB_TREE_MAP)m).clone()), "Error (" + level + ", " + seed + "): m does not equal m.clone()");  
    ensure(((RB_TREE_MAP)m).clone().equals(m), "Error (" + level + ", " + seed + "): m.clone() does not equal m");  
    m = (RB_TREE_MAP)((RB_TREE_MAP)m).clone();  
}
```

```
int h = m.hashCode();
```

```
/* Now we save and read m. */
```

```
SORTED_MAP m2 = null;
```

```
try {  
    java.io.File ff = new java.io.File("it.unimi.dsi.fastutil.test");  
    java.io.OutputStream os = new java.io.FileOutputStream(ff);  
    java.io.ObjectOutputStream oos = new java.io.ObjectOutputStream(os);
```

```
    oos.writeObject(m);  
    oos.close();
```

```
    java.io.InputStream is = new java.io.FileInputStream(ff);  
    java.io.ObjectInputStream ois = new java.io.ObjectInputStream(is);
```

```
    m2 = (SORTED_MAP)ois.readObject();  
    ois.close();  
    ff.delete();
```

```
    }  
    catch(Exception e) {  
        e.printStackTrace();  
        System.exit(1);  
    }
```

```
##if !VALUE_CLASS_Reference
```

```
    ensure(m2.hashCode() == h, "Error (" + level + ", " + seed + "): hashCode() changed after save/read");
```

```
/* Now we check that m2 actually holds that data. */
```

```
    ensure(m2.equals(t), "Error (" + level + ", " + seed + "): ! m2.equals(t) after save/read");  
    ensure(t.equals(m2), "Error (" + level + ", " + seed + "): ! t.equals(m2) after save/read");
```

```
/* Now we take out of m everything, and check that it is empty. */
```

```
##else
```

```

m2.clear();
m2.putAll(m);
#endif

for(Iterator i=t.keySet().iterator(); i.hasNext(); m2.remove(i.next()));

ensure(m2.isEmpty(), "Error (" + level + ", " + seed + "): m2 is not empty (as it should be)");

/* Now we play with iterators. */

{
java.util.ListIterator i, j;
Map.Entry E, F;
Object J;
i = (java.util.ListIterator)m.entrySet().iterator();
j = new java.util.LinkedList(t.entrySet()).listIterator();

for(int k = 0; k < 2*n; k++) {
ensure(i.hasNext() == j.hasNext(), "Error (" + level + ", " + seed + "): divergence in hasNext()");
ensure(i.hasPrevious() == j.hasPrevious(), "Error (" + level + ", " + seed + "): divergence in hasPrevious()");

if (r.nextFloat() < .8 && i.hasNext()) {
ensure((E=(Entry)i.next()).getKey().equals(J = (F=(Map.Entry)j.next()).getKey()), "Error (" + level + ", " + seed +
"): divergence in next()");

if (r.nextFloat() < 0.3) {
i.remove();
j.remove();
t.remove(J);
}
else if (r.nextFloat() < 0.3) {
Object U = VALUE2OBJ(genValue());
E.setValue(U);
t.put(F.getKey(), U);
}
}
else if (r.nextFloat() < .2 && i.hasPrevious()) {
ensure((E=(Entry)i.previous()).getKey().equals(J = (F=(Map.Entry)j.previous()).getKey()), "Error (" + level + ", " +
seed + "): divergence in previous()");

if (r.nextFloat() < 0.3) {
i.remove();
j.remove();
t.remove(J);
}
else if (r.nextFloat() < 0.3) {
Object U = VALUE2OBJ(genValue());
E.setValue(U);
}
}
}
}

```

```

        t.put(F.getKey(), U);
    }
}

ensure(i.nextIndex() == j.nextIndex(), "Error (" + level + ", " + seed + "): divergence in nextIndex()");
ensure(i.previousIndex() == j.previousIndex(), "Error (" + level + ", " + seed + "): divergence in previousIndex()");

}
}

{
boolean badPrevious = false;
Object previous = null;
it.unimi.dsi.fastutil.BidirectionalIterator i;
java.util.ListIterator j;
Object I, J;
KEY_TYPE from = genKey();
j = new java.util.LinkedList(t.keySet()).listIterator();
while(j.hasNext()) {
    Object k = j.next();
    if (((Comparable)k).compareTo(KEY2OBJ(from)) > 0) {
        badPrevious = true;
        j.previous();
        break;
    }
    previous = k;
}

i = (it.unimi.dsi.fastutil.BidirectionalIterator)((SORTED_SET)m.keySet()).iterator(from);

for(int k = 0; k < 2*n; k++) {
    ensure(i.hasNext() == j.hasNext(), "Error (" + level + ", " + seed + "): divergence in hasNext() (iterator with starting point " + from + ")");
    ensure(i.hasPrevious() == j.hasPrevious() || badPrevious && (i.hasPrevious() == (previous != null)), "Error (" + level + ", " + seed + "): divergence in hasPrevious() (iterator with starting point " + from + ")" + badPrevious);

    if (r.nextFloat() < .8 && i.hasNext()) {
        ensure((I = i.next()).equals(J = j.next()), "Error (" + level + ", " + seed + "): divergence in next() (" + I + ", " + J + ", iterator with starting point " + from + ")");
        //System.err.println("Done next " + I + " " + J + " " + badPrevious);

        badPrevious = false;

        if (r.nextFloat() < 0.5) {
            //System.err.println("Removing in next");
            i.remove();
            j.remove();
            t.remove(J);

```

```

    }
    }
    else if (!badPrevious && r.nextFloat() < .2 && i.hasPrevious()) {
        ensure((I = i.previous()).equals(J = j.previous()), "Error (" + level + ", " + seed + "): divergence in previous() (" + I +
", " + J + ", iterator with starting point " + from + ")");

        if (r.nextFloat() < 0.5) {
            //System.err.println("Removing in prev");
            i.remove();
            j.remove();
            t.remove(J);
        }
    }
}

/* Now we check that m actually holds that data. */

ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after iteration");
ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after iteration");

/* Now we select a pair of keys and create a submap. */

if (! m.isEmpty()) {
    java.util.ListIterator i;
    Object start = m.firstKey(), end = m.firstKey();
    for(i = (java.util.ListIterator)m.keySet().iterator(); i.hasNext() && r.nextFloat() < .3; start = end = i.next());
    for(; i.hasNext() && r.nextFloat() < .95; end = i.next());

    //System.err.println("Checking subMap from " + start + " to " + end + " (level=" + (level+1) + ")...");
    testMaps((SORTED_MAP)m.subMap((KEY_CLASS) start, (KEY_CLASS)end), t.subMap(start, end), n, level +
1);

    ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after subMap");
    ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after subMap");

    //System.err.println("Checking headMap to " + end + " (level=" + (level+1) + ")...");
    testMaps((SORTED_MAP)m.headMap((KEY_CLASS)end), t.headMap(end), n, level + 1);

    ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after headMap");
    ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after headMap");

    //System.err.println("Checking tailMap from " + start + " (level=" + (level+1) + ")...");
    testMaps((SORTED_MAP)m.tailMap((KEY_CLASS)start), t.tailMap(start), n, level + 1);

    ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after tailMap");
    ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after tailMap");

```

```

    }

}

private static void runTest(int n) {
    RB_TREE_MAP m = new RB_TREE_MAP();
    SortedMap t = new java.util.TreeMap();
    topMap = m;
    k = new Object[n];
    v = new Object[n];
    nk = new Object[n];
    kt = new KEY_TYPE[n];
    nkt = new KEY_TYPE[n];
    vt = new VALUE_TYPE[n];

    for(int i = 0; i < n; i++) {
#ifdef KEY_CLASS_Object
        k[i] = kt[i] = genKey();
        nk[i] = nkt[i] = genKey();
#else
        k[i] = new KEY_CLASS(kt[i] = genKey());
        nk[i] = new KEY_CLASS(nkt[i] = genKey());
#endif
#ifdef VALUES_REFERENCE
        v[i] = vt[i] = genValue();
#else
        v[i] = new VALUE_CLASS(vt[i] = genValue());
#endif
    }

    /* We add pairs to t. */
    for(int i = 0; i < n; i++) t.put(k[i], v[i]);

    /* We add to m the same data */
    m.putAll(t);

    testMaps(m, t, n, 0);

    System.out.println("Test OK");
    return;
}

public static void main(String args[]) {
    int n = Integer.parseInt(args[1]);
    if (args.length > 2) r = new java.util.Random(seed = Long.parseLong(args[2]));
}

```

```

try {
  if ("speedTest".equals(args[0]) || "speedComp".equals(args[0])) speedTest(n, "speedComp".equals(args[0]));
  else if ("test".equals(args[0])) runTest(n);
} catch(Throwable e) {
  e.printStackTrace(System.err);
  System.err.println("seed: " + seed);
}
}

#endif

}

```

Found in path(s):

```
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-
a775574/drv/RBTreeMap.drv
```

No license file was found, but licenses were detected in source scan.

```

/*
 * Copyright (C) 2002-2017 Sebastiano Vigna
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

```

```
package PACKAGE;
```

```

import java.util.Iterator;
#if defined JDK_PRIMITIVE_ITERATOR && !KEY_WIDENED
import java.util.PrimitiveIterator;
#endif
#if (!defined JDK_PRIMITIVE_ITERATOR || KEY_WIDENED) && KEYS_PRIMITIVE
import java.util.Objects;
#endif
#ifdef KEYS_PRIMITIVE
import java.util.function.Consumer;

```

```

#endif

/** A type-specific {@link Iterator}; provides an additional method to avoid (un)boxing, and
 * the possibility to skip elements.
 *
 * @see Iterator
 */

#if defined JDK_PRIMITIVE_ITERATOR && !KEY_WIDENED
public interface KEY_ITERATOR KEY_GENERIC extends JDK_PRIMITIVE_ITERATOR {
#else
public interface KEY_ITERATOR KEY_GENERIC extends Iterator<KEY_GENERIC_CLASS> {
#endif

#if KEYS_PRIMITIVE
/**
 * Returns the next element as a primitive type.
 *
 * @return the next element in the iteration.
 * @see Iterator#next()
 */

#if defined JDK_PRIMITIVE_ITERATOR && !KEY_WIDENED
@Override
#endif
KEY_TYPE NEXT_KEY();

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default KEY_CLASS next() {
return KEY_CLASS.valueOf(NEXT_KEY());
}

#if !defined JDK_PRIMITIVE_ITERATOR || KEY_WIDENED
/**
 * Performs the given action for each remaining element until all elements
 * have been processed or the action throws an exception.
 * @param action the action to be performed for each element.
 * @see java.util.Iterator#forEachRemaining(java.util.function.Consumer)
 * @since 8.0.0
 */
default void forEachRemaining(final KEY_CONSUMER action) {
Objects.requireNonNull(action);
while (hasNext()) {
action.accept(NEXT_KEY());
}
}
}

```

```

}
#endif

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default void forEachRemaining(final Consumer<? super KEY_GENERIC_CLASS> action) {
#if defined JDK_PRIMITIVE_ITERATOR && !KEY_WIDENED
    forEachRemaining((JDK_PRIMITIVE_KEY_CONSUMER) action::accept);
#else
    forEachRemaining((KEY_CONSUMER) action::accept);
#endif
}

#endif

/** Skips the given number of elements.
 *
 * <p>The effect of this call is exactly the same as that of calling { @link #next()} for { @code n} times (possibly
stopping if { @link #hasNext()} becomes false).
 *
 * @param n the number of elements to skip.
 * @return the number of elements actually skipped.
 * @see Iterator#next()
 */

default int skip(final int n) {
    if (n < 0) throw new IllegalArgumentException("Argument must be nonnegative: " + n);
    int i = n;
    while(i-- != 0 && hasNext()) NEXT_KEY();
    return n - i - 1;
}
}

```

Found in path(s):

```

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-
a775574/drv/Iterator.drv

```

No license file was found, but licenses were detected in source scan.

```

/*
 * Copyright (C) 2003-2017 Paolo Boldi and Sebastiano Vigna
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0

```

```
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

```
package PACKAGE;
```

```
import it.unimi.dsi.fastutil.IndirectPriorityQueue;
```

```
/** A type-specific { @link IndirectPriorityQueue}.
```

```
*
```

```
* <p>Additionally, this interface strengthens { @link #comparator()}.
```

```
*/
```

```
public interface INDIRECT_PRIORITY_QUEUE extends IndirectPriorityQueue<KEY_CLASS> {
```

```
/** Returns the type-specific comparator associated with this queue.
```

```
*
```

```
* <p>Note that this specification strengthens the one given in { @link IndirectPriorityQueue}.
```

```
*
```

```
* @return the comparator associated with this queue.
```

```
* @see IndirectPriorityQueue#comparator()
```

```
*/
```

```
@Override
```

```
KEY_COMPARATOR comparator();
```

```
}
```

```
Found in path(s):
```

```
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-
a775574/drv/IndirectPriorityQueue.drv
```

```
No license file was found, but licenses were detected in source scan.
```

```
/*
```

```
* Copyright (C) 2002-2017 Sebastiano Vigna
```

```
*
```

```
* Licensed under the Apache License, Version 2.0 (the "License");
```

```
* you may not use this file except in compliance with the License.
```

```
* You may obtain a copy of the License at
```

```
*
```

```
* http://www.apache.org/licenses/LICENSE-2.0
```

```
*
```

```
* Unless required by applicable law or agreed to in writing, software
```

```
* distributed under the License is distributed on an "AS IS" BASIS,
```

```
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
```

* See the License for the specific language governing permissions and
* limitations under the License.
*/

package PACKAGE;

import java.lang.Iterable;
#if KEYS_PRIMITIVE
import java.util.Objects;
import java.util.function.Consumer;

/** A type-specific {@link Iterable} that strengthens that specification of {@link #iterator()} and {@link #forEach(Consumer)}.

*

* <p>Note that whenever there exist a primitive consumer in {@link java.util.function} (e.g., {@link java.util.function.IntConsumer}),

* trying to access any version of {@link #forEach(Consumer)} using a lambda expression with untyped arguments

* will generate an ambiguous method error. This can be easily solved by specifying the type of the argument, as in

* <pre>

* intIterable.forEach((int x) -> { // Do something with x });

* </pre>

* <p>The same problem plagues, for example, {@link

java.util.PrimitiveIterator.OfInt#forEachRemaining(java.util.function.IntConsumer)}.

*

* <p>Warning: Java will let you write “colon” {@code for} statements with primitive-type

* loop variables; however, what is (unfortunately) really happening is that at each iteration an

* unboxing (and, in the case of {@code fastutil} type-specific data structures, a boxing) will be performed. Watch out.

*

* @see Iterable

*/

#else

/** A type-specific {@link Iterable} that strengthens that specification of {@link #iterator()}.

*

* @see Iterable

*/

#endif

public interface KEY_ITERABLE KEY_GENERIC extends Iterable<KEY_GENERIC_CLASS> {

/** Returns a type-specific iterator.

*

```

* <p>Note that this specification strengthens the one given in { @link Iterable#iterator()}.
*
* @return a type-specific iterator.
* @see Iterable#iterator()
*/

@Override
KEY_ITERATOR KEY_GENERIC iterator();

#if KEYS_PRIMITIVE
/**
 * Performs the given action for each element of this type-specific { @link java.lang.Iterable}
 * until all elements have been processed or the action throws an
 * exception.
 * @param action the action to be performed for each element.
 * @see java.lang.Iterable#forEach(java.util.function.Consumer)
 * @since 8.0.0
 */
#endif

#ifdef JDK_PRIMITIVE_KEY_CONSUMER
@SuppressWarnings("overloads")
default void forEach(final JDK_PRIMITIVE_KEY_CONSUMER action) {
#else
default void forEach(final KEY_CONSUMER action) {
#endif
Objects.requireNonNull(action);
for(final KEY_ITERATOR iterator = iterator(); iterator.hasNext();)
    action.accept(iterator.NEXT_KEY());
}

/** { @inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default void forEach(final Consumer<? super KEY_GENERIC_CLASS> action) {
#ifdef JDK_PRIMITIVE_KEY_CONSUMER
#ifdef KEY_WIDENED
forEach(new JDK_PRIMITIVE_KEY_CONSUMER() {
    public void accept(final KEY_TYPE_WIDENED key) {
        action.accept(KEY2OBJ((KEY_TYPE)key));
    }
});
#else
forEach((JDK_PRIMITIVE_KEY_CONSUMER) action::accept);
#endif
#else
forEach((KEY_CONSUMER) action::accept);
#endif
}

```

```
}  
#endif
```

```
}
```

Found in path(s):

```
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-  
a775574/drv/Iterable.drv
```

No license file was found, but licenses were detected in source scan.

```
/*
```

```
* Copyright (C) 2002-2017 Sebastiano Vigna
```

```
*
```

```
* Licensed under the Apache License, Version 2.0 (the "License");
```

```
* you may not use this file except in compliance with the License.
```

```
* You may obtain a copy of the License at
```

```
*
```

```
* http://www.apache.org/licenses/LICENSE-2.0
```

```
*
```

```
* Unless required by applicable law or agreed to in writing, software
```

```
* distributed under the License is distributed on an "AS IS" BASIS,
```

```
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
```

```
* See the License for the specific language governing permissions and
```

```
* limitations under the License.
```

```
*/
```

```
package PACKAGE;
```

```
import java.util.Arrays;
```

```
import java.util.Collection;
```

```
import java.util.Iterator;
```

```
import java.util.RandomAccess;
```

```
import java.util.NoSuchElementException;
```

```
#if KEYS_PRIMITIVE
```

```
/** A type-specific array-based list; provides some additional methods that use polymorphism to avoid (un)boxing.
```

```
*
```

```
* <p>This class implements a lightweight, fast, open, optimized,
```

```
* reuse-oriented version of array-based lists. Instances of this class
```

```
* represent a list with an array that is enlarged as needed when new entries
```

```
* are created (by doubling its current length), but is
```

```
* <em>never</em> made smaller (even on a { @link #clear()}). A family of
```

```
* { @linkplain #trim() trimming methods} lets you control the size of the
```

```
* backing array; this is particularly useful if you reuse instances of this class.
```

```
* Range checks are equivalent to those of { @link java.util}'s classes, but
```

```
* they are delayed as much as possible. The backing array is exposed by the
```

```

* {@link #elements()} method.
*
* <p>This class implements the bulk methods {@code removeElements()},
* {@code addElements()} and {@code getElements()} using
* high-performance system calls (e.g., {@link
* System#arraycopy(Object,int,Object,int,int) System.arraycopy()} instead of
* expensive loops.
*
* @see java.util.ArrayList
*/

```

```

public class ARRAY_LIST KEY_GENERIC extends ABSTRACT_LIST KEY_GENERIC implements
RandomAccess, Cloneable, java.io.Serializable {
private static final long serialVersionUID = -7046029254386353130L;

```

```

#else

```

```

/** A type-specific array-based list; provides some additional methods that use polymorphism to avoid (un)boxing.
*
* <p>This class implements a lightweight, fast, open, optimized,
* reuse-oriented version of array-based lists. Instances of this class
* represent a list with an array that is enlarged as needed when new entries
* are created (by doubling the current length), but is
* <em>never</em> made smaller (even on a {@link #clear()}). A family of
* {@linkplain #trim() trimming methods} lets you control the size of the
* backing array; this is particularly useful if you reuse instances of this class.
* Range checks are equivalent to those of {@link java.util}'s classes, but
* they are delayed as much as possible.
*
* <p>The backing array is exposed by the {@link #elements()} method. If an instance
* of this class was created {@linkplain #wrap(Object[],int) by wrapping},
* backing-array reallocations will be performed using reflection, so that
* {@link #elements()} can return an array of the same type of the original array: the comments
* about efficiency made in {@link it.unimi.dsi.fastutil.objects.ObjectArrays} apply here.
* Moreover, you must take into consideration that assignment to an array
* not of type {@code Object[]} is slower due to type checking.
*
* <p>This class implements the bulk methods {@code removeElements()},
* {@code addElements()} and {@code getElements()} using
* high-performance system calls (e.g., {@link
* System#arraycopy(Object,int,Object,int,int) System.arraycopy()} instead of
* expensive loops.
*
* @see java.util.ArrayList
*/

```

```

public class ARRAY_LIST KEY_GENERIC extends ABSTRACT_LIST KEY_GENERIC implements

```

```

RandomAccess, Cloneable, java.io.Serializable {
    private static final long serialVersionUID = -7046029254386353131L;

#endif

    /** The initial default capacity of an array list. */
    public static final int DEFAULT_INITIAL_CAPACITY = 10;

#if ! KEYS_PRIMITIVE
    /** Whether the backing array was passed to {@code wrap()}. In
     * this case, we must reallocate with the same type of array. */
    protected final boolean wrapped;
#endif

    /** The backing array. */
    protected transient KEY_GENERIC_TYPE a[];

    /** The current actual size of the list (never greater than the backing-array length). */
    protected int size;

    /** Creates a new array list using a given array.
     *
     * <p>This constructor is only meant to be used by the wrapping methods.
     *
     * @param a the array that will be used to back this array list.
     */

    protected ARRAY_LIST(final KEY_GENERIC_TYPE a[], @SuppressWarnings("unused") boolean dummy) {
        this.a = a;
    }

#if ! KEYS_PRIMITIVE
        this.wrapped = true;
#endif
    }

    /** Creates a new array list with given capacity.
     *
     * @param capacity the initial capacity of the array list (may be 0).
     */

    SUPPRESS_WARNINGS_KEY_UNCHECKED
    public ARRAY_LIST(final int capacity) {
        if (capacity < 0) throw new IllegalArgumentException("Initial capacity (" + capacity + ") is negative");
        if (capacity == 0) a = KEY_GENERIC_ARRAY_CAST ARRAYS.EMPTY_ARRAY;
        else a = KEY_GENERIC_ARRAY_CAST new KEY_TYPE[capacity];
    }

#if ! KEYS_PRIMITIVE
        wrapped = false;
#endif
}

```

```

}

/** Creates a new array list with {@link #DEFAULT_INITIAL_CAPACITY} capacity. */

SUPPRESS_WARNINGS_KEY_UNCHECKED
public ARRAY_LIST() {
    a = KEY_GENERIC_ARRAY_CAST ARRAYS.DEFAULT_EMPTY_ARRAY; // We delay allocation
#if ! KEYS_PRIMITIVE
    wrapped = false;
#endif
}

/** Creates a new array list and fills it with a given collection.
 *
 * @param c a collection that will be used to fill the array list.
 */

public ARRAY_LIST(final Collection<? extends KEY_GENERIC_CLASS> c) {
    this(c.size());
#if KEYS_PRIMITIVE
    size = ITERATORS.unwrap(ITERATORS.AS_KEY_ITERATOR(c.iterator()), a);
#else
    size = ITERATORS.unwrap(c.iterator(), a);
#endif
}

/** Creates a new array list and fills it with a given type-specific collection.
 *
 * @param c a type-specific collection that will be used to fill the array list.
 */

public ARRAY_LIST(final COLLECTION KEY_EXTENDS_GENERIC c) {
    this(c.size());
    size = ITERATORS.unwrap(c.iterator(), a);
}

/** Creates a new array list and fills it with a given type-specific list.
 *
 * @param l a type-specific list that will be used to fill the array list.
 */

public ARRAY_LIST(final LIST KEY_EXTENDS_GENERIC l) {
    this(l.size());
    l.getElements(0, a, 0, size = l.size());
}

/** Creates a new array list and fills it with the elements of a given array.
 *

```

```

* @param a an array whose elements will be used to fill the array list.
*/

public ARRAY_LIST(final KEY_GENERIC_TYPE a[]) {
    this(a, 0, a.length);
}

/** Creates a new array list and fills it with the elements of a given array.
*
* @param a an array whose elements will be used to fill the array list.
* @param offset the first element to use.
* @param length the number of elements to use.
*/

public ARRAY_LIST(final KEY_GENERIC_TYPE a[], final int offset, final int length) {
    this(length);
    System.arraycopy(a, offset, this.a, 0, length);
    size = length;
}

/** Creates a new array list and fills it with the elements returned by an iterator..
*
* @param i an iterator whose returned elements will fill the array list.
*/

public ARRAY_LIST(final Iterator<? extends KEY_GENERIC_CLASS> i) {
    this();
    while(i.hasNext()) this.add(KEY_CLASS2TYPE(i.next()));
}

/** Creates a new array list and fills it with the elements returned by a type-specific iterator..
*
* @param i a type-specific iterator whose returned elements will fill the array list.
*/

public ARRAY_LIST(final KEY_ITERATOR KEY_EXTENDS_GENERIC i) {
    this();
    while(i.hasNext()) this.add(i.NEXT_KEY());
}

#if KEYS_PRIMITIVE
/** Returns the backing array of this list.
*
* @return the backing array.
*/

public KEY_GENERIC_TYPE[] elements() {
    return a;
}

```

```

}
#else
/** Returns the backing array of this list.
 *
 * <p>If this array list was created by wrapping a given array, it is guaranteed
 * that the type of the returned array will be the same. Otherwise, the returned
 * array will be of type {@link Object Object[]} (in spite of the declared return type).
 *
 * <p><strong>Warning</strong>: This behaviour may cause (unfathomable)
 * run-time errors if a method expects an array
 * actually of type {@code K[]}, but this methods returns an array
 * of type {@link Object Object[]}.
 *
 * @return the backing array.
 */

public K[] elements() {
    return a;
}
#endif

/** Wraps a given array into an array list of given size.
 *
 * <p>Note it is guaranteed
 * that the type of the array returned by {@link #elements()} will be the same
 * (see the comments in the class documentation).
 *
 * @param a an array to wrap.
 * @param length the length of the resulting array list.
 * @return a new array list of the given size, wrapping the given array.
 */

public static KEY_GENERIC ARRAY_LIST KEY_GENERIC wrap(final KEY_GENERIC_TYPE a[], final int
length) {
    if (length > a.length) throw new IllegalArgumentException("The specified length (" + length + ") is greater than the
array size (" + a.length + ")");
    final ARRAY_LIST KEY_GENERIC l = new ARRAY_LIST KEY_GENERIC DIAMOND(a, false);
    l.size = length;
    return l;
}

/** Wraps a given array into an array list.
 *
 * <p>Note it is guaranteed
 * that the type of the array returned by {@link #elements()} will be the same
 * (see the comments in the class documentation).
 *
 * @param a an array to wrap.

```

```

* @return a new array list wrapping the given array.
*/

public static KEY_GENERIC ARRAY_LIST KEY_GENERIC wrap(final KEY_GENERIC_TYPE a[]) {
    return wrap(a, a.length);
}

/** Ensures that this array list can contain the given number of entries without resizing.
 *
 * @param capacity the new minimum capacity for this array list.
 */
SUPPRESS_WARNINGS_KEY_UNCHECKED
public void ensureCapacity(final int capacity) {
    if (capacity <= a.length || (a == ARRAYS.DEFAULT_EMPTY_ARRAY && capacity <=
DEFAULT_INITIAL_CAPACITY)) return;
#if KEYS_PRIMITIVE
    a = ARRAYS.ensureCapacity(a, capacity, size);
#else
    if (wrapped) a = ARRAYS.ensureCapacity(a, capacity, size);
    else {
        if (capacity > a.length) {
            final Object t[] = new Object[capacity];
            System.arraycopy(a, 0, t, 0, size);
            a = (KEY_GENERIC_TYPE[])t;
        }
    }
#endif
    assert size <= a.length;
}

/** Grows this array list, ensuring that it can contain the given number of entries without resizing,
 * and in case increasing the current capacity at least by a factor of 50%.
 *
 * @param capacity the new minimum capacity for this array list.
 */
SUPPRESS_WARNINGS_KEY_UNCHECKED
private void grow(int capacity) {
    if (capacity <= a.length) return;
    if (a != ARRAYS.DEFAULT_EMPTY_ARRAY)
        capacity = (int)Math.max(Math.min((long)a.length + (a.length >> 1),
it.unimi.dsi.fastutil.Arrays.MAX_ARRAY_SIZE), capacity);
    else if (capacity < DEFAULT_INITIAL_CAPACITY) capacity = DEFAULT_INITIAL_CAPACITY;
#if KEYS_PRIMITIVE
    a = ARRAYS.forceCapacity(a, capacity, size);
#else
    if (wrapped) a = ARRAYS.forceCapacity(a, capacity, size);
    else {

```

```

    final Object t[] = new Object[capacity];
    System.arraycopy(a, 0, t, 0, size);
    a = (KEY_GENERIC_TYPE[])t;
}
#endif
assert size <= a.length;
}

@Override
public void add(final int index, final KEY_GENERIC_TYPE k) {
    ensureIndex(index);
    grow(size + 1);
    if (index != size) System.arraycopy(a, index, a, index + 1, size - index);
    a[index] = k;
    size++;
    assert size <= a.length;
}

@Override
public boolean add(final KEY_GENERIC_TYPE k) {
    grow(size + 1);
    a[size++] = k;
    assert size <= a.length;
    return true;
}

@Override
public KEY_GENERIC_TYPE GET_KEY(final int index) {
    if (index >= size) throw new IndexOutOfBoundsException("Index (" + index + ") is greater than or equal to list size (" + size + ")");
    return a[index];
}

@Override
public int indexOf(final KEY_TYPE k) {
    for(int i = 0; i < size; i++) if (KEY_EQUALS(k, a[i])) return i;
    return -1;
}

@Override
public int lastIndexOf(final KEY_TYPE k) {
    for(int i = size; i-- != 0;) if (KEY_EQUALS(k, a[i])) return i;
    return -1;
}

@Override
public KEY_GENERIC_TYPE REMOVE_KEY(final int index) {

```

```

    if (index >= size) throw new IndexOutOfBoundsException("Index (" + index + ") is greater than or equal to list size (" + size + ")");
    final KEY_GENERIC_TYPE old = a[index];
    size--;
    if (index != size) System.arraycopy(a, index + 1, a, index, size - index);
#if KEYS_REFERENCE
    a[size] = null;
#endif
    assert size <= a.length;
    return old;
}

```

```

@Override
public boolean REMOVE(final KEY_TYPE k) {
    int index = indexOf(k);
    if (index == -1) return false;
    REMOVE_KEY(index);
    assert size <= a.length;
    return true;
}

```

```

@Override
public KEY_GENERIC_TYPE set(final int index, final KEY_GENERIC_TYPE k) {
    if (index >= size) throw new IndexOutOfBoundsException("Index (" + index + ") is greater than or equal to list size (" + size + ")");
    KEY_GENERIC_TYPE old = a[index];
    a[index] = k;
    return old;
}

```

```

@Override
public void clear() {
#if KEYS_REFERENCE
    Arrays.fill(a, 0, size, null);
#endif
    size = 0;
    assert size <= a.length;
}

```

```

@Override
public int size() {
    return size;
}

```

```

@Override
public void size(final int size) {
    if (size > a.length) a = ARRAYS.forceCapacity(a, size, this.size);
    if (size > this.size) Arrays.fill(a, this.size, size, KEY_NULL);
}

```

```

#if KEYS_REFERENCE
    else Arrays.fill(a, size, this.size, KEY_NULL);
#endif
    this.size = size;
}

@Override
public boolean isEmpty() {
    return size == 0;
}

/** Trims this array list so that the capacity is equal to the size.
 *
 * @see java.util.ArrayList#trimToSize()
 */
public void trim() {
    trim(0);
}

/** Trims the backing array if it is too large.
 *
 * If the current array length is smaller than or equal to
 * { @code n}, this method does nothing. Otherwise, it trims the
 * array length to the maximum between { @code n} and { @link #size()}.
 *
 * <p>This method is useful when reusing lists. { @linkplain #clear() Clearing a
 * list} leaves the array length untouched. If you are reusing a list
 * many times, you can call this method with a typical
 * size to avoid keeping around a very large array just
 * because of a few large transient lists.
 *
 * @param n the threshold for the trimming.
 */

SUPPRESS_WARNINGS_KEY_UNCHECKED
public void trim(final int n) {
    // TODO: use Arrays.trim() and preserve type only if necessary
    if (n >= a.length || size == a.length) return;
    final KEY_GENERIC_TYPE t[] = KEY_GENERIC_ARRAY_CAST new KEY_TYPE[Math.max(n, size)];
    System.arraycopy(a, 0, t, 0, size);
    a = t;
    assert size <= a.length;
}

/** Copies element of this type-specific list into the given array using optimized system calls.
 *
 * @param from the start index (inclusive).

```

```

* @param a the destination array.
* @param offset the offset into the destination array where to store the first element copied.
* @param length the number of elements to be copied.
*/
@Override
public void getElements(final int from, final KEY_TYPE[] a, final int offset, final int length) {
    ARRAYS.ensureOffsetLength(a, offset, length);
    System.arraycopy(this.a, from, a, offset, length);
}

/** Removes elements of this type-specific list using optimized system calls.
*
* @param from the start index (inclusive).
* @param to the end index (exclusive).
*/
@Override
public void removeElements(final int from, final int to) {
    it.unimi.dsi.fastutil.Arrays.ensureFromTo(size, from, to);
    System.arraycopy(a, to, a, from, size - to);
    size -= (to - from);
#if KEYS_REFERENCE
    int i = to - from;
    while(i-- != 0) a[size + i] = null;
#endif
}

/** Adds elements to this type-specific list using optimized system calls.
*
* @param index the index at which to add elements.
* @param a the array containing the elements.
* @param offset the offset of the first element to add.
* @param length the number of elements to add.
*/
@Override
public void addElements(final int index, final KEY_GENERIC_TYPE a[], final int offset, final int length) {
    ensureIndex(index);
    ARRAYS.ensureOffsetLength(a, offset, length);
    grow(size + length);
    System.arraycopy(this.a, index, this.a, index + length, size - index);
    System.arraycopy(a, offset, this.a, index, length);
    size += length;
}

#if KEYS_PRIMITIVE

```

```

@Override
public KEY_TYPE[] toArray(KEY_TYPE a[]) {
    if (a == null || a.length < size) a = new KEY_TYPE[size];
    System.arraycopy(this.a, 0, a, 0, size);
    return a;
}

@Override
public boolean addAll(int index, final COLLECTION c) {
    ensureIndex(index);
    int n = c.size();
    if (n == 0) return false;
    grow(size + n);
    if (index != size) System.arraycopy(a, index, a, index + n, size - index);
    final KEY_ITERATOR i = c.iterator();
    size += n;
    while(n-- != 0) a[index++] = i.NEXT_KEY();
    assert size <= a.length;
    return true;
}

@Override
public boolean addAll(final int index, final LIST l) {
    ensureIndex(index);
    final int n = l.size();
    if (n == 0) return false;
    grow(size + n);
    if (index != size) System.arraycopy(a, index, a, index + n, size - index);
    l.getElements(0, a, index, n);
    size += n;
    assert size <= a.length;
    return true;
}

@Override
public boolean removeAll(final COLLECTION c) {
    final KEY_TYPE[] a = this.a;
    int j = 0;
    for(int i = 0; i < size; i++)
        if (! c.contains(a[i])) a[j++] = a[i];
    #if KEYS_REFERENCE
        Arrays.fill(a, j, size, null);
    #endif
    final boolean modified = size != j;
    size = j;
    return modified;
}

```

```

#endif

@Override
public boolean removeAll(final Collection<?> c) {
    final KEY_TYPE[] a = this.a;
    int j = 0;
    for(int i = 0; i < size; i++)
        if (! c.contains(KEY2OBJ(a[i]))) a[j++] = a[i];
    #if KEYS_REFERENCE
        Arrays.fill(a, j, size, null);
    #endif
    final boolean modified = size != j;
    size = j;
    return modified;
}

@Override
public KEY_LIST_ITERATOR KEY_GENERIC listIterator(final int index) {
    ensureIndex(index);

    return new KEY_LIST_ITERATOR KEY_GENERIC() {
        int pos = index, last = -1;

        @Override
        public boolean hasNext() { return pos < size; }
        @Override
        public boolean hasPrevious() { return pos > 0; }
        @Override
        public KEY_GENERIC_TYPE NEXT_KEY() { if (! hasNext()) throw new NoSuchElementException(); return
a[last = pos++]; }
        @Override
        public KEY_GENERIC_TYPE PREV_KEY() { if (! hasPrevious()) throw new NoSuchElementException(); return
a[last = --pos]; }
        @Override
        public int nextIndex() { return pos; }
        @Override
        public int previousIndex() { return pos - 1; }
        @Override
        public void add(KEY_GENERIC_TYPE k) {
            ARRAY_LIST.this.add(pos++, k);
            last = -1;
        }
        @Override
        public void set(KEY_GENERIC_TYPE k) {
            if (last == -1) throw new IllegalStateException();
            ARRAY_LIST.this.set(last, k);
        }
        @Override

```

```

public void remove() {
    if (last == -1) throw new IllegalStateException();
    ARRAY_LIST.this.REMOVE_KEY(last);
    /* If the last operation was a next(), we are removing an element *before* us, and we must decrease pos
correspondingly. */
    if (last < pos) pos--;
    last = -1;
}
};
}

```

@Override

```

public ARRAY_LIST KEY_GENERIC clone() {
    ARRAY_LIST KEY_GENERIC c = new ARRAY_LIST KEY_GENERIC_DIAMOND(size);
    System.arraycopy(a, 0, c.a, 0, size);
    c.size = size;
    return c;
}

```

#if KEY_CLASS_Object

```

private boolean valEquals(final K a, final K b) { return a == null ? b == null : a.equals(b); }
#endif

```

/** Compares this type-specific array list to another one.

*

* <p>This method exists only for sake of efficiency. The implementation

* inherited from the abstract implementation would already work.

*

* @param l a type-specific array list.

* @return true if the argument contains the same elements of this type-specific array list.

*/

```

public boolean equals(final ARRAY_LIST KEY_GENERIC l) {
    if (l == this) return true;
    int s = size();
    if (s != l.size()) return false;
    final KEY_GENERIC_TYPE[] a1 = a;
    final KEY_GENERIC_TYPE[] a2 = l.a;

```

#if KEY_CLASS_Object

```

while(s-- != 0) if (! valEquals(a1[s], a2[s])) return false;

```

#else

```

while(s-- != 0) if (a1[s] != a2[s]) return false;

```

#endif

```

return true;

```

```

}

```

#if ! KEY_CLASS_Reference

```

/** Compares this array list to another array list.
 *
 * <p>This method exists only for sake of efficiency. The implementation
 * inherited from the abstract implementation would already work.
 *
 * @param l an array list.
 * @return a negative integer,
 * zero, or a positive integer as this list is lexicographically less than, equal
 * to, or greater than the argument.
 */
SUPPRESS_WARNINGS_KEY_UNCHECKED
public int compareTo(final ARRAY_LIST KEY_EXTENDS_GENERIC l) {
    final int s1 = size(), s2 = l.size();
    final KEY_GENERIC_TYPE a1[] = a, a2[] = l.a;
    KEY_GENERIC_TYPE e1, e2;
    int r, i;

    for(i = 0; i < s1 && i < s2; i++) {
        e1 = a1[i];
        e2 = a2[i];
        if ((r = KEY_CMP(e1, e2)) != 0) return r;
    }

    return i < s2 ? -1 : (i < s1 ? 1 : 0);
}
#endif

private void writeObject(java.io.ObjectOutputStream s) throws java.io.IOException {
    s.defaultWriteObject();
    for(int i = 0; i < size; i++) s.WRITE_KEY(a[i]);
}

SUPPRESS_WARNINGS_KEY_UNCHECKED
private void readObject(java.io.ObjectInputStream s) throws java.io.IOException, ClassNotFoundException {
    s.defaultReadObject();
    a = KEY_GENERIC_ARRAY_CAST new KEY_TYPE[size];
    for(int i = 0; i < size; i++) a[i] = KEY_GENERIC_CAST s.READ_KEY();
}

#ifdef TEST

private static long seed = System.currentTimeMillis();
private static java.util.Random r = new java.util.Random(seed);

private static KEY_TYPE genKey() {

```

```

#if KEY_CLASS_Byte || KEY_CLASS_Short || KEY_CLASS_Character
    return (KEY_TYPE)(r.nextInt());
#elif KEYS_PRIMITIVE
    return r.NEXT_KEY();
#elif KEY_CLASS_Object
    return Integer.toBinaryString(r.nextInt());
#else
    return new java.io.Serializable() {};
#endif
}

private static java.text.NumberFormat format = new java.text.DecimalFormat("#,###.00");
private static java.text.FieldPosition p = new java.text.FieldPosition(0);

private static String format(double d) {
    StringBuffer s = new StringBuffer();
    return format.format(d, s, p).toString();
}

private static void speedTest(int n, boolean comp) {
    System.out.println("There are presently no speed tests for this class.");
}

private static void fatal(String msg) {
    System.out.println(msg);
    System.exit(1);
}

private static void ensure(boolean cond, String msg) {
    if (cond) return;
    fatal(msg);
}

private static Object[] k, v, nk;
private static KEY_TYPE kt[];
private static KEY_TYPE nkt[];
private static ARRAY_LIST topList;

protected static void testLists(LIST m, java.util.List t, int n, int level) {
    long ms;
    Exception mThrowsIllegal, tThrowsIllegal, mThrowsOutOfBounds, tThrowsOutOfBounds;
    Object rt = null;
    KEY_TYPE rm = KEY_NULL;

    if (level > 4) return;

```

```

/* Now we check that both sets agree on random keys. For m we use the polymorphic method. */

for(int i = 0; i < n; i++) {
    int p = r.nextInt() % (n * 2);

    KEY_TYPE T = genKey();
    mThrowsOutOfBounds = tThrowsOutOfBounds = null;

    try {
        m.set(p, T);
    }
    catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }
    try {
        t.set(p, KEY2OBJ(T));
    }
    catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }

    ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + level + ", " + seed + "):
set() divergence at start in IndexOutOfBoundsException for index " + p + " (" + mThrowsOutOfBounds + ", " +
tThrowsOutOfBounds + ")");
    if (mThrowsOutOfBounds == null) ensure(t.get(p).equals(KEY2OBJ(m.GET_KEY(p))), "Error (" + level + ", " +
seed + "): m and t differ after set() on position " + p + " (" + m.GET_KEY(p) + ", " + t.get(p) + ")");

    p = r.nextInt() % (n * 2);
    mThrowsOutOfBounds = tThrowsOutOfBounds = null;

    try {
        m.GET_KEY(p);
    }
    catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }
    try {
        t.get(p);
    }
    catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }

    ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + level + ", " + seed + "):
get() divergence at start in IndexOutOfBoundsException for index " + p + " (" + mThrowsOutOfBounds + ", " +
tThrowsOutOfBounds + ")");
    if (mThrowsOutOfBounds == null) ensure(t.get(p).equals(KEY2OBJ(m.GET_KEY(p))), "Error (" + level + ", " +
seed + "): m and t differ afre get() on position " + p + " (" + m.GET_KEY(p) + ", " + t.get(p) + ")");

}

/* Now we check that both sets agree on random keys. For m we use the standard method. */

for(int i = 0; i < n; i++) {
    int p = r.nextInt() % (n * 2);

```

```

mThrowsOutOfBounds = tThrowsOutOfBounds = null;

try {
    m.get(p);
}
catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }
try {
    t.get(p);
}
catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }

ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + level + ", " + seed + "):
get() divergence at start in IndexOutOfBoundsException for index " + p + " (" + mThrowsOutOfBounds + ", " +
tThrowsOutOfBounds + ")");
if (mThrowsOutOfBounds == null) ensure(t.get(p).equals(m.get(p)), "Error (" + level + ", " + seed + "): m and t
differ at start on position " + p + " (" + m.get(p) + ", " + t.get(p) + ")");

}

/* Now we check that m and t are equal. */
if (!m.equals(t) || !t.equals(m)) System.err.println("m: " + m + " t: " + t);

ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) at start");
ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) at start");

/* Now we check that m actually holds that data. */
for(Iterator i=t.iterator(); i.hasNext(); ) {
    ensure(m.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after insertion (iterating on
t)");
}

/* Now we check that m actually holds that data, but iterating on m. */
for(Iterator i=m.listIterator(); i.hasNext(); ) {
    ensure(t.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after insertion (iterating on
m)");
}

/* Now we check that inquiries about random data give the same answer in m and t. For
m we use the polymorphic method. */

for(int i=0; i<n; i++) {
    KEY_TYPE T = genKey();
    ensure(m.contains(T) == t.contains(KEY2OBJ(T)), "Error (" + level + ", " + seed + "): divergence in content
between t and m (polymorphic method)");
}

```

```

/* Again, we check that inquiries about random data give the same answer in m and t, but
for m we use the standard method. */

for(int i=0; i<n; i++) {
    KEY_TYPE T = genKey();
    ensure(m.contains(KEY2OBJ(T)) == t.contains(KEY2OBJ(T)), "Error (" + level + ", " + seed + "): divergence in
content between t and m (polymorphic method)");
}

/* Now we add and remove random data in m and t, checking that the result is the same. */

for(int i=0; i<2*n; i++) {
    KEY_TYPE T = genKey();

    try {
        m.add(T);
    }
    catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }

    try {
        t.add(KEY2OBJ(T));
    }
    catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }

    T = genKey();
    int p = r.nextInt() % (2 * n + 1);

    mThrowsOutOfBounds = tThrowsOutOfBounds = null;

    try {
        m.add(p, T);
    }
    catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }

    try {
        t.add(p, KEY2OBJ(T));
    }
    catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }

    ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + level + ", " + seed + "):
add() divergence in IndexOutOfBoundsException for index " + p + " for " + T + " (" + mThrowsOutOfBounds + ", "
+ tThrowsOutOfBounds + ")");

    p = r.nextInt() % (2 * n + 1);

    mThrowsOutOfBounds = tThrowsOutOfBounds = null;

```

```

try {
    rm = m.REMOVE_KEY(p);
}
catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }

try {
    rt = t.remove(p);
}
catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }

ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + level + ", " + seed + "):
remove() divergence in IndexOutOfBoundsException for index " + p + " (" + mThrowsOutOfBounds + ", " +
tThrowsOutOfBounds + ")");
if (mThrowsOutOfBounds == null) ensure(rt.equals(KEY2OBJ(rm)), "Error (" + level + ", " + seed + "): divergence
in remove() between t and m (" + rt + ", " + rm + ")");
}

ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after add/remove");
ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after add/remove");

/* Now we add random data in m and t using addAll on a collection, checking that the result is the same. */

for(int i=0; i<n; i++) {
    int p = r.nextInt() % (2 * n + 1);
    Collection m1 = new java.util.ArrayList();
    int s = r.nextInt(n / 2 + 1);
    for(int j = 0; j < s; j++) m1.add(KEY2OBJ(genKey()));

    mThrowsOutOfBounds = tThrowsOutOfBounds = null;

    try {
        m.addAll(p, m1);
    }
    catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }

    try {
        t.addAll(p, m1);
    }
    catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }

    ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + level + ", " + seed + "):
addAll() divergence in IndexOutOfBoundsException for index " + p + " for " + m1 + " (" + mThrowsOutOfBounds
+ ", " + tThrowsOutOfBounds + ")");

    ensure(m.equals(t), "Error (" + level + ", " + seed + m + t + "): ! m.equals(t) after addAll");
    ensure(t.equals(m), "Error (" + level + ", " + seed + m + t + "): ! t.equals(m) after addAll");
}

```

```

if (m.size() > n) {
    m.size(n);
    while(t.size() != n) t.remove(t.size() -1);
}

/* Now we add random data in m and t using addAll on a type-specific collection, checking that the result is the
same. */

for(int i=0; i<n; i++) {
    int p = r.nextInt() % (2 * n + 1);
    COLLECTION m1 = new ARRAY_LIST();
    Collection t1 = new java.util.ArrayList();
    int s = r.nextInt(n / 2 + 1);
    for(int j = 0; j < s; j++) {
        KEY_TYPE x = genKey();
        m1.add(x);
        t1.add(KEY2OBJ(x));
    }

    mThrowsOutOfBounds = tThrowsOutOfBounds = null;

    try {
        m.addAll(p, m1);
    }
    catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }

    try {
        t.addAll(p, t1);
    }
    catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }

    ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + level + ", " + seed + "):
polymorphic addAll() divergence in IndexOutOfBoundsException for index " + p + " for " + m1 + " (" +
mThrowsOutOfBounds + ", " + tThrowsOutOfBounds + ")");

    ensure(m.equals(t), "Error (" + level + ", " + seed + m + t + "): ! m.equals(t) after polymorphic addAll");
    ensure(t.equals(m), "Error (" + level + ", " + seed + m + t + "): ! t.equals(m) after polymorphic addAll");
}

if (m.size() > n) {
    m.size(n);
    while(t.size() != n) t.remove(t.size() -1);
}

/* Now we add random data in m and t using addAll on a list, checking that the result is the same. */

for(int i=0; i<n; i++) {

```

```

int p = r.nextInt() % (2 * n + 1);
LIST m1 = new ARRAY_LIST();
Collection t1 = new java.util.ArrayList();
int s = r.nextInt(n / 2 + 1);
for(int j = 0; j < s; j++) {
    KEY_TYPE x = genKey();
    m1.add(x);
    t1.add(KEY2OBJ(x));
}

mThrowsOutOfBounds = tThrowsOutOfBounds = null;

try {
    m.addAll(p, m1);
}
catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }

try {
    t.addAll(p, t1);
}
catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }

ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + level + ", " + seed + "): list
addAll() divergence in IndexOutOfBoundsException for index " + p + " for " + m1 + " (" + mThrowsOutOfBounds
+ ", " + tThrowsOutOfBounds + ")");

ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after list addAll");
ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after list addAll");
}

/* Now we add random data in m and t using addElements, checking that the result is the same. */

for(int i=0; i<n; i++) {
    int p = r.nextInt() % (2 * n + 1);
    Collection t1 = new java.util.ArrayList();
    int s = r.nextInt(n / 2 + 1);
    KEY_TYPE a[] = new KEY_TYPE [s];
    for(int j = 0; j < s; j++) {
        KEY_TYPE x = genKey();
        t1.add(KEY2OBJ(x));
        a[j] = x;
    }

    mThrowsOutOfBounds = tThrowsOutOfBounds = null;

    try {
        m.addElements(p, a);
    }

```

```

catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }

try {
    t.addAll(p, t1);
}
catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }

ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + level + ", " + seed + "): list
addElement() divergence in IndexOutOfBoundsException for index " + p + " for " + t1 + " (" +
mThrowsOutOfBounds + ", " + tThrowsOutOfBounds + ")");

ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after list addElements");
ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after list addElements");
}

if (m.size() > n) {
    m.size(n);
    while(t.size() != n) t.remove(t.size() -1);
}

/* Now we check that m actually holds the same data. */

for(Iterator i=t.iterator(); i.hasNext();) {
    ensure(m.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after removal (iterating on
t)");
}

/* Now we check that m actually holds that data, but iterating on m. */

for(Iterator i=m.listIterator(); i.hasNext();) {
    ensure(t.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after removal (iterating on
m)");
}

/* Now we check that both sets agree on random keys. For m we use the standard method. */

for(int i = 0; i < n; i++) {
    int p = r.nextInt() % (n * 2);

    mThrowsOutOfBounds = tThrowsOutOfBounds = null;

    try {
        m.get(p);
    }
    catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }

    try {
        t.get(p);
    }
}

```

```

catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }

ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + level + ", " + seed + "):
get() divergence in IndexOutOfBoundsException for index " + p + " (" + mThrowsOutOfBounds + ", " +
tThrowsOutOfBounds + ")");
if (mThrowsOutOfBounds == null) ensure(t.get(p).equals(m.get(p)), "Error (" + level + ", " + seed + "): m and t
differ on position " + p + " (" + m.get(p) + ", " + t.get(p) + ")");

}

/* Now we inquiry about the content with indexOf()/lastIndexOf(). */

for(int i=0; i<10*n; i++) {
KEY_TYPE T = genKey();
ensure(m.indexOf(KEY2OBJ(T)) == t.indexOf(KEY2OBJ(T)),
"Error (" + level + ", " + seed + "): indexOf() divergence for " + T + " (" + m.indexOf(KEY2OBJ(T)) + ", " +
t.indexOf(KEY2OBJ(T)) + ")");
ensure(m.lastIndexOf(KEY2OBJ(T)) == t.lastIndexOf(KEY2OBJ(T)),
"Error (" + level + ", " + seed + "): lastIndexOf() divergence for " + T + " (" + m.lastIndexOf(KEY2OBJ(T)) + ", "
+ t.lastIndexOf(KEY2OBJ(T)) + ")");
ensure(m.indexOf(T) == t.indexOf(KEY2OBJ(T)),
"Error (" + level + ", " + seed + "): polymorphic indexOf() divergence for " + T + " (" + m.indexOf(T) + ", " +
t.indexOf(KEY2OBJ(T)) + ")");
ensure(m.lastIndexOf(T) == t.lastIndexOf(KEY2OBJ(T)),
"Error (" + level + ", " + seed + "): polymorphic lastIndexOf() divergence for " + T + " (" + m.lastIndexOf(T) + ", "
+ t.lastIndexOf(KEY2OBJ(T)) + ")");
}

/* Now we check cloning. */

if (level == 0) {
ensure(m.equals(((ARRAY_LIST)m).clone()), "Error (" + level + ", " + seed + "): m does not equal m.clone()");
ensure(((ARRAY_LIST)m).clone().equals(m), "Error (" + level + ", " + seed + "): m.clone() does not equal m");
}

/* Now we play with constructors. */
ensure(m.equals(new ARRAY_LIST((Collection)m)), "Error (" + level + ", " + seed + "): m does not equal new
(Collection m)");
ensure((new ARRAY_LIST((Collection)m)).equals(m), "Error (" + level + ", " + seed + "): new (Collection m)does
not equal m");
ensure(m.equals(new ARRAY_LIST((COLLECTION)m)), "Error (" + level + ", " + seed + "): m does not equal
new (type-specific Collection m)");
ensure((new ARRAY_LIST((COLLECTION)m)).equals(m), "Error (" + level + ", " + seed + "): new (type-specific
Collection m) does not equal m");
ensure(m.equals(new ARRAY_LIST((LIST)m)), "Error (" + level + ", " + seed + "): m does not equal new (type-
specific List m)");
ensure((new ARRAY_LIST((LIST)m)).equals(m), "Error (" + level + ", " + seed + "): new (type-specific List m)
does not equal m");

```

```

    ensure(m.equals(new ARRAY_LIST(m.listIterator())), "Error (" + level + ", " + seed + "): m does not equal new
(m.listIterator())");
    ensure((new ARRAY_LIST(m.listIterator())).equals(m), "Error (" + level + ", " + seed + "): new (m.listIterator())
does not equal m");
    ensure(m.equals(new ARRAY_LIST(m.iterator())), "Error (" + level + ", " + seed + "): m does not equal new
(m.type_specific_iterator())");
    ensure((new ARRAY_LIST(m.iterator())).equals(m), "Error (" + level + ", " + seed + "): new
(m.type_specific_iterator()) does not equal m");

    /* Now we play with conversion to array, wrapping and copying. */
    ensure(m.equals(new ARRAY_LIST(m.TO_KEY_ARRAY())), "Error (" + level + ", " + seed + "): m does not
equal new (toArray(m))");
    ensure((new ARRAY_LIST(m.TO_KEY_ARRAY())).equals(m), "Error (" + level + ", " + seed + "): new
(toArray(m)) does not equal m");
    ensure(m.equals(wrap(m.TO_KEY_ARRAY())), "Error (" + level + ", " + seed + "): m does not equal wrap
(toArray(m))");
    ensure((wrap(m.TO_KEY_ARRAY())).equals(m), "Error (" + level + ", " + seed + "): wrap (toArray(m)) does not
equal m");

    int h = m.hashCode();

    /* Now we save and read m. */

    LIST m2 = null;

    try {
        java.io.File ff = new java.io.File("it.unimi.dsi.fastutil.test");
        java.io.OutputStream os = new java.io.FileOutputStream(ff);
        java.io.ObjectOutputStream oos = new java.io.ObjectOutputStream(os);

        oos.writeObject(m);
        oos.close();

        java.io.InputStream is = new java.io.FileInputStream(ff);
        java.io.ObjectInputStream ois = new java.io.ObjectInputStream(is);

        m2 = (LIST)ois.readObject();
        ois.close();
        ff.delete();
    }
    catch(Exception e) {
        e.printStackTrace();
        System.exit(1);
    }

    #if ! KEY_CLASS_Reference
        ensure(m2.hashCode() == h, "Error (" + level + ", " + seed + "): hashCode() changed after save/read");
    #endif

```

```

/* Now we check that m2 actually holds that data. */

ensure(m2.equals(t), "Error (" + level + ", " + seed + "): ! m2.equals(t) after save/read");
ensure(t.equals(m2), "Error (" + level + ", " + seed + "): ! t.equals(m2) after save/read");
/* Now we take out of m everything, and check that it is empty. */

for(Iterator i=t.iterator(); i.hasNext();) m2.remove(i.next());

ensure(m2.isEmpty(), "Error (" + level + ", " + seed + "): m2 is not empty (as it should be)");
#endif

/* Now we play with iterators. */

{
  KEY_LIST_ITERATOR i;
  java.util.ListIterator j;
  Object J;
  i = m.listIterator();
  j = t.listIterator();

  for(int k = 0; k < 2*n; k++) {
    ensure(i.hasNext() == j.hasNext(), "Error (" + level + ", " + seed + "): divergence in hasNext()");
    ensure(i.hasPrevious() == j.hasPrevious(), "Error (" + level + ", " + seed + "): divergence in hasPrevious()");

    if (r.nextFloat() < .8 && i.hasNext()) {
      ensure(i.next().equals(J = j.next()), "Error (" + level + ", " + seed + "): divergence in next()");

      if (r.nextFloat() < 0.2) {
        i.remove();
        j.remove();
      }
      else if (r.nextFloat() < 0.2) {
        KEY_TYPE T = genKey();
        i.set(T);
        j.set(KEY2OBJ(T));
      }
      else if (r.nextFloat() < 0.2) {
        KEY_TYPE T = genKey();
        i.add(T);
        j.add(KEY2OBJ(T));
      }
      }
    else if (r.nextFloat() < .2 && i.hasPrevious()) {
      ensure(i.previous().equals(J = j.previous()), "Error (" + level + ", " + seed + "): divergence in previous()");

      if (r.nextFloat() < 0.2) {
        i.remove();

```

```

    j.remove();
  }
  else if (r.nextFloat() < 0.2) {
    KEY_TYPE T = genKey();
    i.set(T);
    j.set(KEY2OBJ(T));
  }
  else if (r.nextFloat() < 0.2) {
    KEY_TYPE T = genKey();
    i.add(T);
    j.add(KEY2OBJ(T));
  }
}

ensure(i.nextIndex() == j.nextIndex(), "Error (" + level + ", " + seed + "): divergence in nextIndex()");
ensure(i.previousIndex() == j.previousIndex(), "Error (" + level + ", " + seed + "): divergence in previousIndex()");

}

}

{
  Object previous = null;
  Object I, J;
  int from = r.nextInt(m.size() + 1);
  KEY_LIST_ITERATOR i;
  java.util.ListIterator j;
  i = m.listIterator(from);
  j = t.listIterator(from);

  for(int k = 0; k < 2*n; k++) {
    ensure(i.hasNext() == j.hasNext(), "Error (" + level + ", " + seed + "): divergence in hasNext() (iterator with starting
point " + from + ")");
    ensure(i.hasPrevious() == j.hasPrevious(), "Error (" + level + ", " + seed + "): divergence in hasPrevious() (iterator
with starting point " + from + ")");

    if (r.nextFloat() < .8 && i.hasNext()) {
      ensure((I = i.next()).equals(J = j.next()), "Error (" + level + ", " + seed + "): divergence in next() (" + I + ", " + J + ",
iterator with starting point " + from + ")");
      //System.err.println("Done next " + I + " " + J + " " + badPrevious);
    }

    if (r.nextFloat() < 0.2) {
      //System.err.println("Removing in next");
      i.remove();
      j.remove();
    }
    else if (r.nextFloat() < 0.2) {
      KEY_TYPE T = genKey();

```

```

    i.set(T);
    j.set(KEY2OBJ(T));
  }
  else if (r.nextFloat() < 0.2) {
    KEY_TYPE T = genKey();
    i.add(T);
    j.add(KEY2OBJ(T));
  }
}
else if (r.nextFloat() < .2 && i.hasPrevious()) {
  ensure((I = i.previous()).equals(J = j.previous()), "Error (" + level + ", " + seed + "): divergence in previous() (" + I +
", " + J + ", iterator with starting point " + from + ")");

  if (r.nextFloat() < 0.2) {
    //System.err.println("Removing in prev");
    i.remove();
    j.remove();
  }
  else if (r.nextFloat() < 0.2) {
    KEY_TYPE T = genKey();
    i.set(T);
    j.set(KEY2OBJ(T));
  }
  else if (r.nextFloat() < 0.2) {
    KEY_TYPE T = genKey();
    i.add(T);
    j.add(KEY2OBJ(T));
  }
}
}

/* Now we check that m actually holds that data. */

ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after iteration");
ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after iteration");

/* Now we select a pair of keys and create a subset. */

if (! m.isEmpty()) {
  int start = r.nextInt(m.size());
  int end = start + r.nextInt(m.size() - start);
  //System.err.println("Checking subList from " + start + " to " + end + " (level=" + (level+1) + ")...");
  testLists(m.subList(start, end), t.subList(start, end), n, level + 1);

  ensure(m.equals(t), "Error (" + level + ", " + seed + m + t + "): ! m.equals(t) after subList");
  ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after subList");
}

```

```

    }

    m.clear();
    t.clear();
    ensure(m.isEmpty(), "Error (" + level + ", " + seed + "): m is not empty after clear()");
}

protected static void runTest(int n) {
    ARRAY_LIST m = new ARRAY_LIST();
    java.util.ArrayList t = new java.util.ArrayList();
    topList = m;
    k = new Object[n];
    nk = new Object[n];
    kt = new KEY_TYPE[n];
    nkt = new KEY_TYPE[n];

    for(int i = 0; i < n; i++) {
#ifdef KEYS_REFERENCE
        k[i] = kt[i] = genKey();
        nk[i] = nkt[i] = genKey();
#else
        k[i] = new KEY_CLASS(kt[i] = genKey());
        nk[i] = new KEY_CLASS(nkt[i] = genKey());
#endif
    }

    /* We add pairs to t. */
    for(int i = 0; i < n; i++) t.add(k[i]);

    /* We add to m the same data */
    m.addAll(t);

    testLists(m, t, n, 0);

    System.out.println("Test OK");
    return;
}

public static void main(String args[]) {
    int n = Integer.parseInt(args[1]);
    if (args.length > 2) r = new java.util.Random(seed = Long.parseLong(args[2]));

    try {
        if ("speedTest".equals(args[0]) || "speedComp".equals(args[0])) speedTest(n, "speedComp".equals(args[0]));
    }
}

```

```
    else if ("test".equals(args[0])) runTest(n);
  } catch(Throwable e) {
    e.printStackTrace(System.err);
    System.err.println("seed: " + seed);
  }
}
```

```
#endif
```

```
}
```

Found in path(s):

```
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-
a775574/drv/ArrayList.drv
```

No license file was found, but licenses were detected in source scan.

```
/*
```

```
* Copyright (C) 2003-2017 Sebastiano Vigna
```

```
*
```

```
* Licensed under the Apache License, Version 2.0 (the "License");
```

```
* you may not use this file except in compliance with the License.
```

```
* You may obtain a copy of the License at
```

```
*
```

```
* http://www.apache.org/licenses/LICENSE-2.0
```

```
*
```

```
* Unless required by applicable law or agreed to in writing, software
```

```
* distributed under the License is distributed on an "AS IS" BASIS,
```

```
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
```

```
* See the License for the specific language governing permissions and
```

```
* limitations under the License.
```

```
*/
```

```
package PACKAGE;
```

```
/** A class providing static methods and objects that do useful things with type-specific priority queues.
```

```
*
```

```
* @see it.unimi.dsi.fastutil.PriorityQueue
```

```
*/
```

```
public final class PRIORITY_QUEUES {
```

```
    private PRIORITY_QUEUES() {}
```

```
    /** A synchronized wrapper class for priority queues. */
```

```
    public static class SynchronizedPriorityQueue KEY_GENERIC implements PRIORITY_QUEUE KEY_GENERIC
```

```
{
```

```

protected final PRIORITY_QUEUE KEY_GENERIC q;
protected final Object sync;

protected SynchronizedPriorityQueue(final PRIORITY_QUEUE KEY_GENERIC q, final Object sync) {
    this.q = q;
    this.sync = sync;
}

protected SynchronizedPriorityQueue(final PRIORITY_QUEUE KEY_GENERIC q) {
    this.q = q;
    this.sync = this;
}

@Override
public void enqueue(KEY_GENERIC_TYPE x) { synchronized(sync) { q.enqueue(x); } }

@Override
public KEY_GENERIC_TYPE DEQUEUE() { synchronized(sync) { return q.DEQUEUE(); } }

@Override
public KEY_GENERIC_TYPE FIRST() { synchronized(sync) { return q.FIRST(); } }

@Override
public KEY_GENERIC_TYPE LAST() { synchronized(sync) { return q.LAST(); } }

@Override
public boolean isEmpty() { synchronized(sync) { return q.isEmpty(); } }

@Override
public int size() { synchronized(sync) { return q.size(); } }

@Override
public void clear() { synchronized(sync) { q.clear(); } }

@Override
public void changed() { synchronized(sync) { q.changed(); } }

@Override
public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() { synchronized(sync) { return
q.comparator(); } }

#if KEYS_PRIMITIVE

@Deprecated
@Override
public void enqueue(KEY_CLASS x) { synchronized(sync) { q.enqueue(x); } }

@Deprecated

```

```

@Override
public KEY_CLASS dequeue() { synchronized(sync) { return q.dequeue(); } }

@Deprecated
@Override
public KEY_CLASS first() { synchronized(sync) { return q.first(); } }

@Deprecated
@Override
public KEY_CLASS last() { synchronized(sync) { return q.last(); } }

#endif
@Override
public int hashCode() { synchronized(sync) { return q.hashCode(); } }

@Override
public boolean equals(final Object o) { if (o == this) return true; synchronized(sync) { return q.equals(o); } }

private void writeObject(java.io.ObjectOutputStream s) throws java.io.IOException {
    synchronized(sync) { s.defaultWriteObject(); }
}

}

/** Returns a synchronized type-specific priority queue backed by the specified type-specific priority queue.
 *
 * @param q the priority queue to be wrapped in a synchronized priority queue.
 * @return a synchronized view of the specified priority queue.
 */
public static KEY_GENERIC PRIORITY_QUEUE KEY_GENERIC synchronize(final PRIORITY_QUEUE
KEY_GENERIC q) { return new SynchronizedPriorityQueue(q); }

/** Returns a synchronized type-specific priority queue backed by the specified type-specific priority queue, using
an assigned object to synchronize.
 *
 * @param q the priority queue to be wrapped in a synchronized priority queue.
 * @param sync an object that will be used to synchronize the access to the priority queue.
 * @return a synchronized view of the specified priority queue.
 */
public static KEY_GENERIC PRIORITY_QUEUE KEY_GENERIC synchronize(final PRIORITY_QUEUE
KEY_GENERIC q, final Object sync) { return new SynchronizedPriorityQueue(q, sync); }

}

```

Found in path(s):

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-

a775574/drv/PriorityQueues.drv

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2002-2017 Sebastiano Vigna

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

package PACKAGE;

/** An abstract class facilitating the creation of type-specific iterators.

*

* @deprecated As of fastutil 8 this class is no longer necessary, as its previous abstract

* methods are now default methods of the type-specific interface.

*/

@Deprecated

public abstract class KEY_ABSTRACT_ITERATOR KEY_GENERIC implements KEY_ITERATOR

KEY_GENERIC {

protected KEY_ABSTRACT_ITERATOR() {}

}

Found in path(s):

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-

a775574/drv/AbstractIterator.drv

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2003-2017 Paolo Boldi and Sebastiano Vigna

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

```
package PACKAGE;
```

```
import java.util.NoSuchElementException;
```

```
import it.unimi.dsi.fastutil.PriorityQueue;
```

```
/** A type-specific {@link PriorityQueue}; provides some additional methods that use polymorphism to avoid  
(un)boxing.
```

```
*
```

```
* <p>Additionally, this interface strengthens {@link #comparator()}.
```

```
*/
```

```
public interface PRIORITY_QUEUE extends PriorityQueue<KEY_CLASS> {
```

```
/** Enqueues a new element.
```

```
* @see PriorityQueue#enqueue(Object)
```

```
* @param x the element to enqueue.
```

```
*/
```

```
void enqueue(KEY_GENERIC_TYPE x);
```

```
/** Dequeues the {@linkplain #first() first} element from the queue.
```

```
* @see #dequeue()
```

```
* @return the dequeued element.
```

```
* @throws NoSuchElementException if the queue is empty.
```

```
*/
```

```
KEY_GENERIC_TYPE DEQUEUE();
```

```
/** Returns the first element of the queue.
```

```
* @see #first()
```

```
* @return the first element.
```

```
* @throws NoSuchElementException if the queue is empty.
```

```
*/
```

```
KEY_GENERIC_TYPE FIRST();
```

```
/** Returns the last element of the queue, that is, the element the would be dequeued last (optional operation).
```

```
* <p>This default implementation just throws an {@link UnsupportedOperationException}.
```

```
* @see #last()
```

```

* @return the last element.
* @throws NoSuchElementException if the queue is empty.
*/

default KEY_GENERIC_TYPE LAST() { throw new UnsupportedOperationException(); }

/** Returns the comparator associated with this priority queue, or null if it uses its elements' natural ordering.
*
* <p>Note that this specification strengthens the one given in { @link PriorityQueue#comparator() }.
* @see PriorityQueue#comparator()
* @return the comparator associated with this priority queue.
*/
@Override
KEY_COMPARATOR comparator();

/** { @inheritDoc }
* <p>This default implementation delegates to the corresponding type-specific method.
* @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default void enqueue(final KEY_GENERIC_CLASS x) { enqueue(x.KEY_VALUE()); }

/** { @inheritDoc }
* <p>This default implementation delegates to the corresponding type-specific method.
* @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default KEY_GENERIC_CLASS dequeue() { return KEY2OBJ(DEQUEUE()); }

/** { @inheritDoc }
* <p>This default implementation delegates to the corresponding type-specific method.
* @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default KEY_GENERIC_CLASS first() { return KEY2OBJ(FIRST()); }

/** { @inheritDoc }
* <p>This default implementation delegates to the corresponding type-specific method.
* @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default KEY_GENERIC_CLASS last() { return KEY2OBJ(LAST()); }
}

```

Found in path(s):

```

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-
a775574/drv/PriorityQueue.drv

```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2002-2017 Sebastiano Vigna
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

package PACKAGE;

```
/** An abstract class providing basic methods for implementing a type-specific stack interface.
 *
 * @deprecated As of fastutil 8 this class is no longer necessary, as its previous abstract
 * methods are now default methods of the type-specific interface.
 */
@Deprecated
public abstract class ABSTRACT_STACK_KEY_GENERIC extends
it.unimi.dsi.fastutil.AbstractStack<KEY_GENERIC_CLASS> implements STACK_KEY_GENERIC {
    protected ABSTRACT_STACK() {}
}
```

Found in path(s):

```
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-
a775574/drv/AbstractStack.drv
```

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (C) 2003-2017 Paolo Boldi and Sebastiano Vigna
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
```

- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.
- */

Found in path(s):

- * /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/src/it/unimi/dsi/fastutil/PriorityQueue.java
- * /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/src/it/unimi/dsi/fastutil/IndirectPriorityQueue.java
- * /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/src/it/unimi/dsi/fastutil/AbstractPriorityQueue.java
- * /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/src/it/unimi/dsi/fastutil/AbstractIndirectPriorityQueue.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2003-2017 Sebastiano Vigna

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

- * /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/src/it/unimi/dsi/fastutil/PriorityQueues.java
- * /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/src/it/unimi/dsi/fastutil/IndirectPriorityQueues.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2002-2017 Sebastiano Vigna

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

```
package PACKAGE;
```

```
import java.util.Collection;  
import java.util.Iterator;  
import java.util.RandomAccess;  
import java.util.NoSuchElementException;  
import it.unimi.dsi.fastutil.BigArrays;
```

```
#if KEYS_PRIMITIVE
```

```
/** A type-specific big list based on a big array; provides some additional methods that use polymorphism to avoid  
(un)boxing.
```

```
*
```

```
* <p>This class implements a lightweight, fast, open, optimized,  
* reuse-oriented version of big-array-based big lists. Instances of this class  
* represent a big list with a big array that is enlarged as needed when new entries  
* are created (by doubling the current length), but is  
* <em>never</em> made smaller (even on a { @link #clear()}). A family of  
* { @linkplain #trim() trimming methods} lets you control the size of the  
* backing big array; this is particularly useful if you reuse instances of this class.  
* Range checks are equivalent to those of { @link java.util}'s classes, but  
* they are delayed as much as possible. The backing big array is exposed by the  
* { @link #elements()} method.
```

```
*
```

```
* <p>This class implements the bulk methods { @code removeElements()},  
* { @code addElements()} and { @code getElements()} using  
* high-performance system calls (e.g., { @link  
* System#arraycopy(Object,int,Object,int,int) System.arraycopy()} instead of  
* expensive loops.
```

```
*
```

```
* @see java.util.ArrayList
```

```
*/
```

```
public class BIG_ARRAY_BIG_LIST KEY_GENERIC extends ABSTRACT_BIG_LIST KEY_GENERIC  
implements RandomAccess, Cloneable, java.io.Serializable {  
    private static final long serialVersionUID = -7046029254386353130L;
```

```
#else
```

```

/** A type-specific big-array-based big list; provides some additional methods that use polymorphism to avoid
(un)boxing.
*
* <p>This class implements a lightweight, fast, open, optimized,
* reuse-oriented version of big-array-based big lists. Instances of this class
* represent a big list with a big array that is enlarged as needed when new entries
* are created (by doubling the current length), but is
* <em>never</em> made smaller (even on a {@link #clear()}). A family of
* {@linkplain #trim() trimming methods} lets you control the size of the
* backing big array; this is particularly useful if you reuse instances of this class.
* Range checks are equivalent to those of {@link java.util}'s classes, but
* they are delayed as much as possible.
*
* <p>The backing big array is exposed by the {@link #elements()} method. If an instance
* of this class was created {@linkplain #wrap(Object[[[]],long) by wrapping},
* backing-array reallocations will be performed using reflection, so that
* {@link #elements()} can return a big array of the same type of the original big array; the comments
* about efficiency made in {@link it.unimi.dsi.fastutil.objects.ObjectArrays} apply here.
*
* <p>This class implements the bulk methods {@code removeElements()},
* {@code addElements()} and {@code getElements()} using
* high-performance system calls (e.g., {@link
* System#arraycopy(Object,int,Object,int,int) System.arraycopy()} instead of
* expensive loops.
*
* @see java.util.ArrayList
*/

```

```

public class BIG_ARRAY_BIG_LIST KEY_GENERIC extends ABSTRACT_BIG_LIST KEY_GENERIC
implements RandomAccess, Cloneable, java.io.Serializable {
    private static final long serialVersionUID = -7046029254386353131L;

```

```

#endif

```

```

/** The initial default capacity of a big-array big list. */
public static final int DEFAULT_INITIAL_CAPACITY = 10;

```

```

#if ! KEYS_PRIMITIVE

```

```

/** Whether the backing big array was passed to {@code wrap()}. In
* this case, we must reallocate with the same type of big array. */
protected final boolean wrapped;

```

```

#endif

```

```

/** The backing big array. */
protected transient KEY_GENERIC_TYPE a[][];

```

```

/** The current actual size of the big list (never greater than the backing-array length). */

```

protected long size;

```
/** Creates a new big-array big list using a given array.
 *
 * <p>This constructor is only meant to be used by the wrapping methods.
 *
 * @param a the big array that will be used to back this big-array big list.
 */
```

```
protected BIG_ARRAY_BIG_LIST(final KEY_GENERIC_TYPE a[], @SuppressWarnings("unused") boolean
dummy) {
    this.a = a;
    #if ! KEYS_PRIMITIVE
        this.wrapped = true;
    #endif
}
```

```
/** Creates a new big-array big list with given capacity.
 *
 * @param capacity the initial capacity of the array list (may be 0).
 */
```

```
SUPPRESS_WARNINGS_KEY_UNCHECKED
public BIG_ARRAY_BIG_LIST(final long capacity) {
    if (capacity < 0) throw new IllegalArgumentException("Initial capacity (" + capacity + ") is negative");
    if (capacity == 0) a = KEY_GENERIC_BIG_ARRAY_CAST BIG_ARRAYS.EMPTY_BIG_ARRAY;
    else a = KEY_GENERIC_BIG_ARRAY_CAST BIG_ARRAYS.newBigArray(capacity);
    #if ! KEYS_PRIMITIVE
        wrapped = false;
    #endif
}
```

```
/** Creates a new big-array big list with { @link #DEFAULT_INITIAL_CAPACITY } capacity. */
```

```
SUPPRESS_WARNINGS_KEY_UNCHECKED
public BIG_ARRAY_BIG_LIST() {
    a = KEY_GENERIC_BIG_ARRAY_CAST BIG_ARRAYS.DEFAULT_EMPTY_BIG_ARRAY; // We delay
allocation
    #if ! KEYS_PRIMITIVE
        wrapped = false;
    #endif
}
```

```
/** Creates a new big-array big list and fills it with a given type-specific collection.
 *
 * @param c a type-specific collection that will be used to fill the array list.
 */
```

```

public BIG_ARRAY_BIG_LIST(final COLLECTION KEY_EXTENDS_GENERIC c) {
    this(c.size());
    for(KEY_ITERATOR KEY_EXTENDS_GENERIC i = c.iterator(); i.hasNext();) add(i.NEXT_KEY());
}

/** Creates a new big-array big list and fills it with a given type-specific list.
 *
 * @param l a type-specific list that will be used to fill the array list.
 */

public BIG_ARRAY_BIG_LIST(final BIG_LIST KEY_EXTENDS_GENERIC l) {
    this(l.size64());
    l.getElements(0, a, 0, size = l.size64());
}

/** Creates a new big-array big list and fills it with the elements of a given big array.
 *
 * <p>Note that this constructor makes it easy to build big lists from literal arrays
 * declared as <code><var>type</var>[][] {{ <var>init_values</var> }}</code>.
 * The only constraint is that the number of initialisation values is
 * below { @link it.unimi.dsi.fastutil.BigArrays#SEGMENT_SIZE}.
 *
 * @param a a big array whose elements will be used to fill the array list.
 */

public BIG_ARRAY_BIG_LIST(final KEY_GENERIC_TYPE a[][]) {
    this(a, 0, BIG_ARRAYS.length(a));
}

/** Creates a new big-array big list and fills it with the elements of a given big array.
 *
 * <p>Note that this constructor makes it easy to build big lists from literal arrays
 * declared as <code><var>type</var>[][] {{ <var>init_values</var> }}</code>.
 * The only constraint is that the number of initialisation values is
 * below { @link it.unimi.dsi.fastutil.BigArrays#SEGMENT_SIZE}.
 *
 * @param a a big array whose elements will be used to fill the array list.
 * @param offset the first element to use.
 * @param length the number of elements to use.
 */

public BIG_ARRAY_BIG_LIST(final KEY_GENERIC_TYPE a[][], final long offset, final long length) {
    this(length);
    BIG_ARRAYS.copy(a, offset, this.a, 0, length);
    size = length;
}

/** Creates a new big-array big list and fills it with the elements returned by an iterator..

```

```

*
* @param i an iterator whose returned elements will fill the array list.
*/

public BIG_ARRAY_BIG_LIST(final Iterator<? extends KEY_GENERIC_CLASS> i) {
    this();
    while(i.hasNext()) this.add(KEY_CLASS2TYPE(i.next()));
}

/** Creates a new big-array big list and fills it with the elements returned by a type-specific iterator..
*
* @param i a type-specific iterator whose returned elements will fill the array list.
*/

public BIG_ARRAY_BIG_LIST(final KEY_ITERATOR KEY_EXTENDS_GENERIC i) {
    this();
    while(i.hasNext()) this.add(i.NEXT_KEY());
}

#if KEYS_PRIMITIVE
/** Returns the backing big array of this big list.
*
* @return the backing big array.
*/

public KEY_GENERIC_TYPE[][] elements() {
    return a;
}
#else
/** Returns the backing big array of this big list.
*
* <p>If this big-array big list was created by wrapping a given big array, it is guaranteed
* that the type of the returned big array will be the same. Otherwise, the returned
* big array will be an big array of objects.
*
* @return the backing big array.
*/

public KEY_GENERIC_TYPE[][] elements() {
    return a;
}
#endif

/** Wraps a given big array into a big-array list of given size.
*
* @param a a big array to wrap.
* @param length the length of the resulting big-array list.
* @return a new big-array list of the given size, wrapping the given big array.

```

```

*/

public static KEY_GENERIC BIG_ARRAY_BIG_LIST KEY_GENERIC wrap(final KEY_GENERIC_TYPE a[],
final long length) {
    if (length > BIG_ARRAYS.length(a)) throw new IllegalArgumentException("The specified length (" + length + ")
is greater than the array size (" + BIG_ARRAYS.length(a) + ")");
    final BIG_ARRAY_BIG_LIST KEY_GENERIC l = new BIG_ARRAY_BIG_LIST
KEY_GENERIC_DIAMOND(a, false);
    l.size = length;
    return l;
}

/** Wraps a given big array into a big-array big list.
*
* @param a a big array to wrap.
* @return a new big-array big list wrapping the given array.
*/

public static KEY_GENERIC BIG_ARRAY_BIG_LIST KEY_GENERIC wrap(final KEY_GENERIC_TYPE
a[]) {
    return wrap(a, BIG_ARRAYS.length(a));
}

/** Ensures that this big-array big list can contain the given number of entries without resizing.
*
* @param capacity the new minimum capacity for this big-array big list.
*/
SUPPRESS_WARNINGS_KEY_UNCHECKED
public void ensureCapacity(final long capacity) {
    if (capacity <= a.length || a == BIG_ARRAYS.DEFAULT_EMPTY_BIG_ARRAY) return;
#ifdef KEYS_PRIMITIVE
    a = BIG_ARRAYS.forceCapacity(a, capacity, size);
#else
    if (wrapped) a = BIG_ARRAYS.forceCapacity(a, capacity, size);
    else {
        if (capacity > BIG_ARRAYS.length(a)) {
            final Object t[] = BIG_ARRAYS.newBigArray(capacity);
            BIG_ARRAYS.copy(a, 0, t, 0, size);
            a = (KEY_GENERIC_TYPE[][]);t;
        }
    }
#endif
    assert size <= BIG_ARRAYS.length(a);
}

/** Grows this big-array big list, ensuring that it can contain the given number of entries without resizing,
* and in case increasing current capacity at least by a factor of 50%.

```

```

*
* @param capacity the new minimum capacity for this big-array big list.
*/
SUPPRESS_WARNINGS_KEY_UNCHECKED
private void grow(long capacity) {
    final long oldLength = BIG_ARRAYS.length(a);
    if (capacity <= oldLength) return;
    if (a != BIG_ARRAYS.DEFAULT_EMPTY_BIG_ARRAY) capacity = Math.max(oldLength + (oldLength >> 1),
capacity);
    else if (capacity < DEFAULT_INITIAL_CAPACITY) capacity = DEFAULT_INITIAL_CAPACITY;
#if KEYS_PRIMITIVE
    a = BIG_ARRAYS.forceCapacity(a, capacity, size);
#else
    if (wrapped) a = BIG_ARRAYS.forceCapacity(a, capacity, size);
    else {
        final Object t[][] = BIG_ARRAYS.newBigArray(capacity);
        BIG_ARRAYS.copy(a, 0, t, 0, size);
        a = (KEY_GENERIC_TYPE[][])t;
    }
#endif
    assert size <= BIG_ARRAYS.length(a);
}

@Override
public void add(final long index, final KEY_GENERIC_TYPE k) {
    ensureIndex(index);
    grow(size + 1);
    if (index != size) BIG_ARRAYS.copy(a, index, a, index + 1, size - index);
    BIG_ARRAYS.set(a, index, k);
    size++;
    assert size <= BIG_ARRAYS.length(a);
}

@Override
public boolean add(final KEY_GENERIC_TYPE k) {
    grow(size + 1);
    BIG_ARRAYS.set(a, size++, k);
    assert size <= BIG_ARRAYS.length(a);
    return true;
}

@Override
public KEY_GENERIC_TYPE GET_KEY(final long index) {
    if (index >= size) throw new IndexOutOfBoundsException("Index (" + index + ") is greater than or equal to list size (" + size + ")");
    return BIG_ARRAYS.get(a, index);
}

```

```

@Override
public long indexOf(final KEY_TYPE k) {
    for(long i = 0; i < size; i++) if (KEY_EQUALS(k, BIG_ARRAYS.get(a, i))) return i;
    return -1;
}

@Override
public long lastIndexOf(final KEY_TYPE k) {
    for(long i = size; i-- != 0;) if (KEY_EQUALS(k, BIG_ARRAYS.get(a, i))) return i;
    return -1;
}

@Override
public KEY_GENERIC_TYPE REMOVE_KEY(final long index) {
    if (index >= size) throw new IndexOutOfBoundsException("Index (" + index + ") is greater than or equal to list size (" + size + ")");
    final KEY_GENERIC_TYPE old = BIG_ARRAYS.get(a, index);
    size--;
    if (index != size) BIG_ARRAYS.copy(a, index + 1, a, index, size - index);
    #if KEYS_REFERENCE
    BIG_ARRAYS.set(a, size, null);
    #endif
    assert size <= BIG_ARRAYS.length(a);
    return old;
}

@Override
public boolean REMOVE(final KEY_TYPE k) {
    final long index = indexOf(k);
    if (index == -1) return false;
    REMOVE_KEY(index);
    assert size <= BIG_ARRAYS.length(a);
    return true;
}

@Override
public KEY_GENERIC_TYPE set(final long index, final KEY_GENERIC_TYPE k) {
    if (index >= size) throw new IndexOutOfBoundsException("Index (" + index + ") is greater than or equal to list size (" + size + ")");
    KEY_GENERIC_TYPE old = BIG_ARRAYS.get(a, index);
    BIG_ARRAYS.set(a, index, k);
    return old;
}

#if KEYS_PRIMITIVE
@Override
public boolean removeAll(final COLLECTION c) {
    KEY_GENERIC_TYPE[] s = null, d = null;

```

```

int ss = -1, sd = BigArrays.SEGMENT_SIZE, ds = -1, dd = BigArrays.SEGMENT_SIZE;
for (long i = 0; i < size; i++) {
    if (sd == BigArrays.SEGMENT_SIZE) {
        sd = 0;
        s = a[++ss];
    }
    if (!c.contains(s[sd])) {
        if (dd == BigArrays.SEGMENT_SIZE) {
            d = a[++ds];
            dd = 0;
        }
        d[dd++] = s[sd];
    }
    sd++;
}
final long j = BigArrays.index(ds, dd);
#if KEYS_REFERENCE
    BIG_ARRAYS.fill(a, j, size, null);
#endif
final boolean modified = size != j;
size = j;
return modified;
}

```

#endif

@Override

```

public boolean removeAll(final Collection<?> c) {
    KEY_GENERIC_TYPE[] s = null, d = null;
    int ss = -1, sd = BigArrays.SEGMENT_SIZE, ds = -1, dd = BigArrays.SEGMENT_SIZE;
    for (long i = 0; i < size; i++) {
        if (sd == BigArrays.SEGMENT_SIZE) {
            sd = 0;
            s = a[++ss];
        }
        if (!c.contains(KEY2OBJ(s[sd]))) {
            if (dd == BigArrays.SEGMENT_SIZE) {
                d = a[++ds];
                dd = 0;
            }
            d[dd++] = s[sd];
        }
        sd++;
    }
    final long j = BigArrays.index(ds, dd);
#if KEYS_REFERENCE
    BIG_ARRAYS.fill(a, j, size, null);
#endif
}

```

```

    final boolean modified = size != j;
    size = j;
    return modified;
}

@Override
public void clear() {
    #if KEYS_REFERENCE
        BIG_ARRAYS.fill(a, 0, size, null);
    #endif
    size = 0;
    assert size <= BIG_ARRAYS.length(a);
}

@Override
public long size64() {
    return size;
}

@Override
public void size(final long size) {
    if (size > BIG_ARRAYS.length(a)) a = BIG_ARRAYS.forceCapacity(a, size, this.size);
    if (size > this.size) BIG_ARRAYS.fill(a, this.size, size, KEY_NULL);
    #if KEYS_REFERENCE
        else BIG_ARRAYS.fill(a, size, this.size, KEY_NULL);
    #endif
    this.size = size;
}

@Override
public boolean isEmpty() {
    return size == 0;
}

/** Trims this big-array big list so that the capacity is equal to the size.
 *
 * @see java.util.ArrayList#trimToSize()
 */
public void trim() {
    trim(0);
}

/** Trims the backing big array if it is too large.
 *
 * If the current big array length is smaller than or equal to
 * { @code n }, this method does nothing. Otherwise, it trims the
 * big-array length to the maximum between { @code n } and { @link #size64() }.
 */

```

```

* <p>This method is useful when reusing big lists. { @linkplain #clear() Clearing a
* big list} leaves the big-array length untouched. If you are reusing a big list
* many times, you can call this method with a typical
* size to avoid keeping around a very large big array just
* because of a few large transient big lists.
*
* @param n the threshold for the trimming.
*/

```

```

public void trim(final long n) {
    final long arrayLength = BIG_ARRAYS.length(a);
    if (n >= arrayLength || size == arrayLength) return;
    a = BIG_ARRAYS.trim(a, Math.max(n, size));
    assert size <= BIG_ARRAYS.length(a);
}

```

```

/** Copies element of this type-specific list into the given big array using optimized system calls.
*
* @param from the start index (inclusive).
* @param a the destination big array.
* @param offset the offset into the destination array where to store the first element copied.
* @param length the number of elements to be copied.
*/

```

```

@Override
public void getElements(final long from, final KEY_TYPE[][] a, final long offset, final long length) {
    BIG_ARRAYS.copy(this.a, from, a, offset, length);
}

```

```

/** Removes elements of this type-specific list using optimized system calls.
*
* @param from the start index (inclusive).
* @param to the end index (exclusive).
*/

```

```

@Override
public void removeElements(final long from, final long to) {
    BigArrays.ensureFromTo(size, from, to);
    BIG_ARRAYS.copy(a, to, a, from, size - to);
    size -= (to - from);
#ifdef KEYS_REFERENCE
    BIG_ARRAYS.fill(a, size, size + to - from, null);
#endif
}

```

```

/** Adds elements to this type-specific list using optimized system calls.
*

```

```

* @param index the index at which to add elements.
* @param a the big array containing the elements.
* @param offset the offset of the first element to add.
* @param length the number of elements to add.
*/
@Override
public void addElements(final long index, final KEY_GENERIC_TYPE a[], final long offset, final long length) {
    ensureIndex(index);
    BIG_ARRAYS.ensureOffsetLength(a, offset, length);
    grow(size + length);
    BIG_ARRAYS.copy(this.a, index, this.a, index + length, size - index);
    BIG_ARRAYS.copy(a, offset, this.a, index, length);
    size += length;
}

@Override
public KEY_BIG_LIST_ITERATOR KEY_GENERIC listIterator(final long index) {
    ensureIndex(index);

    return new KEY_BIG_LIST_ITERATOR KEY_GENERIC() {
        long pos = index, last = -1;

        @Override
        public boolean hasNext() { return pos < size; }
        @Override
        public boolean hasPrevious() { return pos > 0; }
        @Override
        public KEY_GENERIC_TYPE NEXT_KEY() { if (! hasNext()) throw new NoSuchElementException(); return
BIG_ARRAYS.get(a, last = pos++); }
        @Override
        public KEY_GENERIC_TYPE PREV_KEY() { if (! hasPrevious()) throw new NoSuchElementException(); return
BIG_ARRAYS.get(a, last = --pos); }
        @Override
        public long nextIndex() { return pos; }
        @Override
        public long previousIndex() { return pos - 1; }
        @Override
        public void add(KEY_GENERIC_TYPE k) {
            BIG_ARRAY_BIG_LIST.this.add(pos++, k);
            last = -1;
        }
        @Override
        public void set(KEY_GENERIC_TYPE k) {
            if (last == -1) throw new IllegalStateException();
            BIG_ARRAY_BIG_LIST.this.set(last, k);
        }
        @Override
        public void remove() {

```

```

        if (last == -1) throw new IllegalStateException();
        BIG_ARRAY_BIG_LIST.this.REMOVE_KEY(last);
        /* If the last operation was a next(), we are removing an element *before* us, and we must decrease pos
correspondingly. */
        if (last < pos) pos--;
        last = -1;
    }
};
}

@Override
public BIG_ARRAY_BIG_LIST KEY_GENERIC clone() {
    BIG_ARRAY_BIG_LIST KEY_GENERIC c = new BIG_ARRAY_BIG_LIST
KEY_GENERIC_DIAMOND(size);
    BIG_ARRAYS.copy(a, 0, c.a, 0, size);
    c.size = size;
    return c;
}

#if KEY_CLASS_Object
private boolean valEquals(final K a, final K b) {
    return a == null ? b == null : a.equals(b);
}
#endif

/** Compares this type-specific big-array list to another one.
 *
 * <p>This method exists only for sake of efficiency. The implementation
 * inherited from the abstract implementation would already work.
 *
 * @param l a type-specific big-array list.
 * @return true if the argument contains the same elements of this type-specific big-array list.
 */
public boolean equals(final BIG_ARRAY_BIG_LIST KEY_GENERIC l) {
    if (l == this) return true;
    long s = size64();
    if (s != l.size64()) return false;
    final KEY_GENERIC_TYPE[][] a1 = a;
    final KEY_GENERIC_TYPE[][] a2 = l.a;

#if KEY_CLASS_Object
    while(s-- != 0) if (! valEquals(BIG_ARRAYS.get(a1, s), BIG_ARRAYS.get(a2, s))) return false;
#else
    while(s-- != 0) if (BIG_ARRAYS.get(a1, s) != BIG_ARRAYS.get(a2, s)) return false;
#endif
    return true;
}

```

```

#if ! KEY_CLASS_Reference

/** Compares this big list to another big list.
 *
 * <p>This method exists only for sake of efficiency. The implementation
 * inherited from the abstract implementation would already work.
 *
 * @param l a big list.
 * @return a negative integer,
 * zero, or a positive integer as this big list is lexicographically less than, equal
 * to, or greater than the argument.
 */
SUPPRESS_WARNINGS_KEY_UNCHECKED
public int compareTo(final BIG_ARRAY_BIG_LIST KEY_EXTENDS_GENERIC l) {
    final long s1 = size64(), s2 = l.size64();
    final KEY_GENERIC_TYPE a1[][] = a, a2[][] = l.a;
    KEY_GENERIC_TYPE e1, e2;
    int r, i;

    for(i = 0; i < s1 && i < s2; i++) {
        e1 = BIG_ARRAYS.get(a1, i);
        e2 = BIG_ARRAYS.get(a2, i);
        if ((r = KEY_CMP(e1, e2)) != 0) return r;
    }

    return i < s2 ? -1 : (i < s1 ? 1 : 0);
}
#endif

private void writeObject(java.io.ObjectOutputStream s) throws java.io.IOException {
    s.defaultWriteObject();
    for(int i = 0; i < size; i++) s.WRITE_KEY(BIG_ARRAYS.get(a, i));
}

SUPPRESS_WARNINGS_KEY_UNCHECKED
private void readObject(java.io.ObjectInputStream s) throws java.io.IOException, ClassNotFoundException {
    s.defaultReadObject();
    a = KEY_GENERIC_BIG_ARRAY_CAST BIG_ARRAYS.newBigArray(size);
    for(int i = 0; i < size; i++) BIG_ARRAYS.set(a, i, KEY_GENERIC_CAST s.READ_KEY());
}

#ifdef TEST

private static long seed = System.currentTimeMillis();
private static java.util.Random r = new java.util.Random(seed);

```

```

private static KEY_TYPE genKey() {
#if KEY_CLASS_Byte || KEY_CLASS_Short || KEY_CLASS_Character
    return (KEY_TYPE)(r.nextInt());
#elif KEYS_PRIMITIVE
    return r.NEXT_KEY();
#elif KEY_CLASS_Object
    return Integer.toBinaryString(r.nextInt());
#else
    return new java.io.Serializable() {};
#endif
}

private static java.text.NumberFormat format = new java.text.DecimalFormat("#,###.00");
private static java.text.FieldPosition p = new java.text.FieldPosition(0);

private static String format(double d) {
    StringBuffer s = new StringBuffer();
    return format.format(d, s, p).toString();
}

private static void speedTest(int n, boolean comp) {
    System.out.println("There are presently no speed tests for this class.");
}

private static void fatal(String msg) {
    System.out.println(msg);
    System.exit(1);
}

private static void ensure(boolean cond, String msg) {
    if (cond) return;
    fatal(msg);
}

private static Object[] k, v, nk;
private static KEY_TYPE kt[];
private static KEY_TYPE nkt[];
private static BIG_ARRAY_BIG_LIST topList;

protected static void testLists(BIG_LIST m, BIG_LIST t, int n, int level) {
    long ms;
    Exception mThrowsIllegal, tThrowsIllegal, mThrowsOutOfBounds, tThrowsOutOfBounds;
    Object rt = null;
    KEY_TYPE rm = KEY_NULL;

    if (level > 4) return;

```

```

/* Now we check that both sets agree on random keys. For m we use the polymorphic method. */

for(int i = 0; i < n; i++) {
    int p = r.nextInt() % (n * 2);

    KEY_TYPE T = genKey();
    mThrowsOutOfBounds = tThrowsOutOfBounds = null;

    try {
        m.set(p, T);
    }
    catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }
    try {
        t.set(p, KEY2OBJ(T));
    }
    catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }

    ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + level + ", " + seed + "):
set() divergence at start in IndexOutOfBoundsException for index " + p + " (" + mThrowsOutOfBounds + ", " +
tThrowsOutOfBounds + ")");
    if (mThrowsOutOfBounds == null) ensure(t.get(p).equals(KEY2OBJ(m.GET_KEY(p))), "Error (" + level + ", " +
seed + "): m and t differ after set() on position " + p + " (" + m.GET_KEY(p) + ", " + t.get(p) + ")");

    p = r.nextInt() % (n * 2);
    mThrowsOutOfBounds = tThrowsOutOfBounds = null;

    try {
        m.GET_KEY(p);
    }
    catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }
    try {
        t.get(p);
    }
    catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }

    ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + level + ", " + seed + "):
get() divergence at start in IndexOutOfBoundsException for index " + p + " (" + mThrowsOutOfBounds + ", " +
tThrowsOutOfBounds + ")");
    if (mThrowsOutOfBounds == null) ensure(t.get(p).equals(KEY2OBJ(m.GET_KEY(p))), "Error (" + level + ", " +
seed + "): m and t differ aftr get() on position " + p + " (" + m.GET_KEY(p) + ", " + t.get(p) + ")");

}

/* Now we check that both sets agree on random keys. For m we use the standard method. */

for(int i = 0; i < n; i++) {

```

```

int p = r.nextInt() % (n * 2);

mThrowsOutOfBounds = tThrowsOutOfBounds = null;

try {
    m.get(p);
}
catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }
try {
    t.get(p);
}
catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }

ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + level + ", " + seed + "):
get() divergence at start in IndexOutOfBoundsException for index " + p + " (" + mThrowsOutOfBounds + ", " +
tThrowsOutOfBounds + ")");
if (mThrowsOutOfBounds == null) ensure(t.get(p).equals(m.get(p)), "Error (" + level + ", " + seed + "): m and t
differ at start on position " + p + " (" + m.get(p) + ", " + t.get(p) + ")");

}

/* Now we check that m and t are equal. */
if (!m.equals(t) || !t.equals(m)) System.err.println("m: " + m + " t: " + t);

ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) at start");
ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) at start");

/* Now we check that m actually holds that data. */
for(Iterator i=t.iterator(); i.hasNext();) {
    ensure(m.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after insertion (iterating on
t)");
}

/* Now we check that m actually holds that data, but iterating on m. */
for(Iterator i=m.listIterator(); i.hasNext();) {
    ensure(t.contains(i.next()), "Error (" + level + ", " + seed + "): m and t differ on an entry after insertion (iterating on
m)");
}

/* Now we check that inquiries about random data give the same answer in m and t. For
m we use the polymorphic method. */

for(int i=0; i<n; i++) {
    KEY_TYPE T = genKey();
    ensure(m.contains(T) == t.contains(KEY2OBJ(T)), "Error (" + level + ", " + seed + "): divergence in content
between t and m (polymorphic method)");
}

```

```

}

/* Again, we check that inquiries about random data give the same answer in m and t, but
   for m we use the standard method. */

for(int i=0; i<n; i++) {
    KEY_TYPE T = genKey();
    ensure(m.contains(KEY2OBJ(T)) == t.contains(KEY2OBJ(T)), "Error (" + level + ", " + seed + "): divergence in
content between t and m (polymorphic method)");
}

/* Now we add and remove random data in m and t, checking that the result is the same. */

for(int i=0; i<2*n; i++) {
    KEY_TYPE T = genKey();

    try {
        m.add(T);
    }
    catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }

    try {
        t.add(KEY2OBJ(T));
    }
    catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }

    T = genKey();
    int p = r.nextInt() % (2 * n + 1);

    mThrowsOutOfBounds = tThrowsOutOfBounds = null;

    try {
        m.add(p, T);
    }
    catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }

    try {
        t.add(p, KEY2OBJ(T));
    }
    catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }

    ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + level + ", " + seed + "):
add() divergence in IndexOutOfBoundsException for index " + p + " for " + T + " (" + mThrowsOutOfBounds + ", "
+ tThrowsOutOfBounds + ")");

    p = r.nextInt() % (2 * n + 1);

```

```

mThrowsOutOfBounds = tThrowsOutOfBounds = null;

try {
    rm = m.REMOVE_KEY(p);
}
catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }

try {
    rt = t.remove(p);
}
catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }

ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + level + ", " + seed + "):
remove() divergence in IndexOutOfBoundsException for index " + p + " (" + mThrowsOutOfBounds + ", " +
tThrowsOutOfBounds + ")");
if (mThrowsOutOfBounds == null) ensure(rt.equals(KEY2OBJ(rm)), "Error (" + level + ", " + seed + "): divergence
in remove() between t and m (" + rt + ", " + rm + ")");
}

ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after add/remove");
ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after add/remove");

/* Now we add random data in m and t using addAll on a collection, checking that the result is the same. */

for(int i=0; i<n; i++) {
    int p = r.nextInt() % (2 * n + 1);
    java.util.Collection m1 = new java.util.ArrayList();
    int s = r.nextInt(n / 2 + 1);
    for(int j = 0; j < s; j++) m1.add(KEY2OBJ(genKey()));

    mThrowsOutOfBounds = tThrowsOutOfBounds = null;

    try {
        m.addAll(p, m1);
    }
    catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }

    try {
        t.addAll(p, m1);
    }
    catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }

    ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + level + ", " + seed + "):
addAll() divergence in IndexOutOfBoundsException for index " + p + " for " + m1 + " (" + mThrowsOutOfBounds
+ ", " + tThrowsOutOfBounds + ")");

    ensure(m.equals(t), "Error (" + level + ", " + seed + m + t + "): ! m.equals(t) after addAll");

```

```

ensure(t.equals(m), "Error (" + level + ", " + seed + m + t + "): ! t.equals(m) after addAll");
}

if (m.size64() > n) {
m.size(n);
while(t.size64() != n) t.remove(t.size64() -1);
}

/* Now we add random data in m and t using addAll on a type-specific collection, checking that the result is the
same. */

for(int i=0; i<n; i++) {
int p = r.nextInt() % (2 * n + 1);
COLLECTION m1 = new BIG_ARRAY_BIG_LIST();
java.util.Collection t1 = new java.util.ArrayList();
int s = r.nextInt(n / 2 + 1);
for(int j = 0; j < s; j++) {
KEY_TYPE x = genKey();
m1.add(x);
t1.add(KEY2OBJ(x));
}

mThrowsOutOfBounds = tThrowsOutOfBounds = null;

try {
m.addAll(p, m1);
}
catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }

try {
t.addAll(p, t1);
}
catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }

ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + level + ", " + seed + "):
polymorphic addAll() divergence in IndexOutOfBoundsException for index " + p + " for " + m1 + " (" +
mThrowsOutOfBounds + ", " + tThrowsOutOfBounds + ")");

ensure(m.equals(t), "Error (" + level + ", " + seed + m + t + "): ! m.equals(t) after polymorphic addAll");
ensure(t.equals(m), "Error (" + level + ", " + seed + m + t + "): ! t.equals(m) after polymorphic addAll");
}

if (m.size64() > n) {
m.size(n);
while(t.size64() != n) t.remove(t.size64() -1);
}

/* Now we add random data in m and t using addAll on a list, checking that the result is the same. */

```

```

for(int i=0; i<n; i++) {
    int p = r.nextInt() % (2 * n + 1);
    BIG_LIST m1 = new BIG_ARRAY_BIG_LIST();
    java.util.Collection t1 = new java.util.ArrayList();
    int s = r.nextInt(n / 2 + 1);
    for(int j = 0; j < s; j++) {
        KEY_TYPE x = genKey();
        m1.add(x);
        t1.add(KEY2OBJ(x));
    }

    mThrowsOutOfBounds = tThrowsOutOfBounds = null;

    try {
        m.addAll(p, m1);
    }
    catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }

    try {
        t.addAll(p, t1);
    }
    catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }

    ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + level + ", " + seed + "): list
addAll() divergence in IndexOutOfBoundsException for index " + p + " for " + m1 + " (" + mThrowsOutOfBounds
+ ", " + tThrowsOutOfBounds + ")");

    ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after list addAll");
    ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after list addAll");
}

/* Now we check that both sets agree on random keys. For m we use the standard method. */

for(int i = 0; i < n; i++) {
    int p = r.nextInt() % (n * 2);

    mThrowsOutOfBounds = tThrowsOutOfBounds = null;

    try {
        m.get(p);
    }
    catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }
    try {
        t.get(p);
    }
    catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }
}

```

```

    ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + level + ", " + seed + "):
get() divergence in IndexOutOfBoundsException for index " + p + " (" + mThrowsOutOfBounds + ", " +
tThrowsOutOfBounds + ")");
    if (mThrowsOutOfBounds == null) ensure(t.get(p).equals(m.get(p)), "Error (" + level + ", " + seed + "): m and t
differ on position " + p + " (" + m.get(p) + ", " + t.get(p) + ")");

}

/* Now we inquiry about the content with indexOf()/lastIndexOf(). */

for(int i=0; i<10*n; i++) {
    KEY_TYPE T = genKey();
    ensure(m.indexOf(KEY2OBJ(T)) == t.indexOf(KEY2OBJ(T)),
        "Error (" + level + ", " + seed + "): indexOf() divergence for " + T + " (" + m.indexOf(KEY2OBJ(T)) + ", " +
t.indexOf(KEY2OBJ(T)) + ")");
    ensure(m.lastIndexOf(KEY2OBJ(T)) == t.lastIndexOf(KEY2OBJ(T)),
        "Error (" + level + ", " + seed + "): lastIndexOf() divergence for " + T + " (" + m.lastIndexOf(KEY2OBJ(T)) + ", "
+ t.lastIndexOf(KEY2OBJ(T)) + ")");
    ensure(m.indexOf(T) == t.indexOf(KEY2OBJ(T)),
        "Error (" + level + ", " + seed + "): polymorphic indexOf() divergence for " + T + " (" + m.indexOf(T) + ", " +
t.indexOf(KEY2OBJ(T)) + ")");
    ensure(m.lastIndexOf(T) == t.lastIndexOf(KEY2OBJ(T)),
        "Error (" + level + ", " + seed + "): polymorphic lastIndexOf() divergence for " + T + " (" + m.lastIndexOf(T) + ", "
+ t.lastIndexOf(KEY2OBJ(T)) + ")");
}

/* Now we check cloning. */

if (level == 0) {
    ensure(m.equals(((BIG_ARRAY_BIG_LIST)m).clone()), "Error (" + level + ", " + seed + "): m does not equal
m.clone()");
    ensure(((BIG_ARRAY_BIG_LIST)m).clone().equals(m), "Error (" + level + ", " + seed + "): m.clone() does not
equal m");
}

/* Now we play with constructors. */
ensure(m.equals(new BIG_ARRAY_BIG_LIST((COLLECTION)m)), "Error (" + level + ", " + seed + "): m does
not equal new (type-specific Collection m)");
ensure((new BIG_ARRAY_BIG_LIST((COLLECTION)m)).equals(m), "Error (" + level + ", " + seed + "): new
(type-specific nCollection m) does not equal m");
ensure(m.equals(new BIG_ARRAY_BIG_LIST((BIG_LIST)m)), "Error (" + level + ", " + seed + "): m does not
equal new (type-specific List m)");
ensure((new BIG_ARRAY_BIG_LIST((BIG_LIST)m)).equals(m), "Error (" + level + ", " + seed + "): new (type-
specific List m) does not equal m");
ensure(m.equals(new BIG_ARRAY_BIG_LIST(m.listIterator())), "Error (" + level + ", " + seed + "): m does not
equal new (m.listIterator())");
ensure((new BIG_ARRAY_BIG_LIST(m.listIterator())).equals(m), "Error (" + level + ", " + seed + "): new
(m.listIterator()) does not equal m");

```

```

    ensure(m.equals(new BIG_ARRAY_BIG_LIST(m.iterator())), "Error (" + level + ", " + seed + "): m does not equal
new (m.type_specific_iterator())");
    ensure((new BIG_ARRAY_BIG_LIST(m.iterator())).equals(m), "Error (" + level + ", " + seed + "): new
(m.type_specific_iterator()) does not equal m");

    int h = m.hashCode();

    /* Now we save and read m. */

    BIG_LIST m2 = null;

    try {
        java.io.File ff = new java.io.File("it.unimi.dsi.fastutil.test");
        java.io.OutputStream os = new java.io.FileOutputStream(ff);
        java.io.ObjectOutputStream oos = new java.io.ObjectOutputStream(os);

        oos.writeObject(m);
        oos.close();

        java.io.InputStream is = new java.io.FileInputStream(ff);
        java.io.ObjectInputStream ois = new java.io.ObjectInputStream(is);

        m2 = (BIG_LIST)ois.readObject();
        ois.close();
        ff.delete();
    }
    catch(Exception e) {
        e.printStackTrace();
        System.exit(1);
    }

    #if ! KEY_CLASS_Reference
        ensure(m2.hashCode() == h, "Error (" + level + ", " + seed + "): hashCode() changed after save/read");

        /* Now we check that m2 actually holds that data. */

        ensure(m2.equals(t), "Error (" + level + ", " + seed + "): ! m2.equals(t) after save/read");
        ensure(t.equals(m2), "Error (" + level + ", " + seed + "): ! t.equals(m2) after save/read");
        /* Now we take out of m everything, and check that it is empty. */

        for(Iterator i=t.iterator(); i.hasNext();) m2.remove(i.next());

        ensure(m2.isEmpty(), "Error (" + level + ", " + seed + "): m2 is not empty (as it should be)");
    #endif

    /* Now we play with iterators. */

```

```

{
KEY_BIG_LIST_ITERATOR i;
KEY_BIG_LIST_ITERATOR j;
Object J;
i = m.listIterator();
j = t.listIterator();

for(int k = 0; k < 2*n; k++) {
ensure(i.hasNext() == j.hasNext(), "Error (" + level + ", " + seed + "): divergence in hasNext()");
ensure(i.hasPrevious() == j.hasPrevious(), "Error (" + level + ", " + seed + "): divergence in hasPrevious()");

if (r.nextFloat() < .8 && i.hasNext()) {
ensure(i.next().equals(J = j.next()), "Error (" + level + ", " + seed + "): divergence in next()");

if (r.nextFloat() < 0.2) {
i.remove();
j.remove();
}
else if (r.nextFloat() < 0.2) {
KEY_TYPE T = genKey();
i.set(T);
j.set(KEY2OBJ(T));
}
else if (r.nextFloat() < 0.2) {
KEY_TYPE T = genKey();
i.add(T);
j.add(KEY2OBJ(T));
}
}
else if (r.nextFloat() < .2 && i.hasPrevious()) {
ensure(i.previous().equals(J = j.previous()), "Error (" + level + ", " + seed + "): divergence in previous()");

if (r.nextFloat() < 0.2) {
i.remove();
j.remove();
}
else if (r.nextFloat() < 0.2) {
KEY_TYPE T = genKey();
i.set(T);
j.set(KEY2OBJ(T));
}
else if (r.nextFloat() < 0.2) {
KEY_TYPE T = genKey();
i.add(T);
j.add(KEY2OBJ(T));
}
}
}
}

```

```

ensure(i.nextIndex() == j.nextIndex(), "Error (" + level + ", " + seed + "): divergence in nextIndex()");
ensure(i.previousIndex() == j.previousIndex(), "Error (" + level + ", " + seed + "): divergence in previousIndex()");

}

}

{
Object previous = null;
Object I, J;
long from = m.isEmpty() ? 0 : (r.nextLong() & 0x7FFFFFFFFFFFFFFFL) % m.size64();
KEY_BIG_LIST_ITERATOR i;
KEY_BIG_LIST_ITERATOR j;
i = m.listIterator(from);
j = t.listIterator(from);

for(int k = 0; k < 2*n; k++) {
ensure(i.hasNext() == j.hasNext(), "Error (" + level + ", " + seed + "): divergence in hasNext() (iterator with starting
point " + from + ")");
ensure(i.hasPrevious() == j.hasPrevious() , "Error (" + level + ", " + seed + "): divergence in hasPrevious() (iterator
with starting point " + from + ")");

if (r.nextFloat() < .8 && i.hasNext()) {
ensure((I = i.next()).equals(J = j.next()), "Error (" + level + ", " + seed + "): divergence in next() (" + I + ", " + J + ",
iterator with starting point " + from + ")");
//System.err.println("Done next " + I + " " + J + " " + badPrevious);

if (r.nextFloat() < 0.2) {
//System.err.println("Removing in next");
i.remove();
j.remove();
}
else if (r.nextFloat() < 0.2) {
KEY_TYPE T = genKey();
i.set(T);
j.set(KEY2OBJ(T));
}
else if (r.nextFloat() < 0.2) {
KEY_TYPE T = genKey();
i.add(T);
j.add(KEY2OBJ(T));
}
}
else if (r.nextFloat() < .2 && i.hasPrevious()) {
ensure((I = i.previous()).equals(J = j.previous()), "Error (" + level + ", " + seed + "): divergence in previous() (" + I +
", " + J + ", iterator with starting point " + from + ")");

if (r.nextFloat() < 0.2) {

```

```

//System.err.println("Removing in prev");
i.remove();
j.remove();
}
else if (r.nextFloat() < 0.2) {
    KEY_TYPE T = genKey();
    i.set(T);
    j.set(KEY2OBJ(T));
}
else if (r.nextFloat() < 0.2) {
    KEY_TYPE T = genKey();
    i.add(T);
    j.add(KEY2OBJ(T));
}
}
}

}

/* Now we check that m actually holds that data. */

ensure(m.equals(t), "Error (" + level + ", " + seed + "): ! m.equals(t) after iteration");
ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after iteration");

/* Now we select a pair of keys and create a subset. */

if (! m.isEmpty()) {
    long start = (r.nextLong() & 0x7FFFFFFFFFFFFFFFL) % m.size64();
    long end = start + (r.nextLong() & 0x7FFFFFFFFFFFFFFFL) % (m.size64() - start);
    //System.err.println("Checking subList from " + start + " to " + end + " (level=" + (level+1) + ")...");
    testLists(m.subList(start, end), t.subList(start, end), n, level + 1);

    ensure(m.equals(t), "Error (" + level + ", " + seed + m + t + "): ! m.equals(t) after subList");
    ensure(t.equals(m), "Error (" + level + ", " + seed + "): ! t.equals(m) after subList");

}

m.clear();
t.clear();
ensure(m.isEmpty(), "Error (" + level + ", " + seed + "): m is not empty after clear()");
}

protected static void runTest(int n) {
    BIG_ARRAY_BIG_LIST m = new BIG_ARRAY_BIG_LIST();
    BIG_LIST t = BIG_LISTS.asBigList(new ARRAY_LIST());
    topList = m;
    k = new Object[n];

```

```

nk = new Object[n];
kt = new KEY_TYPE[n];
nkt = new KEY_TYPE[n];

for(int i = 0; i < n; i++) {
#if KEYS_REFERENCE
k[i] = kt[i] = genKey();
nk[i] = nkt[i] = genKey();
#else
k[i] = new KEY_CLASS(kt[i] = genKey());
nk[i] = new KEY_CLASS(nkt[i] = genKey());
#endif
}

/* We add pairs to t. */
#if KEYS_PRIMITIVE
for(int i = 0; i < n; i++) t.add((KEY_GENERIC_CLASS)k[i]);
#else
for(int i = 0; i < n; i++) t.add(k[i]);
#endif

/* We add to m the same data */
m.addAll(t);

testLists(m, t, n, 0);

System.out.println("Test OK");
return;
}

public static void main(String args[]) {
int n = Integer.parseInt(args[1]);
if (args.length > 2) r = new java.util.Random(seed = Long.parseLong(args[2]));

try {
if ("speedTest".equals(args[0]) || "speedComp".equals(args[0])) speedTest(n, "speedComp".equals(args[0]));
else if ("test".equals(args[0])) runTest(n);
} catch(Throwable e) {
e.printStackTrace(System.err);
System.err.println("seed: " + seed);
}
}

#endif
}

```

Found in path(s):

```
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-
a775574/drv/BigArrayBigList.drv
```

No license file was found, but licenses were detected in source scan.

```
/*
```

```
* Copyright (C) 2003-2017 Paolo Boldi and Sebastiano Vigna
```

```
*
```

```
* Licensed under the Apache License, Version 2.0 (the "License");
```

```
* you may not use this file except in compliance with the License.
```

```
* You may obtain a copy of the License at
```

```
*
```

```
* http://www.apache.org/licenses/LICENSE-2.0
```

```
*
```

```
* Unless required by applicable law or agreed to in writing, software
```

```
* distributed under the License is distributed on an "AS IS" BASIS,
```

```
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
```

```
* See the License for the specific language governing permissions and
```

```
* limitations under the License.
```

```
*/
```

```
package PACKAGE;
```

```
#if KEY_CLASS_Object
```

```
import java.util.Comparator;
```

```
import it.unimi.dsi.fastutil.IndirectPriorityQueue;
```

```
#endif
```

```
import java.util.NoSuchElementException;
```

```
import it.unimi.dsi.fastutil.ints.IntArrays;
```

```
/** A type-specific heap-based semi-indirect priority queue.
```

```
*
```

```
* <p>Instances of this class use as reference list a <em>reference array</em>,</p>
```

```
* which must be provided to each constructor. The priority queue is
```

```
* represented using a heap. The heap is enlarged as needed, but it is never
```

```
* shrunk. Use the { @link #trim() } method to reduce its size, if necessary.
```

```
*
```

```
* <p>This implementation allows one to enqueue several time the same index, but
```

```
* you must be careful when calling { @link #changed() }.
```

```
*/
```

```
public class HEAP_SEMI_INDIRECT_PRIORITY_QUEUE KEY_GENERIC implements
INDIRECT_PRIORITY_QUEUE KEY_GENERIC {
```

```

/** The reference array. */
protected final KEY_GENERIC_TYPE refArray[];

/** The semi-indirect heap. */
protected int heap[] = IntArrays.EMPTY_ARRAY;

/** The number of elements in this queue. */
protected int size;

/** The type-specific comparator used in this queue. */
protected KEY_COMPARATOR KEY_SUPER_GENERIC c;

/** Creates a new empty queue without elements with a given capacity and comparator.
 *
 * @param refArray the reference array.
 * @param capacity the initial capacity of this queue.
 * @param c the comparator used in this queue, or { @code null } for the natural order.
 */
public HEAP_SEMI_INDIRECT_PRIORITY_QUEUE(KEY_GENERIC_TYPE[] refArray, int capacity,
KEY_COMPARATOR KEY_SUPER_GENERIC c) {
    if (capacity > 0) this.heap = new int[capacity];
    this.refArray = refArray;
    this.c = c;
}

/** Creates a new empty queue with given capacity and using the natural order.
 *
 * @param refArray the reference array.
 * @param capacity the initial capacity of this queue.
 */
public HEAP_SEMI_INDIRECT_PRIORITY_QUEUE(KEY_GENERIC_TYPE[] refArray, int capacity) {
    this(refArray, capacity, null);
}

/** Creates a new empty queue with capacity equal to the length of the reference array and a given comparator.
 *
 * @param refArray the reference array.
 * @param c the comparator used in this queue, or { @code null } for the natural order.
 */
public HEAP_SEMI_INDIRECT_PRIORITY_QUEUE(KEY_GENERIC_TYPE[] refArray,
KEY_COMPARATOR KEY_SUPER_GENERIC c) {
    this(refArray, refArray.length, c);
}

/** Creates a new empty queue with capacity equal to the length of the reference array and using the natural order.
 *
 * @param refArray the reference array.
 */
public HEAP_SEMI_INDIRECT_PRIORITY_QUEUE(final KEY_GENERIC_TYPE[] refArray) {

```

```
this(refArray, refArray.length, null);
}
```

```
/** Wraps a given array in a queue using a given comparator.
```

```
*
```

```
* <p>The queue returned by this method will be backed by the given array.
```

```
* The first { @code size } element of the array will be rearranged so to form a heap (this is
```

```
* more efficient than enqueueing the elements of { @code a } one by one).
```

```
*
```

```
* @param refArray the reference array.
```

```
* @param a an array of indices into { @code refArray }.
```

```
* @param size the number of elements to be included in the queue.
```

```
* @param c the comparator used in this queue, or { @code null } for the natural order.
```

```
*/
```

```
public HEAP_SEMI_INDIRECT_PRIORITY_QUEUE(final KEY_GENERIC_TYPE[] refArray, final int[] a, int
size, final KEY_COMPARATOR KEY_SUPER_GENERIC c) {
    this(refArray, 0, c);
    this.heap = a;
    this.size = size;
    SEMI_INDIRECT_HEAPS.makeHeap(refArray, a, size, c);
}
```

```
/** Wraps a given array in a queue using a given comparator.
```

```
*
```

```
* <p>The queue returned by this method will be backed by the given array.
```

```
* The elements of the array will be rearranged so to form a heap (this is
```

```
* more efficient than enqueueing the elements of { @code a } one by one).
```

```
*
```

```
* @param refArray the reference array.
```

```
* @param a an array of indices into { @code refArray }.
```

```
* @param c the comparator used in this queue, or { @code null } for the natural order.
```

```
*/
```

```
public HEAP_SEMI_INDIRECT_PRIORITY_QUEUE(final KEY_GENERIC_TYPE[] refArray, final int[] a, final
KEY_COMPARATOR KEY_SUPER_GENERIC c) {
    this(refArray, a, a.length, c);
}
```

```
/** Wraps a given array in a queue using the natural order.
```

```
*
```

```
* <p>The queue returned by this method will be backed by the given array.
```

```
* The first { @code size } element of the array will be rearranged so to form a heap (this is
```

```
* more efficient than enqueueing the elements of { @code a } one by one).
```

```
*
```

```
* @param refArray the reference array.
```

```
* @param a an array of indices into { @code refArray }.
```

```
* @param size the number of elements to be included in the queue.
```

```

*/
public HEAP_SEMI_INDIRECT_PRIORITY_QUEUE(final KEY_GENERIC_TYPE[] refArray, final int[] a, int
size) {
    this(refArray, a, size, null);
}

/** Wraps a given array in a queue using the natural order.
 *
 * <p>The queue returned by this method will be backed by the given array.
 * The elements of the array will be rearranged so to form a heap (this is
 * more efficient than enqueueing the elements of { @code a } one by one).
 *
 * @param refArray the reference array.
 * @param a an array of indices into { @code refArray }.
 */
public HEAP_SEMI_INDIRECT_PRIORITY_QUEUE(final KEY_GENERIC_TYPE[] refArray, final int[] a) {
    this(refArray, a, a.length);
}

/** Ensures that the given index is a valid reference.
 *
 * @param index an index in the reference array.
 * @throws IndexOutOfBoundsException if the given index is negative or larger than the reference array length.
 */
protected void ensureElement(final int index) {
    if (index < 0) throw new IndexOutOfBoundsException("Index (" + index + ") is negative");
    if (index >= refArray.length) throw new IndexOutOfBoundsException("Index (" + index + ") is larger than or equal
to reference array size (" + refArray.length + ")");
}

@Override
public void enqueue(int x) {
    ensureElement(x);

    if (size == heap.length) heap = IntArrays.grow(heap, size + 1);

    heap[size++] = x;
    SEMI_INDIRECT_HEAPS.upHeap(refArray, heap, size, size - 1, c);
}

@Override
public int dequeue() {
    if (size == 0) throw new NoSuchElementException();
    final int result = heap[0];
    heap[0] = heap[--size];
    if (size != 0) SEMI_INDIRECT_HEAPS.downHeap(refArray, heap, size, 0, c);
    return result;
}

```

```

}

@Override
public int first() {
    if (size == 0) throw new NoSuchElementException();
    return heap[0];
}

/** {@inheritDoc}
 *
 * <p>The caller <strong>must</strong> guarantee that when this method is called the
 * index of the first element appears just once in the queue. Failure to do so
 * will bring the queue in an inconsistent state, and will cause
 * unpredictable behaviour.
 */
@Override
public void changed() {
    SEMI_INDIRECT_HEAPS.downHeap(refArray, heap, size, 0, c);
}

/** Rebuilds this heap in a bottom-up fashion (in linear time). */

@Override
public void allChanged() { SEMI_INDIRECT_HEAPS.makeHeap(refArray, heap, size, c); }

@Override
public int size() { return size; }

@Override
public void clear() { size = 0; }

/** Trims the backing array so that it has exactly {@link #size()} elements. */

public void trim() {
    heap = IntArrays.trim(heap, size);
}

@Override
public KEY_COMPARATOR KEY_SUPER_GENERIC comparator() { return c; }

/** Writes in the provided array the <em>front</em> of the queue, that is, the set of indices
 * whose elements have the same priority as the top.
 * @param a an array whose initial part will be filled with the front (must be sized as least as the heap size).
 * @return the number of elements of the front.
 */
@Override
public int front(final int[] a) { return c == null ? SEMI_INDIRECT_HEAPS.front(refArray, heap, size, a) :
    SEMI_INDIRECT_HEAPS.front(refArray, heap, size, a, c); }

```

```

@Override
public String toString() {
    StringBuffer s = new StringBuffer();
    s.append("[");
    for (int i = 0; i < size; i++) {
        if (i != 0) s.append(", ");
        s.append(refArray[heap [i]]);
    }
    s.append("]");
    return s.toString();
}

#ifdef TEST

/** The original class, now just used for testing. */

private static class TestQueue {

    /** The reference array */
    private KEY_TYPE refArray[];
    /** Its length */
    private int N;
    /** The number of elements in the heaps */
    private int n;
    /** The two comparators */
    private KEY_COMPARATOR primaryComp, secondaryComp;
    /** Two indirect heaps are used, called { @code primary } and { @code secondary }. Each of them contains
    a permutation of { @code n } among the indices 0, 1, ..., { @code N }-1 in such a way that the corresponding
    objects be sorted with respect to the two comparators.
    We also need an array { @code inSec[] } so that { @code inSec[k] } is the index of { @code secondary }
    containing { @code k }.
    */
    private int primary[], secondary[], inSec[];

    /** Builds a double indirect priority queue.
    * @param refArray The reference array.
    * @param primaryComp The primary comparator.
    * @param secondaryComp The secondary comparator.
    */
    public TestQueue(KEY_TYPE refArray[], KEY_COMPARATOR primaryComp, KEY_COMPARATOR
secondaryComp) {
        this.refArray = refArray;
        this.N = refArray.length;
        assert this.N != 0;
        this.n = 0;
        this.primaryComp = primaryComp;
        this.secondaryComp = secondaryComp;
    }
}

```

```

this.primary = new int[N];
this.secondary = new int[N];
this.inSec = new int[N];
java.util.Arrays.fill(inSec, -1);
}

/** Adds an index to the queue. Notice that the index should not be already present in the queue.
 * @param i The index to be added
 */
public void add(int i) {
    if (i < 0 || i >= refArray.length) throw new IndexOutOfBoundsException();
    //if (inSec[i] >= 0) throw new IllegalArgumentException();
    primary[n] = i;
    n++;
    swimPrimary(n-1);
}

/** Heapify the primary heap.
 * @param i The index of the heap to be heapified.
 */
private void heapifyPrimary(int i) {
    int dep = primary[i];
    int child;

    while ((child = 2*i+1) < n) {
        if (child+1 < n && primaryComp.compare(refArray[primary[child+1]], refArray[primary[child]]) < 0) child++;
        if (primaryComp.compare(refArray[dep], refArray[primary[child]]) <= 0) break;
        primary[i] = primary[child];
        i = child;
    }
    primary[i] = dep;
}

/** Heapify the secondary heap.
 * @param i The index of the heap to be heapified.
 */
private void heapifySecondary(int i) {
    int dep = secondary[i];
    int child;

    while ((child = 2*i+1) < n) {
        if (child+1 < n && secondaryComp.compare(refArray[secondary[child+1]], refArray[secondary[child]]) < 0)
child++;
        if (secondaryComp.compare(refArray[dep], refArray[secondary[child]]) <= 0) break;
        secondary[i] = secondary[child]; inSec[secondary[i]] = i;
        i = child;
    }
    secondary[i] = dep; inSec[secondary[i]] = i;
}

```

```

}

/** Swim and heapify the primary heap.
 * @param i The index to be moved.
 */
private void swimPrimary(int i) {
    int dep = primary[i];
    int parent;

    while (i != 0 && (parent = (i - 1) / 2) >= 0) {
        if (primaryComp.compare(refArray[primary[parent]], refArray[dep]) <= 0) break;
        primary[i] = primary[parent];
        i = parent;
    }
    primary[i] = dep;
    heapifyPrimary(i);
}

/** Swim and heapify the secondary heap.
 * @param i The index to be moved.
 */
private void swimSecondary(int i) {
    int dep = secondary[i];
    int parent;

    while (i != 0 && (parent = (i - 1) / 2) >= 0) {
        if (secondaryComp.compare(refArray[secondary[parent]], refArray[dep]) <= 0) break;
        secondary[i] = secondary[parent]; inSec[secondary[i]] = i;
        i = parent;
    }
    secondary[i] = dep; inSec[secondary[i]] = i;
    heapifySecondary(i);
}

/** Returns the minimum element with respect to the primary comparator.
 * @return the minimum element.
 */
public int top() {
    if (n == 0) throw new NoSuchElementException();
    return primary[0];
}

/** Returns the minimum element with respect to the secondary comparator.
 * @return the minimum element.
 */
public int secTop() {
    if (n == 0) throw new NoSuchElementException();
    return secondary[0];
}

```

```

}

/** Removes the minimum element with respect to the primary comparator.
 * @return the removed element.
 */
public void remove() {
    if (n == 0) throw new NoSuchElementException();
    int result = primary[0];
    // Copy a leaf
    primary[0] = primary[n-1];
    n--;
    heapifyPrimary(0);
    return;
}

public void clear() {
    while(size() != 0) remove();
}

/** Signals that the minimum element with respect to the comparator has changed.
 */
public void change() {
    heapifyPrimary(0);
}

/** Returns the number of elements in the queue.
 * @return the size of the queue
 */
public int size() {
    return n;
}

public String toString() {
    String s = "[";
    for (int i = 0; i < n; i++)
        s += refArray[primary[i]]+", ";
    return s+ "]";
}
}

private static long seed = System.currentTimeMillis();
private static java.util.Random r = new java.util.Random(seed);

private static KEY_TYPE genKey() {
    #if KEY_CLASS_Byte || KEY_CLASS_Short || KEY_CLASS_Character
        return (KEY_TYPE)(r.nextInt());
    #elif KEYS_PRIMITIVE
        return r.NEXT_KEY();
    #endif
}

```

```

#elif KEY_CLASS_Object
    return Integer.toBinaryString(r.nextInt());
#else
    return new java.io.Serializable() {};
#endif
}

private static java.text.NumberFormat format = new java.text.DecimalFormat("#,###.00");
private static java.text.FieldPosition p = new java.text.FieldPosition(0);

private static String format(double d) {
    StringBuffer s = new StringBuffer();
    return format.format(d, s, p).toString();
}

private static void speedTest(int n, boolean comp) {
    System.out.println("There are presently no speed tests for this class.");
}

private static void fatal(String msg) {
    System.out.println(msg);
    System.exit(1);
}

private static void ensure(boolean cond, String msg) {
    if (cond) return;
    fatal(msg);
}

private static boolean heapEqual(int[] a, int[] b, int sizea, int sizeb) {
    if (sizea != sizeb) return false;
    while(sizea-- != 0) if (a[sizea] != b[sizea]) return false;
    return true;
}

protected static void runTest(int n) {
    long ms;
    Exception mThrowsIllegal, tThrowsIllegal, mThrowsOutOfBounds, tThrowsOutOfBounds, mThrowsNoElement,
    tThrowsNoElement;
    int rm = 0, rt = 0;
    KEY_TYPE[] refArray = new KEY_TYPE[n];

    for(int i = 0; i < n; i++) refArray[i] = genKey();

    HEAP_SEMI_INDIRECT_PRIORITY_QUEUE m = new
    HEAP_SEMI_INDIRECT_PRIORITY_QUEUE(refArray, COMPARATORS.NATURAL_COMPARATOR);
    TestQueue t = new TestQueue(refArray, COMPARATORS.NATURAL_COMPARATOR,

```

```

COMPARATORS.OPPOSITE_COMPARATOR);

/* We add pairs to t. */
for(int i = 0; i < n / 2; i++) {
    t.add(i);
    m.enqueue(i);
}

ensure(heapEqual(m.heap, t.primary, m.size(), t.size()), "Error (" + seed + "): m and t differ after creation (" + m + ",
" + t + ")");

/* Now we add and remove random data in m and t, checking that the result is the same. */

for(int i=0; i<2*n; i++) {
    if (r.nextDouble() < 0.01) {
        t.clear();
        m.clear();
        for(int j = 0; j < n / 2; j++) {
            t.add(j);
            m.enqueue(j);
        }
    }
}

int T = r.nextInt(2 * n);

mThrowsNoElement = tThrowsNoElement = mThrowsOutOfBounds = tThrowsOutOfBounds = mThrowsIllegal =
tThrowsIllegal = null;

try {
    m.enqueue(T);
}
catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }
catch (IllegalArgumentException e) { mThrowsIllegal = e; }

try {
    t.add(T);
}
catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }
catch (IllegalArgumentException e) { tThrowsIllegal = e; }

ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + seed + "): enqueue()
divergence in IndexOutOfBoundsException for " + T + " (" + mThrowsOutOfBounds + ", " + tThrowsOutOfBounds
+ ")");
ensure((mThrowsIllegal == null) == (tThrowsIllegal == null), "Error (" + seed + "): enqueue() divergence in
IllegalArgumentException for " + T + " (" + mThrowsIllegal + ", " + tThrowsIllegal + ")");

ensure(heapEqual(m.heap, t.primary, m.size(), t.size()), "Error (" + seed + "): m and t differ after enqueue (" + m +
", " + t + ")");

```

```

    if (m.size() != 0) {
        ensure(m.first() == t.top(), "Error (" + seed + "): m and t differ in first element after enqueue (" + m.first() + ", " +
t.top() + ")");
    }

    mThrowsNoElement = tThrowsNoElement = mThrowsOutOfBounds = tThrowsOutOfBounds = mThrowsIllegal =
tThrowsIllegal = null;

    try {
        rm = m.dequeue();
    }
    catch (IndexOutOfBoundsException e) { mThrowsOutOfBounds = e; }
    catch (IllegalArgumentException e) { mThrowsIllegal = e; }
    catch (NoSuchElementException e) { mThrowsNoElement = e; }

    try {
        rt = t.top();
        t.remove();
    }
    catch (IndexOutOfBoundsException e) { tThrowsOutOfBounds = e; }
    catch (IllegalArgumentException e) { tThrowsIllegal = e; }
    catch (NoSuchElementException e) { tThrowsNoElement = e; }

    ensure((mThrowsOutOfBounds == null) == (tThrowsOutOfBounds == null), "Error (" + seed + "): dequeue()
divergence in IndexOutOfBoundsException (" + mThrowsOutOfBounds + ", " + tThrowsOutOfBounds + ")");
    ensure((mThrowsIllegal == null) == (tThrowsIllegal == null), "Error (" + seed + "): dequeue() divergence in
IllegalArgumentException (" + mThrowsIllegal + ", " + tThrowsIllegal + ")");
    ensure((mThrowsNoElement == null) == (tThrowsNoElement == null), "Error (" + seed + "): dequeue() divergence
in NoSuchElementException (" + mThrowsNoElement + ", " + tThrowsNoElement + ")");
    if (mThrowsOutOfBounds == null) ensure(rt == rm, "Error (" + seed + "): divergence in dequeue() between t and m
(" + rt + ", " + rm + ")");

    ensure(heapEqual(m.heap, t.primary, m.size(), t.size()), "Error (" + seed + "): m and t differ after dequeue (" + m +
", " + t + ")");

    if (m.size() != 0) {
        ensure(m.first() == t.top(), "Error (" + seed + "): m and t differ in first element after dequeue (" + m.first() + ", " +
t.top() + ")");
    }

    if (m.size() != 0 && ((new it.unimi.dsi.fastutil.ints.IntOpenHashSet(m.heap, 0, m.size())).size() == m.size())) {

        refArray[m.first()] = genKey();

        m.changed();
        t.change();

```

```

    ensure(heapEqual(m.heap, t.primary, m.size(), t.size()), "Error (" + seed + "): m and t differ after change (" + m + ",
" + t + ")");

    if (m.size() != 0) {
        ensure(m.first() == t.top(), "Error (" + seed + "): m and t differ in first element after change (" + m.first() + ", " +
t.top() + ")");
    }
}
}
}

```

```

/* Now we check that m actually holds the same data. */

```

```

m.clear();
ensure(m.isEmpty(), "Error (" + seed + "): m is not empty after clear()");

```

```

System.out.println("Test OK");
}

```

```

public static void main(String args[]) {
    int n = Integer.parseInt(args[1]);
    if (args.length > 2) r = new java.util.Random(seed = Long.parseLong(args[2]));

    try {
        if ("speedTest".equals(args[0]) || "speedComp".equals(args[0])) speedTest(n, "speedComp".equals(args[0]));
        else if ("test".equals(args[0])) runTest(n);
    } catch(Throwable e) {
        e.printStackTrace(System.err);
        System.err.println("seed: " + seed);
    }
}

#endif
}

```

Found in path(s):

```

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-
a775574/drv/HeapSemiIndirectPriorityQueue.drv

```

No license file was found, but licenses were detected in source scan.

```

/*

```

```

* Copyright (C) 2002-2017 Sebastiano Vigna

```

```

*

```

```

* Licensed under the Apache License, Version 2.0 (the "License");

```

- * you may not use this file except in compliance with the License.
- * You may obtain a copy of the License at
- *
- * <http://www.apache.org/licenses/LICENSE-2.0>
- *
- * Unless required by applicable law or agreed to in writing, software
- * distributed under the License is distributed on an "AS IS" BASIS,
- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.
- */

```
package PACKAGE;
```

```
import VALUE_PACKAGE.VALUE_COLLECTION;
```

```
#if !KEY_CLASS_Object
import it.unimi.dsi.fastutil.objects.ObjectSet;
import it.unimi.dsi.fastutil.objects.ObjectIterator;
#endif
```

```
import java.util.function.Consumer;
import java.util.Map;
```

```
/** A type-specific {@link Map}; provides some additional methods that use polymorphism to avoid (un)boxing,
and handling of a default return value.
```

```
*
* <p>Besides extending the corresponding type-specific {@linkplain it.unimi.dsi.fastutil.Function function}, this
interface strengthens {@link Map#entrySet()},
* {@link #keySet()} and {@link #values()}. Moreover, a number of methods, such as {@link #size()}, {@link
#defaultReturnValue()}, etc., are un-defaulted
* as their function default do not make sense for a map.
* Maps returning entry sets of type {@link FastEntrySet} support also fast iteration.
*
* <p>A submap or subset may or may not have an
* independent default return value (which however must be initialized to the
* default return value of the originator).
*
* @see Map
*/
```

```
public interface MAP KEY_VALUE_GENERIC extends FUNCTION KEY_VALUE_GENERIC,
Map<KEY_GENERIC_CLASS,VALUE_GENERIC_CLASS> {
```

```
/** An entry set providing fast iteration.
```

```
*
* <p>In some cases (e.g., hash-based classes) iteration over an entry set requires the creation
```

```

* of a large number of { @link java.util.Map.Entry } objects. Some { @code fastutil }
* maps might return { @link plain Map#entrySet() entry set } objects of type { @code FastEntrySet }; in this case,
{ @link #fastIterator() fastIterator() }
* will return an iterator that is guaranteed not to create a large number of objects, <em>possibly
* by returning always the same entry</em> (of course, mutated), and { @link #fastForEach(Consumer) } will apply
* the provided consumer to all elements of the entry set, <em>which might be represented
* always by the same entry</em> (of course, mutated).
*/

```

```

interface FastEntrySet KEY_VALUE_GENERIC extends ObjectSet<MAP.Entry KEY_VALUE_GENERIC> {
/** Returns a fast iterator over this entry set; the iterator might return always the same entry instance, suitably
mutated.
*
* @return a fast iterator over this entry set; the iterator might return always the same { @link java.util.Map.Entry }
instance, suitably mutated.
*/

```

```

ObjectIterator<MAP.Entry KEY_VALUE_GENERIC> fastIterator();

```

```

/** Iterates quickly over this entry set; the iteration might happen always on the same entry instance, suitably
mutated.

```

```

*
* <p>This default implementation just delegates to { @link #forEach(Consumer) }.
*
* @param consumer a consumer that will by applied to the entries of this set; the entries might be represented
* by the same entry instance, suitably mutated.
* @since 8.1.0
*/

```

```

default void fastForEach(final Consumer<? super MAP.Entry KEY_VALUE_GENERIC> consumer) {
    forEach(consumer);
}
}

```

```

/**
* Returns the number of key/value mappings in this map. If the
* map contains more than { @link Integer#MAX_VALUE } elements, returns { @link Integer#MAX_VALUE }.
*
* @return the number of key-value mappings in this map.
* @see it.unimi.dsi.fastutil.Size64
*/

```

```

@Override

```

```

int size();

```

```

/** Removes all of the mappings from this map (optional operation).
* The map will be empty after this call returns.
*
* @throws UnsupportedOperationException if the <tt>clear</tt> operation is not supported by this map
*/

```

```

@Override
default void clear() { throw new UnsupportedOperationException(); }

#if VALUES_PRIMITIVE
/** Sets the default return value (optional operation).
 *
 * This value must be returned by type-specific versions of { @code get() },
 * { @code put() } and { @code remove() } to denote that the map does not contain
 * the specified key. It must be 0/{ @code false } by default.
 *
 * @param rv the new default return value.
 * @see #defaultReturnValue()
 */
#else
/** Sets the default return value (optional operation).
 *
 * This value must be returned by type-specific versions of { @code get() },
 * { @code put() } and { @code remove() } to denote that the map does not contain
 * the specified key. It must be { @code null } by default.
 *
 * <p><strong>Warning</strong>: Changing this to a non-null value can have
 * unforeseen consequences. Especially, it breaks compatibility with the
 * specifications of Java's { @link java.util.Map } interface. It has to be
 * used with great care and thorough study of all method comments is
 * recommended.
 *
 * @param rv the new default return value.
 * @see #defaultReturnValue()
 */
#endif

@Override
void defaultReturnValue(VALUE_GENERIC_TYPE rv);

/** Gets the default return value.
 *
 * @return the current default return value.
 */

@Override
VALUE_GENERIC_TYPE defaultReturnValue();

/** Returns a type-specific set view of the mappings contained in this map.
 *
 * <p>This method is necessary because there is no inheritance along
 * type parameters: it is thus impossible to strengthen { @link Map#entrySet() }
 * so that it returns an { @link it.unimi.dsi.fastutil.objects.ObjectSet }

```

```

* of type-specific entries (the latter makes it possible to
* access keys and values with type-specific methods).
*
* @return a type-specific set view of the mappings contained in this map.
* @see Map#entrySet()
*/
ObjectSet<MAP.Entry KEY_VALUE_GENERIC> ENTRYSET();

#if KEYS_PRIMITIVE || VALUES_PRIMITIVE
/** Returns a set view of the mappings contained in this map.
* <p>Note that this specification strengthens the one given in { @link Map#entrySet()}.
*
* @return a set view of the mappings contained in this map.
* @see Map#entrySet()
* @deprecated Please use the corresponding type-specific method instead.
*/
@Deprecated
@Override
@SuppressWarnings({ "unchecked", "rawtypes" })
default ObjectSet<Map.Entry<KEY_GENERIC_CLASS, VALUE_GENERIC_CLASS>> entrySet() {
    return (ObjectSet)ENTRYSET();
}
#else
/** Returns a set view of the mappings contained in this map.
* <p>Note that this specification strengthens the one given in { @link Map#entrySet()}.
*
* @return a set view of the mappings contained in this map.
* @see Map#entrySet()
*/

@Override
@SuppressWarnings({ "unchecked", "rawtypes" })
default ObjectSet<Map.Entry<KEY_GENERIC_CLASS, VALUE_GENERIC_CLASS>> entrySet() {
    return (ObjectSet)ENTRYSET();
}
#endif

#if KEYS_PRIMITIVE || VALUES_PRIMITIVE
/** { @inheritDoc }
* <p>This default implementation just delegates to the corresponding type-specific&ndash;{@linkplain
it.unimi.dsi.fastutil.Function function} method.
* @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
#else
/** { @inheritDoc }
* <p>This default implementation just delegates to the corresponding function method.
*/
#endif

```

```

@Override
default VALUE_GENERIC_CLASS put(final KEY_GENERIC_CLASS key, final VALUE_GENERIC_CLASS
value) {
    return FUNCTION.super.put(key, value);
}

#if KEYS_PRIMITIVE || VALUES_PRIMITIVE

/** {@inheritDoc}
 * <p>This default implementation just delegates to the corresponding type-specific&ndash;{@linkplain
it.unimi.dsi.fastutil.Function function} method.
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default VALUE_GENERIC_CLASS get(final Object key) {
    return FUNCTION.super.get(key);
}

#endif

#if KEYS_PRIMITIVE || VALUES_PRIMITIVE
/** {@inheritDoc}
 * <p>This default implementation just delegates to the corresponding type-specific&ndash;{@linkplain
it.unimi.dsi.fastutil.Function function} method.
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
#else
/** {@inheritDoc}
 * <p>This default implementation just delegates to the corresponding type-specific&ndash;{@linkplain
it.unimi.dsi.fastutil.Function function} method.
 */
#endif
@Override
default VALUE_GENERIC_CLASS remove(final Object key) {
    return FUNCTION.super.remove(key);
}

/** {@inheritDoc}
 * <p>Note that this specification strengthens the one given in {@link Map#keySet()}.
 * @return a set view of the keys contained in this map.
 * @see Map#keySet()
 */

@Override
SET KEY_GENERIC keySet();

/** {@inheritDoc}
 * <p>Note that this specification strengthens the one given in {@link Map#values()}.

```

```

* @return a set view of the values contained in this map.
* @see Map#values()
*/

@Override
VALUE_COLLECTION VALUE_GENERIC values();

#if KEYS_PRIMITIVE

/** Returns true if this function contains a mapping for the specified key.
*
* @param key the key.
* @return true if this function associates a value to {@code key}.
* @see Map#containsKey(Object)
*/

@Override
boolean containsKey(KEY_TYPE key);

/** Returns true if this function contains a mapping for the specified key.
* <p>This default implementation just delegates to the corresponding type-specific {@link plain
it.unimi.dsi.fastutil.Function function} method.
* @deprecated Please use the corresponding type-specific method instead.
*/

@Deprecated
@Override
default boolean containsKey(final Object key) {
    return FUNCTION.super.containsKey(key);
}
#else

/** Returns true if this function contains a mapping for the specified key.
*
* @param key the key.
* @return true if this function associates a value to {@code key}.
* @see Map#containsKey(Object)
*/

@Override
boolean containsKey(Object key);

#endif

#if VALUES_PRIMITIVE

/** Returns {@code true} if this map maps one or more keys to the specified value.
* @see Map#containsValue(Object)

```

```

*/

boolean containsValue(VALUE_TYPE value);

/** { @inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */

@Deprecated
@Override
default boolean containsValue(final Object value) {
    return value == null ? false : containsValue(VALUE_OBJ2TYPE(value));
}

#endif

// Defaultable methods

#if KEYS_PRIMITIVE || VALUES_PRIMITIVE

/** Returns the value to which the specified key is mapped, or the { @code defaultValue } if this
 * map contains no mapping for the key.
 *
 * @param key the key.
 * @param defaultValue the default mapping of the key.
 *
 * @return the value to which the specified key is mapped, or the { @code defaultValue } if this map contains no
 mapping for the key.
 *
 * @see java.util.Map#getOrDefault(Object, Object)
 * @since 8.0.0
 */

default VALUE_GENERIC_TYPE getOrDefault(final KEY_TYPE key, final VALUE_GENERIC_TYPE
defaultValue) {
    final VALUE_GENERIC_TYPE v;
    return ((v = GET_VALUE(key)) != defaultReturnValue() || containsKey(key)) ? v : defaultValue;
}

/** If the specified key is not already associated with a value, associates it with the given
 * value and returns the { @linkplain #defaultReturnValue() default return value }, else returns
 * the current value.
 *
 * @param key key with which the specified value is to be associated.
 * @param value value to be associated with the specified key.
 *
 * @return the previous value associated with the specified key, or the { @linkplain #defaultReturnValue() default
 return value } if there was no mapping for the key.
 *

```

```
* @see java.util.Map#putIfAbsent(Object, Object)
```

```
* @since 8.0.0
```

```
*/
```

```
default VALUE_GENERIC_TYPE putIfAbsent(final KEY_GENERIC_TYPE key, final  
VALUE_GENERIC_TYPE value) {  
    final VALUE_GENERIC_TYPE v = GET_VALUE(key), drv = defaultReturnValue();  
    if (v != drv || containsKey(key)) return v;  
    put(key, value);  
    return drv;  
}
```

```
/** Removes the entry for the specified key only if it is currently mapped to the specified value.
```

```
*
```

```
* @param key key with which the specified value is associated.
```

```
* @param value value expected to be associated with the specified key.
```

```
*
```

```
* @return { @code true } if the value was removed.
```

```
*
```

```
* @see java.util.Map#remove(Object, Object)
```

```
* @since 8.0.0
```

```
*/
```

```
default boolean remove(final KEY_TYPE key, final VALUE_TYPE value) {  
    final VALUE_GENERIC_TYPE curValue = GET_VALUE(key);  
    if (!VALUE_EQUALS(curValue, value) || (curValue == defaultReturnValue() && !containsKey(key))) return false;  
    REMOVE_VALUE(key);  
    return true;  
}
```

```
/** Replaces the entry for the specified key only if currently mapped to the specified value.
```

```
*
```

```
* @param key key with which the specified value is associated.
```

```
* @param oldValue value expected to be associated with the specified key.
```

```
* @param newValue value to be associated with the specified key.
```

```
*
```

```
* @return { @code true } if the value was replaced.
```

```
*
```

```
* @see java.util.Map#replace(Object, Object, Object)
```

```
* @since 8.0.0
```

```
*/
```

```
default boolean replace(final KEY_GENERIC_TYPE key, final VALUE_GENERIC_TYPE oldValue, final  
VALUE_GENERIC_TYPE newValue) {  
    final VALUE_GENERIC_TYPE curValue = GET_VALUE(key);  
    if (!VALUE_EQUALS(curValue, oldValue) || (curValue == defaultReturnValue() && !containsKey(key))) return  
false;  
    put(key, newValue);
```

```

return true;
}

/** Replaces the entry for the specified key only if it is currently mapped to some value.
 *
 * @param key key with which the specified value is associated.
 * @param value value to be associated with the specified key.
 *
 * @return the previous value associated with the specified key, or the { @linkplain #defaultReturnValue() default
return value} if there was no mapping for the key.
 *
 * @see java.util.Map#replace(Object, Object)
 * @since 8.0.0
 */

default VALUE_GENERIC_TYPE replace(final KEY_GENERIC_TYPE key, final VALUE_GENERIC_TYPE
value) { return containsKey(key) ? put(key, value) : defaultReturnValue(); }

#ifdef JDK_PRIMITIVE_FUNCTION

/** If the specified key is not already associated with a value, attempts to compute its value
 * using the given mapping function and enters it into this map.
 *
 * <p>Note that contrarily to the default { @linkplain java.util.Map#computeIfAbsent(Object,
java.util.function.Function) computeIfAbsent()},
 * it is not possible to not add a value for a given key, since the { @code mappingFunction} cannot
 * return { @code null}. If such a behavior is needed, please use the corresponding <em>nullable</em> version.
 *
 * @param key key with which the specified value is to be associated.
 * @param mappingFunction the function to compute a value.
 *
 * @return the current (existing or computed) value associated with the specified key.
 *
 * @see java.util.Map#computeIfAbsent(Object, java.util.function.Function)
 * @since 8.0.0
 */

default VALUE_GENERIC_TYPE COMPUTE_IF_ABSENT_JDK(final KEY_GENERIC_TYPE key, final
JDK_PRIMITIVE_FUNCTION KEY_SUPER_GENERIC VALUE_EXTENDS_GENERIC mappingFunction) {
    java.util.Objects.requireNonNull(mappingFunction);
    final VALUE_GENERIC_TYPE v = GET_VALUE(key);
    if (v != defaultReturnValue() || containsKey(key)) return v;

    VALUE_GENERIC_TYPE newValue =
VALUE_NARROWING(mappingFunction.JDK_PRIMITIVE_FUNCTION_APPLY(key));
    put(key, newValue);
    return newValue;
}

```

```

#endif

#if KEYS_PRIMITIVE && VALUES_PRIMITIVE

/** If the specified key is not already associated with a value, attempts to compute its value
 * using the given mapping function and enters it into this map unless it is {@code null}.
 *
 * <p>Note that this version of {@linkplain java.util.Map#computeIfAbsent(Object, java.util.function.Function)
computeIfAbsent()}
 * should be used only if you plan to return {@code null} in the mapping function.
 *
 * @param key key with which the specified value is to be associated.
 * @param mappingFunction the function to compute a value.
 *
 * @return the current (existing or computed) value associated with the specified key,
 * or the {@linkplain #defaultReturnValue() default return value} if the computed value is {@code null}.
 *
 * @see java.util.Map#computeIfAbsent(Object, java.util.function.Function)
 * @since 8.0.0
 */

default VALUE_GENERIC_TYPE COMPUTE_IF_ABSENT_NULLABLE(final KEY_GENERIC_TYPE key,
final JDK_KEY_TO_GENERIC_FUNCTION<? extends VALUE_GENERIC_CLASS> mappingFunction) {
    java.util.Objects.requireNonNull(mappingFunction);
    final VALUE_GENERIC_TYPE v = GET_VALUE(key), drv = defaultReturnValue();
    if (v != drv || containsKey(key)) return v;

    VALUE_GENERIC_CLASS mappedValue = mappingFunction.apply(key);
    if (mappedValue == null) return drv;
    VALUE_GENERIC_TYPE newValue = VALUE_CLASS2TYPE(mappedValue);
    put(key, newValue);
    return newValue;
}

#endif

/** If the specified key is not already associated with a value, attempts to compute its value
 * using the given mapping function and enters it into this map, unless the key is not present
 * in the given mapping function.
 *
 * <p>This version of {@linkplain java.util.Map#computeIfAbsent(Object, java.util.function.Function)
computeIfAbsent()}
 * uses a type-specific version of <code>fastutil</code>'s {@link it.unimi.dsi.fastutil.Function Function}.
 * Since {@link it.unimi.dsi.fastutil.Function Function} has a {@link
it.unimi.dsi.fastutil.Function#containsKey(Object) containsKey()}
 * method, it is possible to avoid adding a key by having {@code containsKey()} return {@code false} for that key.
 *

```

```

* @param key key with which the specified value is to be associated.
* @param mappingFunction the function to compute a value.
*
* @return the current (existing or computed) value associated with the specified key.
*
* @see java.util.Map#computeIfAbsent(Object, java.util.function.Function)
* @since 8.0.0
*/

default VALUE_GENERIC_TYPE COMPUTE_IF_ABSENT_PARTIAL(final KEY_GENERIC_TYPE key, final
FUNCTION KEY_SUPER_GENERIC VALUE_EXTENDS_GENERIC mappingFunction) {
    java.util.Objects.requireNonNull(mappingFunction);
    final VALUE_GENERIC_TYPE v = GET_VALUE(key), drv = defaultReturnValue();
    if (v != drv || containsKey(key)) return v;

    if (!mappingFunction.containsKey(key)) return drv;
    VALUE_GENERIC_TYPE newValue = mappingFunction.GET_VALUE(key);
    put(key, newValue);
    return newValue;
}

/** If the value for the specified key is present, attempts to compute a new mapping given the key and its current
mapped value.
*
* @param key key with which the specified value is to be associated.
* @param remappingFunction the function to compute a value.
*
* @return the new value associated with the specified key, or the { @linkplain #defaultReturnValue() default return
value} if none.
*
* @see java.util.Map#computeIfPresent(Object, java.util.function.BiFunction)
* @since 8.0.0
*/

default VALUE_GENERIC_TYPE COMPUTE_IF_PRESENT(final KEY_GENERIC_TYPE key, final
java.util.function.BiFunction<? super KEY_GENERIC_CLASS, ? super VALUE_GENERIC_CLASS, ? extends
VALUE_GENERIC_CLASS> remappingFunction) {
    java.util.Objects.requireNonNull(remappingFunction);

    final VALUE_GENERIC_TYPE oldValue = GET_VALUE(key), drv = defaultReturnValue();
    if (oldValue == drv && !containsKey(key)) return drv;
    final VALUE_GENERIC_CLASS newValue = remappingFunction.apply(KEY2OBJ(key),
VALUE2OBJ(oldValue));

    if (newValue == null) { REMOVE_VALUE(key); return drv; }

    #if VALUES_PRIMITIVE
    VALUE_GENERIC_TYPE newVal = VALUE_CLASS2TYPE(newValue);

```

```

    put(key, newVal);
    return newVal;
#else
    put(key, newValue);
    return newValue;
#endif
}

/** Attempts to compute a mapping for the specified key and its current mapped value (or {@code null} if there is
no current mapping).
*
* <p>If the function returns {@code null}, the mapping is removed (or remains absent if initially absent).
* If the function itself throws an (unchecked) exception, the exception is rethrown, and the current mapping is left
unchanged.
*
* @param key key with which the specified value is to be associated.
* @param remappingFunction the function to compute a value.
*
* @return the new value associated with the specified key, or the {@linkplain #defaultReturnValue() default return
value} if none.
*
* @see java.util.Map#compute(Object, java.util.function.BiFunction)
* @since 8.0.0
*/

default VALUE_GENERIC_TYPE COMPUTE(final KEY_GENERIC_TYPE key, final
java.util.function.BiFunction<? super KEY_GENERIC_CLASS, ? super VALUE_GENERIC_CLASS, ? extends
VALUE_GENERIC_CLASS> remappingFunction) {
    java.util.Objects.requireNonNull(remappingFunction);
    final VALUE_GENERIC_TYPE oldValue = GET_VALUE(key), drv = defaultReturnValue();
    final boolean contained = oldValue != drv || containsKey(key);
    final VALUE_GENERIC_CLASS newValue = remappingFunction.apply(KEY2OBJ(key), contained ?
VALUE2OBJ(oldValue) : null);

    if (newValue == null) {
        if (contained) REMOVE_VALUE(key);
        return drv;
    }
#if VALUES_PRIMITIVE
    final VALUE_GENERIC_TYPE newVal = VALUE_CLASS2TYPE(newValue);
    put(key, newVal);
    return newVal;
#else
    put(key, newValue);
    return newValue;
#endif
}

```

```

#if VALUES_PRIMITIVE
/**
 * If the specified key is not already associated with a value, associates it with the given {@code value}.
 * Otherwise, replaces the associated value with the results of the given remapping function, or removes if the result
is {@code null}.
 *
 * @param key key with which the resulting value is to be associated.
 * @param value the value to be merged with the existing value associated with the key or, if no existing value is
associated with the key, to be associated with the key.
 * @param remappingFunction the function to recompute a value if present.
 *
 * @return the new value associated with the specified key, or the {@linkplain #defaultReturnValue() default return
value} if no value is associated with the key.
 *
 * @see java.util.Map#merge(Object, Object, java.util.function.BiFunction)
 * @since 8.0.0
 */
#else
/**
 * If the specified key is not already associated with a value or is associated with null, associates it with the given
non-null {@code value}.
 * Otherwise, replaces the associated value with the results of the given remapping function, or removes if the result
is {@code null}.
 *
 * @param key key with which the resulting value is to be associated.
 * @param value the non-{@code null} value to be merged with the existing value associated with the key or, if no
existing value is associated with the key, to be associated with the key.
 * @param remappingFunction the function to recompute a value if present.
 *
 * @return the new value associated with the specified key, or the {@linkplain #defaultReturnValue() default return
value} if no value is associated with the key.
 *
 * @see java.util.Map#merge(Object, Object, java.util.function.BiFunction)
 * @since 8.0.0
 */
#endif

default VALUE_GENERIC_TYPE MERGE(final KEY_GENERIC_TYPE key, final VALUE_GENERIC_TYPE
value, final java.util.function.BiFunction<? super VALUE_GENERIC_CLASS, ? super
VALUE_GENERIC_CLASS, ? extends VALUE_GENERIC_CLASS> remappingFunction) {
    java.util.Objects.requireNonNull(remappingFunction);
#if VALUES_REFERENCE
    java.util.Objects.requireNonNull(value);
#endif
    final VALUE_GENERIC_TYPE oldValue = GET_VALUE(key), drv = defaultReturnValue();
    final VALUE_GENERIC_TYPE newValue;
    if (oldValue != drv || containsKey(key)) {
        final VALUE_GENERIC_CLASS mergedValue = remappingFunction.apply(VALUE2OBJ(oldValue),

```

```

VALUE2OBJ(value));
    if (mergedValue == null) { REMOVE_VALUE(key); return drv; }
    newValue = VALUE_CLASS2TYPE(mergedValue);
} else {
    newValue = value;
}

put(key, newValue);
return newValue;
}

/** {@inheritDoc}
 * <p>This default implementation just delegates to the corresponding {@link Map} method.
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default VALUE_GENERIC_CLASS getOrDefault(final Object key, final VALUE_GENERIC_CLASS
defaultValue) {
    return Map.super.getOrDefault(key, defaultValue);
}

/** {@inheritDoc}
 * <p>This default implementation just delegates to the corresponding {@link Map} method.
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default VALUE_GENERIC_CLASS putIfAbsent(final KEY_GENERIC_CLASS key, final
VALUE_GENERIC_CLASS value) {
    return Map.super.putIfAbsent(key, value);
}

/** {@inheritDoc}
 * <p>This default implementation just delegates to the corresponding {@link Map} method.
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default boolean remove(final Object key, final Object value) {
    return Map.super.remove(key, value);
}

/** {@inheritDoc}
 * <p>This default implementation just delegates to the corresponding {@link Map} method.
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default boolean replace(final KEY_GENERIC_CLASS key, final VALUE_GENERIC_CLASS oldValue, final
VALUE_GENERIC_CLASS newValue) {
    return Map.super.replace(key, oldValue, newValue);
}

```

```

}

/** {@inheritDoc}
 * <p>This default implementation just delegates to the corresponding {@link Map} method.
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default VALUE_GENERIC_CLASS replace(final KEY_GENERIC_CLASS key, final
VALUE_GENERIC_CLASS value) {
    return Map.super.replace(key, value);
}

#if KEYS_PRIMITIVE

/** {@inheritDoc}
 * <p>This default implementation just delegates to the corresponding {@link Map} method.
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default VALUE_GENERIC_CLASS computeIfAbsent(final KEY_GENERIC_CLASS key, final
java.util.function.Function<? super KEY_GENERIC_CLASS, ? extends VALUE_GENERIC_CLASS>
mappingFunction) {
    return Map.super.computeIfAbsent(key, mappingFunction);
}

/** {@inheritDoc}
 * <p>This default implementation just delegates to the corresponding {@link Map} method.
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default VALUE_GENERIC_CLASS computeIfPresent(final KEY_GENERIC_CLASS key, final
java.util.function.BiFunction<? super KEY_GENERIC_CLASS, ? super VALUE_GENERIC_CLASS, ? extends
VALUE_GENERIC_CLASS> remappingFunction) {
    return Map.super.computeIfPresent(key, remappingFunction);
}

/** {@inheritDoc}
 * <p>This default implementation just delegates to the corresponding {@link Map} method.
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default VALUE_GENERIC_CLASS compute(final KEY_GENERIC_CLASS key, final
java.util.function.BiFunction<? super KEY_GENERIC_CLASS, ? super VALUE_GENERIC_CLASS, ? extends
VALUE_GENERIC_CLASS> remappingFunction) {
    return Map.super.compute(key, remappingFunction);
}

#endif

```

```

/** {@inheritDoc}
 * <p>This default implementation just delegates to the corresponding {@link Map} method.
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default VALUE_GENERIC_CLASS merge(final KEY_GENERIC_CLASS key, final VALUE_GENERIC_CLASS
value, final java.util.function.BiFunction<? super VALUE_GENERIC_CLASS, ? super
VALUE_GENERIC_CLASS, ? extends VALUE_GENERIC_CLASS> remappingFunction) {
    return Map.super.merge(key, value, remappingFunction);
}

```

```
#endif
```

```

/** A type-specific {@link java.util.Map.Entry}; provides some additional methods
 * that use polymorphism to avoid (un)boxing.
 *
 * @see java.util.Map.Entry
 */

```

```

interface Entry KEY_VALUE_GENERIC extends Map.Entry
<KEY_GENERIC_CLASS,VALUE_GENERIC_CLASS> {

```

```
#if KEYS_PRIMITIVE
```

```

/** Returns the key corresponding to this entry.
 * @see java.util.Map.Entry#getKey()
 */

```

```
KEY_TYPE ENTRY_GET_KEY();
```

```

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default KEY_GENERIC_CLASS getKey() {
    return KEY2OBJ(ENTRY_GET_KEY());
}

```

```
#endif
```

```
#if VALUES_PRIMITIVE
```

```

/** Returns the value corresponding to this entry.
 * @see java.util.Map.Entry#getValue()
 */

```

```
VALUE_TYPE ENTRY_GET_VALUE();
```

```

/** Replaces the value corresponding to this entry with the specified value (optional operation).
 * @see java.util.Map.Entry#setValue(Object)
 */

```

```
VALUE_TYPE setValue(final VALUE_TYPE value);
```

```

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default VALUE_GENERIC_CLASS getValue() {
    return VALUE2OBJ(ENTRY_GET_VALUE());
}

/** {@inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default VALUE_GENERIC_CLASS setValue(final VALUE_GENERIC_CLASS value) {
    return VALUE2OBJ(setValue(VALUE_CLASS2TYPE(value)));
}
#endif
}
}

```

Found in path(s):

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/drv/Map.drv

No license file was found, but licenses were detected in source scan.

```

/*
 * Copyright (C) 2002-2017 Sebastiano Vigna
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

```

Found in path(s):

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/src/it/unimi/dsi/fastutil/Arrays.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/src/it/unimi/dsi/fastutil/Function.java

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/src/it/unimi/dsi/fastutil/Stack.java

```
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-
a775574/src/it/unimi/dsi/fastutil/BidirectionalIterator.java
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-
a775574/src/it/unimi/dsi/fastutil/Hash.java
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-
a775574/src/it/unimi/dsi/fastutil/HashCommon.java
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-
a775574/src/it/unimi/dsi/fastutil/AbstractStack.java
```

No license file was found, but licenses were detected in source scan.

```
/*
```

```
* Copyright (C) 2010-2017 Sebastiano Vigna
```

```
*
```

```
* Licensed under the Apache License, Version 2.0 (the "License");
```

```
* you may not use this file except in compliance with the License.
```

```
* You may obtain a copy of the License at
```

```
*
```

```
* http://www.apache.org/licenses/LICENSE-2.0
```

```
*
```

```
* Unless required by applicable law or agreed to in writing, software
```

```
* distributed under the License is distributed on an "AS IS" BASIS,
```

```
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
```

```
* See the License for the specific language governing permissions and
```

```
* limitations under the License.
```

```
*/
```

```
package PACKAGE;
```

```
#if KEYS_REFERENCE
```

```
import it.unimi.dsi.fastutil.Stack;
```

```
#endif
```

```
import java.util.Iterator;
```

```
import java.util.Collection;
```

```
import java.util.NoSuchElementException;
```

```
import it.unimi.dsi.fastutil.BigList;
```

```
import it.unimi.dsi.fastutil.BigListIterator;
```

```
/** An abstract class providing basic methods for big lists implementing a type-specific big list interface. */
```

```
public abstract class ABSTRACT_BIG_LIST KEY_GENERIC extends ABSTRACT_COLLECTION
```

```
KEY_GENERIC implements BIG_LIST KEY_GENERIC, STACK KEY_GENERIC {
```

```
protected ABSTRACT_BIG_LIST() {}
```

```
/** Ensures that the given index is nonnegative and not greater than this big-list size.
```

```

*
* @param index an index.
* @throws IndexOutOfBoundsException if the given index is negative or greater than this big-list size.
*/
protected void ensureIndex(final long index) {
    if (index < 0) throw new IndexOutOfBoundsException("Index (" + index + ") is negative");
    if (index > size64()) throw new IndexOutOfBoundsException("Index (" + index + ") is greater than list size (" +
(size64()) + ")");
}

/** Ensures that the given index is nonnegative and smaller than this big-list size.
*
* @param index an index.
* @throws IndexOutOfBoundsException if the given index is negative or not smaller than this big-list size.
*/
protected void ensureRestrictedIndex(final long index) {
    if (index < 0) throw new IndexOutOfBoundsException("Index (" + index + ") is negative");
    if (index >= size64()) throw new IndexOutOfBoundsException("Index (" + index + ") is greater than or equal to list
size (" + (size64()) + ")");
}

/** {@inheritDoc}
*
* <p>This implementation always throws an {@link UnsupportedOperationException}.
*/
@Override
public void add(final long index, final KEY_GENERIC_TYPE k) {
    throw new UnsupportedOperationException();
}

/** {@inheritDoc}
*
* <p>This implementation delegates to the type-specific version of {@link BigList#add(long, Object)}.
*/
@Override
public boolean add(final KEY_GENERIC_TYPE k) {
    add(size64(), k);
    return true;
}

/** {@inheritDoc}
*
* <p>This implementation always throws an {@link UnsupportedOperationException}.
*/
@Override
public KEY_GENERIC_TYPE REMOVE_KEY(long i) {
    throw new UnsupportedOperationException();
}

```

```

/** {@inheritDoc}
 *
 * <p>This implementation always throws an {@link UnsupportedOperationException}.
 */
@Override
public KEY_GENERIC_TYPE set(final long index, final KEY_GENERIC_TYPE k) {
    throw new UnsupportedOperationException();
}

/** Adds all of the elements in the specified collection to this list (optional operation). */
@Override
public boolean addAll(long index, final Collection<? extends KEY_GENERIC_CLASS> c) {
    ensureIndex(index);
    final Iterator<? extends KEY_GENERIC_CLASS> i = c.iterator();
    final boolean retVal = i.hasNext();
    while(i.hasNext()) add(index++, i.next());
    return retVal;
}

/** {@inheritDoc}
 *
 * <p>This implementation delegates to the type-specific version of {@link BigList#addAll(long, Collection)}.
 */
@Override
public boolean addAll(final Collection<? extends KEY_GENERIC_CLASS> c) {
    return addAll(size64(), c);
}

/** {@inheritDoc}
 *
 * <p>This implementation delegates to {@link #listIterator()}.
 */
@Override
public KEY_BIG_LIST_ITERATOR KEY_GENERIC iterator() {
    return listIterator();
}

/** {@inheritDoc}
 *
 * <p>This implementation delegates to {@link BigList#listIterator(long) listIterator(0)}.
 */
@Override
public KEY_BIG_LIST_ITERATOR KEY_GENERIC listIterator() {
    return listIterator(0L);
}

/** {@inheritDoc}

```

```

* <p>This implementation is based on the random-access methods. */
@Override
public KEY_BIG_LIST_ITERATOR KEY_GENERIC listIterator(final long index) {
    ensureIndex(index);

    return new KEY_BIG_LIST_ITERATOR KEY_GENERIC() {
        long pos = index, last = -1;

        @Override
        public boolean hasNext() { return pos < ABSTRACT_BIG_LIST.this.size64(); }

        @Override
        public boolean hasPrevious() { return pos > 0; }

        @Override
        public KEY_GENERIC_TYPE NEXT_KEY() { if (! hasNext()) throw new NoSuchElementException(); return
ABSTRACT_BIG_LIST.this.GET_KEY(last = pos++); }

        @Override
        public KEY_GENERIC_TYPE PREV_KEY() { if (! hasPrevious()) throw new NoSuchElementException(); return
ABSTRACT_BIG_LIST.this.GET_KEY(last = --pos); }

        @Override
        public long nextIndex() { return pos; }

        @Override
        public long previousIndex() { return pos - 1; }

        @Override
        public void add(KEY_GENERIC_TYPE k) {
            ABSTRACT_BIG_LIST.this.add(pos++, k);
            last = -1;
        }

        @Override
        public void set(KEY_GENERIC_TYPE k) {
            if (last == -1) throw new IllegalStateException();
            ABSTRACT_BIG_LIST.this.set(last, k);
        }

        @Override
        public void remove() {
            if (last == -1) throw new IllegalStateException();
            ABSTRACT_BIG_LIST.this.REMOVE_KEY(last);
            /* If the last operation was a next(), we are removing an element *before* us, and we must decrease pos
correspondingly. */
            if (last < pos) pos--;
            last = -1;
        }
    };
}

```

```

    }
    };
}

/** Returns true if this list contains the specified element.
 * <p>This implementation delegates to { @code indexOf()}.
 * @see BigList#contains(Object)
 */
@Override
public boolean contains(final KEY_TYPE k) { return indexOf(k) >= 0; }

@Override
public long indexOf(final KEY_TYPE k) {
    final KEY_BIG_LIST_ITERATOR KEY_GENERIC i = listIterator();
    KEY_GENERIC_TYPE e;
    while(i.hasNext()) {
        e = i.NEXT_KEY();
        if (KEY_EQUALS(k, e)) return i.previousIndex();
    }
    return -1;
}

@Override
public long lastIndexOf(final KEY_TYPE k) {
    KEY_BIG_LIST_ITERATOR KEY_GENERIC i = listIterator(size64());
    KEY_GENERIC_TYPE e;
    while(i.hasPrevious()) {
        e = i.PREV_KEY();
        if (KEY_EQUALS(k, e)) return i.nextIndex();
    }
    return -1;
}

@Override
public void size(final long size) {
    long i = size64();
    if (size > i) while(i++ < size) add(KEY_NULL);
    else while(i-- != size) remove(i);
}

@Override
public BIG_LIST KEY_GENERIC subList(final long from, final long to) {
    ensureIndex(from);
    ensureIndex(to);
    if (from > to) throw new IndexOutOfBoundsException("Start index (" + from + ") is greater than end index (" + to +
    ")");
}

```

```

return new SUBLIST KEY_GENERIC_DIAMOND(this, from, to);
}

/** { @inheritDoc }
 *
 * <p>This is a trivial iterator-based implementation. It is expected that
 * implementations will override this method with a more optimized version.
 */
@Override
public void removeElements(final long from, final long to) {
    ensureIndex(to);
    KEY_BIG_LIST_ITERATOR KEY_GENERIC i = listIterator(from);
    long n = to - from;
    if (n < 0) throw new IllegalArgumentException("Start index (" + from + ") is greater than end index (" + to + ")");
    while(n-- != 0) {
        i.NEXT_KEY();
        i.remove();
    }
}

/** { @inheritDoc }
 *
 * <p>This is a trivial iterator-based implementation. It is expected that
 * implementations will override this method with a more optimized version.
 */
@Override
public void addElements(long index, final KEY_GENERIC_TYPE a[], long offset, long length) {
    ensureIndex(index);
    BIG_ARRAYS.ensureOffsetLength(a, offset, length);
    while(length-- != 0) add(index++, BIG_ARRAYS.get(a, offset++));
}

/** { @inheritDoc }
 *
 * <p>This implementation delegates to the analogous method for big-array fragments.
 */
@Override
public void addElements(final long index, final KEY_GENERIC_TYPE a[]) {
    addElements(index, a, 0, BIG_ARRAYS.length(a));
}

/** { @inheritDoc }
 *
 * <p>This is a trivial iterator-based implementation. It is expected that
 * implementations will override this method with a more optimized version.
 */
@Override
public void getElements(final long from, final KEY_TYPE a[], long offset, long length) {

```

```

    KEY_BIG_LIST_ITERATOR KEY_GENERIC i = listIterator(from);
    BIG_ARRAYS.ensureOffsetLength(a, offset, length);
    if (from + length > size64()) throw new IndexOutOfBoundsException("End index (" + (from + length) + ") is
greater than list size (" + size64() + ")");
    while(length-- != 0) BIG_ARRAYS.set(a, offset++, i.NEXT_KEY());
}

/** {@inheritDoc}
 * <p>This implementation delegates to {@link #removeElements(long, long)}.*/
@Override
public void clear() {
    removeElements(0, size64());
}

/** {@inheritDoc}
 *
 * <p>This implementation delegates to {@link #size64()}.
 * @deprecated Please use {@link #size64()} instead. */
@Override
@Deprecated
public int size() {
    return (int)Math.min(Integer.MAX_VALUE, size64());
}

#if ! KEY_CLASS_Reference
    private boolean valEquals(final Object a, final Object b) { return a == null ? b == null : a.equals(b); }
#endif

/** Returns the hash code for this big list, which is identical to {@link java.util.List#hashCode()}.
 *
 * @return the hash code for this big list.
 */
@Override
public int hashCode() {
    KEY_ITERATOR KEY_GENERIC i = iterator();
    int h = 1;
    long s = size64();
    while (s-- != 0) {
        KEY_GENERIC_TYPE k = i.NEXT_KEY();
        h = 31 * h + KEY2JAVAHASH(k);
    }
    return h;
}

@Override
public boolean equals(final Object o) {
    if (o == this) return true;
    if (!(o instanceof BigList)) return false;

```

```

final BigList<?> l = (BigList<?>)o;
long s = size64();
if (s != l.size64()) return false;

#if KEYS_PRIMITIVE
if (l instanceof BIG_LIST) {
    final KEY_BIG_LIST_ITERATOR KEY_GENERIC i1 = listIterator(), i2 = ((BIG_LIST
KEY_GENERIC)l).listIterator();
    while(s-- != 0) if (i1.NEXT_KEY() != i2.NEXT_KEY()) return false;
    return true;
}
#endif

final BigListIterator<?> i1 = listIterator(), i2 = l.listIterator();

#if KEY_CLASS_Reference
while(s-- != 0) if (i1.next() != i2.next()) return false;
#else
while(s-- != 0) if (! valEquals(i1.next(), i2.next())) return false;
#endif
return true;
}

#if ! KEY_CLASS_Reference
/** Compares this big list to another object. If the
 * argument is a {@link BigList}, this method performs a lexicographical comparison; otherwise,
 * it throws a {@code ClassCastException}.
 *
 * @param l a big list.
 * @return if the argument is a {@link BigList}, a negative integer,
 * zero, or a positive integer as this list is lexicographically less than, equal
 * to, or greater than the argument.
 * @throws ClassCastException if the argument is not a big list.
 */

```

SUPPRESS_WARNINGS_KEY_UNCHECKED

```

@Override
public int compareTo(final BigList<? extends KEY_GENERIC_CLASS> l) {
    if (l == this) return 0;

    if (l instanceof BIG_LIST) {

        final KEY_BIG_LIST_ITERATOR KEY_GENERIC i1 = listIterator(), i2 = ((BIG_LIST
KEY_GENERIC)l).listIterator();
        int r;
        KEY_GENERIC_TYPE e1, e2;

        while(i1.hasNext() && i2.hasNext()) {

```

```

    e1 = i1.NEXT_KEY();
    e2 = i2.NEXT_KEY();
    if ((r = KEY_CMP(e1, e2)) != 0) return r;
}
return i2.hasNext() ? -1 : (i1.hasNext() ? 1 : 0);
}

BigListIterator<? extends KEY_GENERIC_CLASS> i1 = listIterator(), i2 = l.listIterator();
int r;

while(i1.hasNext() && i2.hasNext()) {
    if ((r = ((Comparable<? super KEY_GENERIC_CLASS>)i1.next()).compareTo(i2.next())) != 0) return r;
}
return i2.hasNext() ? -1 : (i1.hasNext() ? 1 : 0);
}
#endif

@Override
public void push(KEY_GENERIC_TYPE o) {
    add(o);
}

@Override
public KEY_GENERIC_TYPE POP() {
    if (isEmpty()) throw new NoSuchElementException();
    return REMOVE_KEY(size64() - 1);
}

@Override
public KEY_GENERIC_TYPE TOP() {
    if (isEmpty()) throw new NoSuchElementException();
    return GET_KEY(size64() - 1);
}

@Override
public KEY_GENERIC_TYPE PEEK(int i) {
    return GET_KEY(size64() - 1 - i);
}

#if KEYS_PRIMITIVE
/** Removes a single instance of the specified element from this collection, if it is present (optional operation).
 * <p>This implementation delegates to { @code indexOf()}.
 * @see BigList#remove(Object)
 */
@Override
public boolean rem(KEY_TYPE k) {
    long index = indexOf(k);

```

```

    if (index == -1) return false;
    REMOVE_KEY(index);
    return true;
}

/** {@inheritDoc}
 *
 * <p>This implementation delegates to the type-specific version of {@link #addAll(long, Collection)}.
 */
@Override
public boolean addAll(final long index, final COLLECTION c) { return addAll(index, (Collection<? extends
KEY_CLASS>)c); }

/** {@inheritDoc}
 *
 * <p>This implementation delegates to the type-specific version of {@link #addAll(long, Collection)}.
 */
@Override
public boolean addAll(final long index, final BIG_LIST l) { return addAll(index, (COLLECTION)l); }

/** {@inheritDoc}
 *
 * <p>This implementation delegates to the type-specific version of {@link #addAll(long, Collection)}.
 */
@Override
public boolean addAll(final COLLECTION c) { return addAll(size64(), c); }

/** {@inheritDoc}
 *
 * <p>This implementation delegates to the type-specific list version of {@link #addAll(long, Collection)}.
 */
@Override
public boolean addAll(final BIG_LIST l) { return addAll(size64(), l); }

/** {@inheritDoc}
 *
 * <p>This implementation delegates to the corresponding type-specific method.
 * @deprecated Please use the corresponding type-specific method instead.
 */
@Deprecated
@Override
public void add(final long index, final KEY_CLASS ok) { add(index, ok.KEY_VALUE()); }

/** {@inheritDoc}
 *
 * <p>This implementation delegates to the corresponding type-specific method.
 * @deprecated Please use the corresponding type-specific method instead.
 */

```

```

@Deprecated
@Override
public KEY_CLASS set(final long index, final KEY_CLASS ok) { return KEY2OBJ(set(index,
ok.KEY_VALUE())); }

/** { @inheritDoc }
*
* <p>This implementation delegates to the corresponding type-specific method.
* @deprecated Please use the corresponding type-specific method instead.
*/
@Deprecated
@Override
public KEY_CLASS get(final long index) { return KEY2OBJ(GET_KEY(index)); }

/** { @inheritDoc }
*
* <p>This implementation delegates to the corresponding type-specific method.
* @deprecated Please use the corresponding type-specific method instead.
*/
@Deprecated
@Override
public long indexOf(final Object ok) { return indexOf(KEY_OBJ2TYPE(ok)); }

/** { @inheritDoc }
*
* <p>This implementation delegates to the corresponding type-specific method.
* @deprecated Please use the corresponding type-specific method instead.
*/
@Deprecated
@Override
public long lastIndexOf(final Object ok) { return lastIndexOf(KEY_OBJ2TYPE(ok)); }

/** { @inheritDoc }
*
* <p>This implementation delegates to the corresponding type-specific method.
* @deprecated Please use the corresponding type-specific method instead.
*/
@Deprecated
@Override
public KEY_CLASS remove(final long index) { return KEY2OBJ(REMOVE_KEY(index)); }

/** { @inheritDoc }
*
* <p>This implementation delegates to the corresponding type-specific method.
* @deprecated Please use the corresponding type-specific method instead.
*/
@Deprecated
@Override

```

```

public void push(KEY_CLASS o) { push(o.KEY_VALUE()); }

/** { @inheritDoc}
 *
 * <p>This implementation delegates to the corresponding type-specific method.
 * @deprecated Please use the corresponding type-specific method instead.
 */
@Deprecated
@Override
public KEY_CLASS pop() { return KEY_CLASS.valueOf(POP()); }

/** { @inheritDoc}
 *
 * <p>This implementation delegates to the corresponding type-specific method.
 * @deprecated Please use the corresponding type-specific method instead.
 */
@Deprecated
@Override
public KEY_CLASS top() { return KEY_CLASS.valueOf(TOP()); }

/** { @inheritDoc}
 *
 * <p>This implementation delegates to the corresponding type-specific method.
 * @deprecated Please use the corresponding type-specific method instead.
 */
@Deprecated
@Override
public KEY_CLASS peek(int i) { return KEY_CLASS.valueOf(PEEK(i)); }

#endif

@Override
public String toString() {
    final StringBuilder s = new StringBuilder();
    final KEY_ITERATOR KEY_GENERIC i = iterator();
    long n = size64();
    KEY_GENERIC_TYPE k;
    boolean first = true;

    s.append("[");

    while(n-- != 0) {
        if (first) first = false;
        else s.append(", ");
        k = i.NEXT_KEY();
        #if KEYS_REFERENCE
            if (this == k) s.append("(this big list)"); else
        #endif
    }
}

```

```

        s.append(String.valueOf(k));
    }

    s.append("]");
    return s.toString();
}

/** A class implementing a sublist view. */
public static class SUBLIST_KEY_GENERIC extends ABSTRACT_BIG_LIST_KEY_GENERIC implements
java.io.Serializable {
    private static final long serialVersionUID = -7046029254386353129L;
    /** The list this sublist restricts. */
    protected final BIG_LIST_KEY_GENERIC l;
    /** Initial (inclusive) index of this sublist. */
    protected final long from;
    /** Final (exclusive) index of this sublist. */
    protected long to;

    public SUBLIST(final BIG_LIST_KEY_GENERIC l, final long from, final long to) {
        this.l = l;
        this.from = from;
        this.to = to;
    }

    private boolean assertRange() {
        assert from <= l.size64();
        assert to <= l.size64();
        assert to >= from;
        return true;
    }

    @Override
    public boolean add(final KEY_GENERIC_TYPE k) {
        l.add(to, k);
        to++;
        assert assertRange();
        return true;
    }

    @Override
    public void add(final long index, final KEY_GENERIC_TYPE k) {
        ensureIndex(index);
        l.add(from + index, k);
        to++;
        assert assertRange();
    }
}

```

```

@Override
public boolean addAll(final long index, final Collection<? extends KEY_GENERIC_CLASS> c) {
    ensureIndex(index);
    to += c.size();
    return l.addAll(from + index, c);
}

```

```

@Override
public KEY_GENERIC_TYPE GET_KEY(long index) {
    ensureRestrictedIndex(index);
    return l.GET_KEY(from + index);
}

```

```

@Override
public KEY_GENERIC_TYPE REMOVE_KEY(long index) {
    ensureRestrictedIndex(index);
    to--;
    return l.REMOVE_KEY(from + index);
}

```

```

@Override
public KEY_GENERIC_TYPE set(long index, KEY_GENERIC_TYPE k) {
    ensureRestrictedIndex(index);
    return l.set(from + index, k);
}

```

```

@Override
public long size64() { return to - from; }

```

```

@Override
public void getElements(final long from, final KEY_TYPE[][] a, final long offset, final long length) {
    ensureIndex(from);
    if (from + length > size64()) throw new IndexOutOfBoundsException("End index (" + from + length + ") is greater than list size (" + size64() + ")");
    l.getElements(this.from + from, a, offset, length);
}

```

```

@Override
public void removeElements(final long from, final long to) {
    ensureIndex(from);
    ensureIndex(to);
    l.removeElements(this.from + from, this.from + to);
    this.to -= (to - from);
    assert assertRange();
}

```

```

@Override
public void addElements(final long index, final KEY_GENERIC_TYPE a[], long offset, long length) {

```

```

ensureIndex(index);
l.addElements(this.from + index, a, offset, length);
this.to += length;
assert assertRange();
}

@Override
public KEY_BIG_LIST_ITERATOR KEY_GENERIC listIterator(final long index) {
ensureIndex(index);

return new KEY_BIG_LIST_ITERATOR KEY_GENERIC() {
long pos = index, last = -1;

@Override
public boolean hasNext() { return pos < size64(); }
@Override
public boolean hasPrevious() { return pos > 0; }
@Override
public KEY_GENERIC_TYPE NEXT_KEY() { if (! hasNext()) throw new NoSuchElementException(); return
l.GET_KEY(from + (last = pos++)); }
@Override
public KEY_GENERIC_TYPE PREV_KEY() { if (! hasPrevious()) throw new NoSuchElementException(); return
l.GET_KEY(from + (last = --pos)); }
@Override
public long nextIndex() { return pos; }
@Override
public long previousIndex() { return pos - 1; }
@Override
public void add(KEY_GENERIC_TYPE k) {
if (last == -1) throw new IllegalStateException();
SUBLIST.this.add(pos++, k);
last = -1;
assert assertRange();
}
@Override
public void set(KEY_GENERIC_TYPE k) {
if (last == -1) throw new IllegalStateException();
SUBLIST.this.set(last, k);
}
@Override
public void remove() {
if (last == -1) throw new IllegalStateException();
SUBLIST.this.REMOVE_KEY(last);
/* If the last operation was a next(), we are removing an element *before* us, and we must decrease pos
correspondingly. */
if (last < pos) pos--;
last = -1;
assert assertRange();
}
}
}

```

```

    }
};
}

@Override
public BIG_LIST KEY_GENERIC subList(final long from, final long to) {
    ensureIndex(from);
    ensureIndex(to);
    if (from > to) throw new IllegalArgumentException("Start index (" + from + ") is greater than end index (" + to +
    "");

    return new SUBLIST KEY_GENERIC_DIAMOND(this, from, to);
}

#if KEYS_PRIMITIVE
@Override
public boolean rem(KEY_TYPE k) {
    long index = indexOf(k);
    if (index == -1) return false;
    to--;
    l.REMOVE_KEY(from + index);
    assert assertRange();
    return true;
}

@Override
public boolean addAll(final long index, final COLLECTION c) {
    ensureIndex(index);
    return super.addAll(index, c);
}

@Override
public boolean addAll(final long index, final BIG_LIST l) {
    ensureIndex(index);
    return super.addAll(index, l);
}
#endif
}
}

```

Found in path(s):

```

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-
a775574/drv/AbstractBigList.drv

```

No license file was found, but licenses were detected in source scan.

```

/*

```

```

* Copyright (C) 2002-2017 Sebastiano Vigna

```

```

*

```

- * Licensed under the Apache License, Version 2.0 (the "License");
- * you may not use this file except in compliance with the License.
- * You may obtain a copy of the License at
- *
- * <http://www.apache.org/licenses/LICENSE-2.0>
- *
- * Unless required by applicable law or agreed to in writing, software
- * distributed under the License is distributed on an "AS IS" BASIS,
- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.
- */

package PACKAGE;

```
import it.unimi.dsi.fastutil.BigArrays;
import it.unimi.dsi.fastutil.Hash;
import it.unimi.dsi.fastutil.Size64;
import it.unimi.dsi.fastutil.HashCommon;
import static it.unimi.dsi.fastutil.HashCommon.bigArraySize;
import static it.unimi.dsi.fastutil.HashCommon.maxFill;
```

```
import java.util.Collection;
import java.util.Iterator;
import java.util.NoSuchElementException;
```

```
/** A type-specific hash big set with with a fast, small-footprint implementation.
 *
 * <p>Instances of this class use a hash table to represent a big set: the number
 * of elements in the set is limited only by the amount of core memory. The table
 * (backed by a { @linkplain it.unimi.dsi.fastutil.BigArrays big array}) is
 * filled up to a specified <em>load factor</em>, and then doubled in size to
 * accommodate new entries. If the table is emptied below <em>one fourth</em>
 * of the load factor, it is halved in size; however, the table is never reduced to a
 * size smaller than that at creation time: this approach makes it
 * possible to create sets with a large capacity in which insertions and
 * deletions do not cause immediately rehashing. Moreover, halving is
 * not performed when deleting entries from an iterator, as it would interfere
 * with the iteration process.
 *
 * <p>Note that { @link #clear()} does not modify the hash table size.
 * Rather, a family of { @linkplain #trim() trimming
 * methods} lets you control the size of the table; this is particularly useful
 * if you reuse instances of this class.
 *
 * <p>The methods of this class are about 30% slower than those of the corresponding non-big set.
```

```

*
* @see Hash
* @see HashCommon
*/

public class OPEN_HASH_BIG_SET KEY_GENERIC extends ABSTRACT_SET KEY_GENERIC implements
java.io.Serializable, Cloneable, Hash, Size64 {

    private static final long serialVersionUID = 0L;
    private static final boolean ASSERTS = ASSERTS_VALUE;

    /** The big array of keys. */
    protected transient KEY_GENERIC_TYPE[][] key;

    /** The mask for wrapping a position counter. */
    protected transient long mask;

    /** The mask for wrapping a segment counter. */
    protected transient int segmentMask;

    /** The mask for wrapping a base counter. */
    protected transient int baseMask;

    /** Whether this set contains the null key. */
    protected transient boolean containsNull;

    /** The current table size (always a power of 2). */
    protected transient long n;

    /** Threshold after which we rehash. It must be the table size times {@link #f}. */
    protected transient long maxFill;

    /** We never resize below this threshold, which is the construction-time {@link #n}. */
    protected final transient long minN;

    /** The acceptable load factor. */
    protected final float f;

    /** Number of entries in the set. */
    protected long size;

    /** Initialises the mask values. */
    private void initMasks() {
        mask = n - 1;
        /* Note that either we have more than one segment, and in this case all segments
        * are BigArrays.SEGMENT_SIZE long, or we have exactly one segment whose length
        * is a power of two. */
    }

```

```

segmentMask = key[0].length - 1;
baseMask = key.length - 1;
}

/** Creates a new hash big set.
 *
 * <p>The actual table size will be the least power of two greater than { @code expected}/{ @code f}.
 *
 * @param expected the expected number of elements in the set.
 * @param f the load factor.
 */
SUPPRESS_WARNINGS_KEY_UNCHECKED
public OPEN_HASH_BIG_SET(final long expected, final float f) {
    if (f <= 0 || f > 1) throw new IllegalArgumentException("Load factor must be greater than 0 and smaller than or
equal to 1");
    if (n < 0) throw new IllegalArgumentException("The expected number of elements must be nonnegative");

    this.f = f;

    minN = n = bigArraySize(expected, f);
    maxFill = maxFill(n, f);
    key = KEY_GENERIC_BIG_ARRAY_CAST BIG_ARRAYS.newBigArray(n);
    initMasks();
}

/** Creates a new hash big set with { @link Hash#DEFAULT_LOAD_FACTOR} as load factor.
 *
 * @param expected the expected number of elements in the hash big set.
 */

public OPEN_HASH_BIG_SET(final long expected) {
    this(expected, DEFAULT_LOAD_FACTOR);
}

/** Creates a new hash big set with initial expected { @link Hash#DEFAULT_INITIAL_SIZE} elements
 * and { @link Hash#DEFAULT_LOAD_FACTOR} as load factor.
 */

public OPEN_HASH_BIG_SET() {
    this(DEFAULT_INITIAL_SIZE, DEFAULT_LOAD_FACTOR);
}

/** Creates a new hash big set copying a given collection.
 *
 * @param c a { @link Collection} to be copied into the new hash big set.
 * @param f the load factor.
 */

```

```

public OPEN_HASH_BIG_SET(final Collection<? extends KEY_GENERIC_CLASS> c, final float f) {
    this(c.size(), f);
    addAll(c);
}

/** Creates a new hash big set with {@link Hash#DEFAULT_LOAD_FACTOR} as load factor
 * copying a given collection.
 *
 * @param c a {@link Collection} to be copied into the new hash big set.
 */

public OPEN_HASH_BIG_SET(final Collection<? extends KEY_GENERIC_CLASS> c) {
    this(c, DEFAULT_LOAD_FACTOR);
}

/** Creates a new hash big set copying a given type-specific collection.
 *
 * @param c a type-specific collection to be copied into the new hash big set.
 * @param f the load factor.
 */

public OPEN_HASH_BIG_SET(final COLLECTION KEY_EXTENDS_GENERIC c, final float f) {
    this(c.size(), f);
    addAll(c);
}

/** Creates a new hash big set with {@link Hash#DEFAULT_LOAD_FACTOR} as load factor
 * copying a given type-specific collection.
 *
 * @param c a type-specific collection to be copied into the new hash big set.
 */

public OPEN_HASH_BIG_SET(final COLLECTION KEY_EXTENDS_GENERIC c) {
    this(c, DEFAULT_LOAD_FACTOR);
}

/** Creates a new hash big set using elements provided by a type-specific iterator.
 *
 * @param i a type-specific iterator whose elements will fill the new hash big set.
 * @param f the load factor.
 */

public OPEN_HASH_BIG_SET(final STD_KEY_ITERATOR KEY_EXTENDS_GENERIC i, final float f) {
    this(DEFAULT_INITIAL_SIZE, f);
    while(i.hasNext()) add(i.NEXT_KEY());
}

```

```
/** Creates a new hash big set with { @link Hash#DEFAULT_LOAD_FACTOR } as load factor using elements
provided by a type-specific iterator.
*
* @param i a type-specific iterator whose elements will fill the new hash big set.
*/
```

```
public OPEN_HASH_BIG_SET(final STD_KEY_ITERATOR KEY_EXTENDS_GENERIC i) {
    this(i, DEFAULT_LOAD_FACTOR);
}
```

```
#if KEYS_PRIMITIVE
```

```
/** Creates a new hash big set using elements provided by an iterator.
*
* @param i an iterator whose elements will fill the new hash big set.
* @param f the load factor.
*/
```

```
public OPEN_HASH_BIG_SET(final Iterator<?> i, final float f) {
    this(ITERATORS.AS_KEY_ITERATOR(i), f);
}
```

```
/** Creates a new hash big set with { @link Hash#DEFAULT_LOAD_FACTOR } as load factor using elements
provided by an iterator.
*
* @param i an iterator whose elements will fill the new hash big set.
*/
```

```
public OPEN_HASH_BIG_SET(final Iterator<?> i) {
    this(ITERATORS.AS_KEY_ITERATOR(i));
}
```

```
#endif
```

```
/** Creates a new hash big set and fills it with the elements of a given array.
*
* @param a an array whose elements will be used to fill the new hash big set.
* @param offset the first element to use.
* @param length the number of elements to use.
* @param f the load factor.
*/
```

```
public OPEN_HASH_BIG_SET(final KEY_GENERIC_TYPE[] a, final int offset, final int length, final float f) {
    this(length < 0 ? 0 : length, f);
    ARRAYS.ensureOffsetLength(a, offset, length);
    for(int i = 0; i < length; i++) add(a[offset + i]);
}
```

```
/** Creates a new hash big set with {@link Hash#DEFAULT_LOAD_FACTOR} as load factor and fills it with the elements of a given array.
```

```
*
```

```
* @param a an array whose elements will be used to fill the new hash big set.
```

```
* @param offset the first element to use.
```

```
* @param length the number of elements to use.
```

```
*/
```

```
public OPEN_HASH_BIG_SET(final KEY_GENERIC_TYPE[] a, final int offset, final int length) {  
    this(a, offset, length, DEFAULT_LOAD_FACTOR);  
}
```

```
/** Creates a new hash big set copying the elements of an array.
```

```
*
```

```
* @param a an array to be copied into the new hash big set.
```

```
* @param f the load factor.
```

```
*/
```

```
public OPEN_HASH_BIG_SET(final KEY_GENERIC_TYPE[] a, final float f) {  
    this(a, 0, a.length, f);  
}
```

```
/** Creates a new hash big set with {@link Hash#DEFAULT_LOAD_FACTOR} as load factor
```

```
* copying the elements of an array.
```

```
*
```

```
* @param a an array to be copied into the new hash big set.
```

```
*/
```

```
public OPEN_HASH_BIG_SET(final KEY_GENERIC_TYPE[] a) {  
    this(a, DEFAULT_LOAD_FACTOR);  
}
```

```
private long realSize() {  
    return containsNull ? size - 1 : size;  
}
```

```
private void ensureCapacity(final long capacity) {  
    final long needed = bigArraySize(capacity, f);  
    if (needed > n) rehash(needed);  
}
```

```
@Override
```

```
public boolean addAll(Collection<? extends KEY_GENERIC_CLASS> c) {  
    final long size = c instanceof Size64 ? ((Size64)c).size64() : c.size();  
    // The resulting collection will be at least c.size() big  
    if (f <= .5) ensureCapacity(size); // The resulting collection will be sized for c.size() elements  
    else ensureCapacity(size64() + size); // The resulting collection will be sized for size() + c.size() elements
```

```

return super.addAll(c);
}

#if KEYS_PRIMITIVE
@Override
public boolean addAll(COLLECTION c) {
    final long size = c instanceof Size64 ? ((Size64)c).size64() : c.size();
    if (f <= .5) ensureCapacity(size); // The resulting collection will be size for c.size() elements
    else ensureCapacity(size64() + size); // The resulting collection will be sized for size() + c.size() elements
    return super.addAll(c);
}
#endif

@Override
public boolean add(final KEY_GENERIC_TYPE k) {
    int displ, base;

    if (KEY_IS_NULL(k)) {
        if (containsNull) return false;
        containsNull = true;
    }
    else {
        KEY_GENERIC_TYPE curr;
        final KEY_GENERIC_TYPE[][] key = this.key;
        final long h = KEY2LONGHASH(k);

        // The starting point.
        if (! KEY_IS_NULL(curr = key[base = (int)((h & mask) >>> BigArrays.SEGMENT_SHIFT)][displ = (int)(h &
segmentMask)])) {
            if (KEY_EQUALS_NOT_NULL(curr, k)) return false;
            while(! KEY_IS_NULL(curr = key[base = (base + ((displ = (displ + 1) & segmentMask) == 0 ? 1 : 0)) &
baseMask][displ]))
                if (KEY_EQUALS_NOT_NULL(curr, k)) return false;
        }

        key[base][displ] = k;
    }

    if (size++ >= maxFill) rehash(2 * n);
    if (ASSERTS) checkTable();
    return true;
}

#if KEY_CLASS_Object
/** Add a random element if not present, get the existing value if already present.
 *
 * This is equivalent to (but faster than) doing a:
 * <pre>

```

```

* K exist = set.get(k);
* if (exist == null) {
*   set.add(k);
*   exist = k;
* }
* </pre>
*/

public KEY_GENERIC_TYPE addOrGet(final KEY_GENERIC_TYPE k) {
    int displ, base;

    if (KEY_IS_NULL(k)) {
        if (containsNull) return null;
        containsNull = true;
    }
    else {
        KEY_GENERIC_TYPE curr;
        final KEY_GENERIC_TYPE[][] key = this.key;
        final long h = KEY2LONGHASH(k);

        // The starting point.
        if (! KEY_IS_NULL(curr = key[base = (int)((h & mask) >>> BigArrays.SEGMENT_SHIFT)][displ = (int)(h &
segmentMask)])) {
            if (KEY_EQUALS_NOT_NULL(curr, k)) return curr;
            while(! KEY_IS_NULL(curr = key[base = (base + ((displ = (displ + 1) & segmentMask) == 0 ? 1 : 0)) &
baseMask][displ]))
                if (KEY_EQUALS_NOT_NULL(curr, k)) return curr;
        }

        key[base][displ] = k;
    }

    if (size++ >= maxFill) rehash(2 * n);
    if (ASSERTS) checkTable();
    return k;
}
#endif

/** Shifts left entries with the specified hash code, starting at the specified position,
 * and empties the resulting free entry.
 *
 * @param pos a starting position.
 */
protected final void shiftKeys(long pos) {
    // Shift entries with the same hash.
    long last, slot;
    final KEY_GENERIC_TYPE[][] key = this.key;

    for(;;) {

```

```

pos = ((last = pos) + 1) & mask;

for(;;) {
    if (KEY_IS_NULL(BIG_ARRAYS.get(key, pos))) {
        BIG_ARRAYS.set(key, last, KEY_NULL);
        return;
    }
    slot = KEY2LONGHASH(BIG_ARRAYS.get(key, pos)) & mask;
    if (last <= pos ? last >= slot || slot > pos : last >= slot && slot > pos) break;
    pos = (pos + 1) & mask;
}

BIG_ARRAYS.set(key, last, BIG_ARRAYS.get(key, pos));
}
}

private boolean removeEntry(final int base, final int displ) {
    size--;
    shiftKeys(base * (long)BigArrays.SEGMENT_SIZE + displ);
    if (n > minN && size < maxFill / 4 && n > DEFAULT_INITIAL_SIZE) rehash(n / 2);
    return true;
}

private boolean removeNullEntry() {
    containsNull = false;
    size--;
    if (n > minN && size < maxFill / 4 && n > DEFAULT_INITIAL_SIZE) rehash(n / 2);
    return true;
}

@Override
public boolean remove(final KEY_TYPE k) {
    if (KEY_IS_NULL(k)) {
        if (containsNull) return removeNullEntry();
        return false;
    }
}

KEY_GENERIC_TYPE curr;
final KEY_GENERIC_TYPE[][] key = this.key;
final long h = KEY2LONGHASH(k);
int displ, base;

// The starting point.
if (KEY_IS_NULL(curr = key[base = (int)((h & mask) >>> BigArrays.SEGMENT_SHIFT)][displ = (int)(h &
segmentMask)])) return false;
if (KEY_EQUALS_NOT_NULL(curr, k)) return removeEntry(base, displ);
while(true) {
    if (KEY_IS_NULL(curr = key[base = (base + ((displ = (displ + 1) & segmentMask) == 0 ? 1 : 0)) &

```

```

baseMask][displ])) return false;
    if (KEY_EQUALS_NOT_NULL(curr, k)) return removeEntry(base, displ);
}
}

@Override
public boolean contains(final KEY_TYPE k) {
    if (KEY_IS_NULL(k)) return containsNull;

    KEY_GENERIC_TYPE curr;
    final KEY_GENERIC_TYPE[][] key = this.key;
    final long h = KEY2LONGHASH(k);
    int displ, base;

    // The starting point.
    if (KEY_IS_NULL(curr = key[base = (int)((h & mask) >>> BigArrays.SEGMENT_SHIFT)][displ = (int)(h &
segmentMask)])) return false;
    if (KEY_EQUALS_NOT_NULL(curr, k)) return true;
    while(true) {
        if (KEY_IS_NULL(curr = key[base = (base + ((displ = (displ + 1) & segmentMask) == 0 ? 1 : 0)) &
baseMask][displ])) return false;
        if (KEY_EQUALS_NOT_NULL(curr, k)) return true;
    }
}

#if KEY_CLASS_Object
/** Returns the element of this set that is equal to the given key, or { @code null}.
 * @return the element of this set that is equal to the given key, or { @code null}.
 */
public K get(final KEY_TYPE k) {
    if (k == null) return null; // This is correct independently of the value of containsNull

    KEY_GENERIC_TYPE curr;
    final KEY_GENERIC_TYPE[][] key = this.key;
    final long h = KEY2LONGHASH(k);
    int displ, base;

    // The starting point.
    if (KEY_IS_NULL(curr = key[base = (int)((h & mask) >>> BigArrays.SEGMENT_SHIFT)][displ = (int)(h &
segmentMask)])) return null;
    if (KEY_EQUALS_NOT_NULL(curr, k)) return curr;
    while(true) {
        if (KEY_IS_NULL(curr = key[base = (base + ((displ = (displ + 1) & segmentMask) == 0 ? 1 : 0)) &
baseMask][displ])) return null;
        if (KEY_EQUALS_NOT_NULL(curr, k)) return curr;
    }
}
#endif

```

```

/* Removes all elements from this set.
 *
 */

/** {@inheritDoc}
 *
 * <p>To increase object reuse, this method does not change the table size.
 * If you want to reduce the table size, you must use {@link #trim(long)}.
 */
@Override
public void clear() {
    if (size == 0) return;
    size = 0;
    containsNull = false;
    BIG_ARRAYS.fill(key, KEY_NULL);
}

/** An iterator over a hash big set. */

private class SetIterator implements KEY_ITERATOR KEY_GENERIC {
    /** The base of the last entry returned, if positive or zero; initially, the number of components
     * of the key array. If negative, the last element returned was
     * that of index {@code - base - 1} from the {@link #wrapped} list. */
    int base = key.length;
    /** The displacement of the last entry returned; initially, zero. */
    int displ;
    /** The index of the last entry that has been returned (or {@link Long#MIN_VALUE} if {@link #base} is
     * negative).
     * It is -1 if either we did not return an entry yet, or the last returned entry has been removed. */
    long last = -1;
    /** A downward counter measuring how many entries must still be returned. */
    long c = size;
    /** A boolean telling us whether we should return the null key. */
    boolean mustReturnNull = OPEN_HASH_BIG_SET.this.containsNull;
    /** A lazily allocated list containing elements that have wrapped around the table because of removals. */
    ARRAY_LIST KEY_GENERIC wrapped;

    @Override
    public boolean hasNext() { return c != 0; }

    @Override
    public KEY_GENERIC_TYPE NEXT_KEY() {
        if (!hasNext()) throw new NoSuchElementException();
        c--;
    }
}

```

```

if (mustReturnNull) {
    mustReturnNull = false;
    last = n;
    return KEY_NULL;
}

final KEY_GENERIC_TYPE[][] key = OPEN_HASH_BIG_SET.this.key;

for(;;) {
    if (displ == 0 && base <= 0) {
        // We are just enumerating elements from the wrapped list.
        last = Long.MIN_VALUE;
        return wrapped.GET_KEY(- (--base) - 1);
    }

    if (displ-- == 0) displ = key[--base].length - 1;

    final KEY_GENERIC_TYPE k = key[base][displ];
    if (! KEY_IS_NULL(k)) {
        last = base * (long)BigArrays.SEGMENT_SIZE + displ;
        return k;
    }
}

/** Shifts left entries with the specified hash code, starting at the specified position,
 * and empties the resulting free entry.
 *
 * @param pos a starting position.
 */
private final void shiftKeys(long pos) {
    // Shift entries with the same hash.
    long last, slot;
    KEY_GENERIC_TYPE curr;
    final KEY_GENERIC_TYPE[][] key = OPEN_HASH_BIG_SET.this.key;

    for(;;) {
        pos = ((last = pos) + 1) & mask;

        for(;;) {
            if(KEY_IS_NULL(curr = BIG_ARRAYS.get(key, pos))) {
                BIG_ARRAYS.set(key, last, KEY_NULL);
                return;
            }
            slot = KEY2LONGHASH(curr) & mask;
            if (last <= pos ? last >= slot || slot > pos : last >= slot && slot > pos) break;
            pos = (pos + 1) & mask;
        }
    }
}

```

```

    if (pos < last) { // Wrapped entry.
        if (wrapped == null) wrapped = new ARRAY_LIST KEY_GENERIC_DIAMOND();
        wrapped.add(BIG_ARRAYS.get(key, pos));
    }

    BIG_ARRAYS.set(key, last, curr);
}

@Override
public void remove() {
    if (last == -1) throw new IllegalStateException();
    if (last == n) OPEN_HASH_BIG_SET.this.containsNull = false;
    else if (base >= 0) shiftKeys(last);
    else {
        // We're removing wrapped entries.
#ifdef KEYS_REFERENCE
        OPEN_HASH_BIG_SET.this.remove(wrapped.set(- base - 1, null));
#else
        OPEN_HASH_BIG_SET.this.remove(wrapped.GET_KEY(- base - 1));
#endif
        last = -1; // Note that we must not decrement size
        return;
    }

    size--;
    last = -1; // You can no longer remove this entry.
    if (ASSERTS) checkTable();
}

@Override
public KEY_ITERATOR KEY_GENERIC iterator() {
    return new SetIterator();
}

/** Reshapes this set, making the table as small as possible.
 *
 * <p>This method reshapes the table to the smallest size satisfying the
 * load factor. It can be used when the set will not be changed anymore, so
 * to optimize access speed and size.
 *
 * <p>If the table size is already the minimum possible, this method
 * does nothing.
 *
 * @return true if there was enough memory to trim the set.

```

```

* @see #trim(long)
*/

public boolean trim() {
    final long l = bigArraySize(size, f);
    if (l >= n || size > maxFill(l, f)) return true;
    try {
        rehash(l);
    }
    catch(OutOfMemoryError cantDoIt) { return false; }
    return true;
}

/** Reshapes this set if the table is too large.
 *
 * <p>Let <var>N</var> be the smallest table size that can hold
 * <code>max(n, {@link #size64()})</code> entries, still satisfying the load factor. If the current
 * table size is smaller than or equal to <var>N</var>, this method does
 * nothing. Otherwise, it reshapes this set in a table of size
 * <var>N</var>.
 *
 * <p>This method is useful when reusing sets. {@linkplain #clear() Clearing a
 * set} leaves the table size untouched. If you are reusing a set
 * many times, you can call this method with a typical
 * size to avoid keeping around a very large table just
 * because of a few large transient sets.
 *
 * @param n the threshold for the trimming.
 * @return true if there was enough memory to trim the set.
 * @see #trim()
 */

public boolean trim(final long n) {
    final long l = bigArraySize(n, f);
    if (this.n <= l) return true;
    try {
        rehash(l);
    }
    catch(OutOfMemoryError cantDoIt) { return false; }
    return true;
}

/** Resizes the set.
 *
 * <p>This method implements the basic rehashing strategy, and may be
 * overridden by subclasses implementing different rehashing strategies (e.g.,
 * disk-based rehashing). However, you should not override this method
 * unless you understand the internal workings of this class.

```

```

*
* @param newN the new size
*/

SUPPRESS_WARNINGS_KEY_UNCHECKED
protected void rehash(final long newN) {
    final KEY_GENERIC_TYPE key[][] = this.key;
    final KEY_GENERIC_TYPE newKey[][] = KEY_GENERIC_BIG_ARRAY_CAST
BIG_ARRAYS.newBigArray(newN);
    final long mask = newN - 1; // Note that this is used by the hashing macro
    final int newSegmentMask = newKey[0].length - 1;
    final int newBaseMask = newKey.length - 1;

    int base = 0, displ = 0, b, d;
    long h;
    KEY_GENERIC_TYPE k;

    for(long i = realSize(); i-- != 0;) {

        while(KEY_IS_NULL(key[base][displ])) base = (base + ((displ = (displ + 1) & segmentMask) == 0 ? 1 : 0));

        k = key[base][displ];
        h = KEY2LONGHASH(k);

        // The starting point.
        if (! KEY_IS_NULL(newKey[b = (int)((h & mask) >>> BigArrays.SEGMENT_SHIFT)][d = (int)(h &
newSegmentMask)]))
            while(! KEY_IS_NULL(newKey[b = (b + ((d = (d + 1) & newSegmentMask) == 0 ? 1 : 0)) & newBaseMask][d]));

        newKey[b][d] = k;

        base = (base + ((displ = (displ + 1) & segmentMask) == 0 ? 1 : 0));
    }

    this.n = newN;
    this.key = newKey;
    initMasks();
    maxFill = maxFill(n, f);
}

@Deprecated
@Override
public int size() {
    return (int)Math.min(Integer.MAX_VALUE, size);
}

@Override
public long size64() {

```

```
return size;
}
```

```
@Override
public boolean isEmpty() {
    return size == 0;
}
```

```
/** Returns a deep copy of this big set.
```

```
*
```

```
* <p>This method performs a deep copy of this big hash set; the data stored in the  
* set, however, is not cloned. Note that this makes a difference only for object keys.
```

```
*
```

```
* @return a deep copy of this big set.
```

```
*/
```

```
@Override
SUPPRESS_WARNINGS_KEY_UNCHECKED
public OPEN_HASH_BIG_SET KEY_GENERIC clone() {
    OPEN_HASH_BIG_SET KEY_GENERIC c;
    try {
        c = (OPEN_HASH_BIG_SET KEY_GENERIC)super.clone();
    }
    catch(CloneNotSupportedException cantHappen) {
        throw new InternalError();
    }
    c.key = BIG_ARRAYS.copy(key);
    c.containsNull = containsNull;
    return c;
}
```

```
/** Returns a hash code for this set.
```

```
*
```

```
* This method overrides the generic method provided by the superclass.
```

```
* Since { @code equals()} is not overridden, it is important
```

```
* that the value returned by this method is the same value as
```

```
* the one returned by the overridden method.
```

```
*
```

```
* @return a hash code for this set.
```

```
*/
```

```
@Override
public int hashCode() {
    final KEY_GENERIC_TYPE key[][] = this.key;
    int h = 0, base = 0, displ = 0;

    for(long j = realSize(); j-- != 0;) {
```

```

    while(KEY_IS_NULL(key[base][displ])) base = (base + ((displ = (displ + 1) & segmentMask) == 0 ? 1 : 0));
#if KEYS_REFERENCE
    if (this != key[base][displ])
#endif
    h += KEY2JAVAHASH_NOT_NULL(key[base][displ]);
    base = (base + ((displ = (displ + 1) & segmentMask) == 0 ? 1 : 0));
}
return h;
}

```

```

private void writeObject(java.io.ObjectOutputStream s) throws java.io.IOException {
    final KEY_ITERATOR KEY_GENERIC i = iterator();
    s.defaultWriteObject();
    for(long j = size; j-- != 0;) s.WRITE_KEY(i.NEXT_KEY());
}

```

SUPPRESS_WARNINGS_KEY_UNCHECKED

```

private void readObject(java.io.ObjectInputStream s) throws java.io.IOException, ClassNotFoundException {
    s.defaultReadObject();

```

```

    n = bigArraySize(size, f);
    maxFill = maxFill(n, f);

```

```

    final KEY_GENERIC_TYPE[][] key = this.key = KEY_GENERIC_BIG_ARRAY_CAST
    BIG_ARRAYS.newBigArray(n);

```

```

    initMasks();

```

```

    long h;
    KEY_GENERIC_TYPE k;
    int base, displ;

```

```

    for(long i = size; i-- != 0;) {
        k = KEY_GENERIC_CAST s.READ_KEY();

```

```

        if (KEY_IS_NULL(k)) containsNull = true;
        else {
            h = KEY2LONGHASH(k);
            if (! KEY_IS_NULL(key[base = (int)((h & mask) >>> BigArrays.SEGMENT_SHIFT)][displ = (int)(h &
segmentMask)]))
                while(! KEY_IS_NULL(key[base = (base + ((displ = (displ + 1) & segmentMask) == 0 ? 1 : 0)) &
baseMask][displ]));
            key[base][displ] = k;
        }
    }
}

```

```

if (ASSERTS) checkTable();
}

#ifdef ASSERTS_CODE
private void checkTable() {
    assert (n & -n) == n : "Table length is not a power of two: " + n;
    assert n == BIG_ARRAYS.length(key);
    long n = this.n;
    while(n-- != 0)
        if (! KEY_IS_NULL(BIG_ARRAYS.get(key, n)) && ! contains(BIG_ARRAYS.get(key, n)))
            throw new AssertionError("Hash table has key " + BIG_ARRAYS.get(key, n) + " marked as occupied, but the key
does not belong to the table");

#ifdef KEYS_PRIMITIVE
    java.util.HashSet<KEY_GENERIC_CLASS> s = new java.util.HashSet<KEY_GENERIC_CLASS> ();
#else
    java.util.HashSet<Object> s = new java.util.HashSet<Object>();
#endif

    for(long i = size(); i-- != 0;)
        if (! KEY_IS_NULL(BIG_ARRAYS.get(key, i)) && ! s.add(BIG_ARRAYS.get(key, i))) throw new
AssertionError("Key " + BIG_ARRAYS.get(key, i) + " appears twice");

}
#else
private void checkTable() {}
#endif

#ifdef TEST

private static long seed = System.currentTimeMillis();
private static java.util.Random r = new java.util.Random(seed);

private static KEY_TYPE genKey() {
#ifdef KEY_CLASS_Byte || KEY_CLASS_Short || KEY_CLASS_Character
    return (KEY_TYPE)(r.nextInt());
#elif KEYS_PRIMITIVE
    return r.NEXT_KEY();
#elif KEY_CLASS_Object
    return Integer.toBinaryString(r.nextInt());
#else
    return new java.io.Serializable() {};
#endif
}

private static final class ArrayComparator implements java.util.Comparator {

```

```

public int compare(Object a, Object b) {
    byte[] aa = (byte[])a;
    byte[] bb = (byte[])b;
    int length = Math.min(aa.length, bb.length);
    for(int i = 0; i < length; i++) {
        if (aa[i] < bb[i]) return -1;
        if (aa[i] > bb[i]) return 1;
    }
    return aa.length == bb.length ? 0 : (aa.length < bb.length ? -1 : 1);
}
}

```

```

private static final class MockSet extends java.util.TreeSet {
    private java.util.List list = new java.util.ArrayList();

```

```

    public MockSet(java.util.Comparator c) { super(c); }

```

```

    public boolean add(Object k) {
        if (! contains(k)) list.add(k);
        return super.add(k);
    }

```

```

    public boolean addAll(Collection c) {
        java.util.Iterator i = c.iterator();
        boolean result = false;
        while(i.hasNext()) result |= add(i.next());
        return result;
    }

```

```

    public boolean removeAll(Collection c) {
        java.util.Iterator i = c.iterator();
        boolean result = false;
        while(i.hasNext()) result |= remove(i.next());
        return result;
    }

```

```

    public boolean remove(Object k) {
        if (contains(k)) {
            int i = list.size();
            while(i-- != 0) if (comparator().compare(list.get(i), k) == 0) {
                list.remove(i);
                break;
            }
        }
        return super.remove(k);
    }

```

```

    private void justRemove(Object k) { super.remove(k); }

```

```

public java.util.Iterator iterator() {
return new java.util.Iterator() {
    final java.util.Iterator iterator = list.iterator();
    Object curr;
    public Object next() { return curr = iterator.next(); }
    public boolean hasNext() { return iterator.hasNext(); }
    public void remove() {
        justRemove(curr);
        iterator.remove();
    }
};
}
}

private static java.text.NumberFormat format = new java.text.DecimalFormat("#,###.00");
private static java.text.FieldPosition fp = new java.text.FieldPosition(0);

private static String format(double d) {
    StringBuffer s = new StringBuffer();
    return format.format(d, s, fp).toString();
}

private static void speedTest(int n, float f, boolean comp) {
    int i, j;
    OPEN_HASH_BIG_SET m;
    java.util.HashSet t;

    KEY_TYPE k[] = new KEY_TYPE[n];
    KEY_TYPE nk[] = new KEY_TYPE[n];
    long ms;

    for(i = 0; i < n; i++) {
        k[i] = genKey();
        nk[i] = genKey();
    }

    double totAdd = 0, totYes = 0, totNo = 0, totIter = 0, totRemYes = 0, totRemNo = 0, d;

    if (comp) { for(j = 0; j < 20; j++) {

        t = new java.util.HashSet(16);

        /* We add pairs to t. */
        ms = System.currentTimeMillis();
        for(i = 0; i < n; i++) t.add(KEY2OBJ(k[i]));
        d = 1.0 * n / (System.currentTimeMillis() - ms);
        if (j > 2) totAdd += d;
    }
}

```

```

System.out.print("Add: " + format(d) + " K/s ");

/* We check for pairs in t. */
ms = System.currentTimeMillis();
for(i = 0; i < n; i++) t.contains(KEY2OBJ(k[i]));
d = 1.0 * n / (System.currentTimeMillis() - ms);
if (j > 2) totYes += d;
System.out.print("Yes: " + format(d) + " K/s ");

/* We check for pairs not in t. */
ms = System.currentTimeMillis();
for(i = 0; i < n; i++) t.contains(KEY2OBJ(nk[i]));
d = 1.0 * n / (System.currentTimeMillis() - ms);
if (j > 2) totNo += d;
System.out.print("No: " + format(d) + " K/s ");

/* We iterate on t. */
ms = System.currentTimeMillis();
for(java.util.Iterator it = t.iterator(); it.hasNext(); it.next());
d = 1.0 * n / (System.currentTimeMillis() - ms);
if (j > 2) totIter += d;
System.out.print("Iter: " + format(d) + " K/s ");

/* We delete pairs not in t. */
ms = System.currentTimeMillis();
for(i = 0; i < n; i++) t.remove(KEY2OBJ(nk[i]));
d = 1.0 * n / (System.currentTimeMillis() - ms);
if (j > 2) totRemNo += d;
System.out.print("RemNo: " + format(d) + " K/s ");

/* We delete pairs in t. */
ms = System.currentTimeMillis();
for(i = 0; i < n; i++) t.remove(KEY2OBJ(k[i]));
d = 1.0 * n / (System.currentTimeMillis() - ms);
if (j > 2) totRemYes += d;
System.out.print("RemYes: " + format(d) + " K/s ");

System.out.println();
}

System.out.println();
System.out.println("java.util Add: " + format(totAdd/(j-3)) + " K/s Yes: " + format(totYes/(j-3)) + " K/s No: " +
format(totNo/(j-3)) + " K/s Iter: " + format(totIter/(j-3)) + " K/s RemNo: " + format(totRemNo/(j-3)) + " K/s
RemYes: " + format(totRemYes/(j-3)) + " K/s");

System.out.println();

totAdd = totYes = totNo = totIter = totRemYes = totRemNo = 0;

```

```

}

for(j = 0; j < 20; j++) {

    m = new OPEN_HASH_BIG_SET(16, f);

    /* We add pairs to m. */
    ms = System.currentTimeMillis();
    for(i = 0; i < n; i++) m.add(k[i]);
    d = 1.0 * n / (System.currentTimeMillis() - ms);
    if (j > 2) totAdd += d;
    System.out.print("Add: " + format(d) + " K/s ");

    /* We check for pairs in m. */
    ms = System.currentTimeMillis();
    for(i = 0; i < n; i++) m.contains(k[i]);
    d = 1.0 * n / (System.currentTimeMillis() - ms);
    if (j > 2) totYes += d;
    System.out.print("Yes: " + format(d) + " K/s ");

    /* We check for pairs not in m. */
    ms = System.currentTimeMillis();
    for(i = 0; i < n; i++) m.contains(nk[i]);
    d = 1.0 * n / (System.currentTimeMillis() - ms);
    if (j > 2) totNo += d;
    System.out.print("No: " + format(d) + " K/s ");

    /* We iterate on m. */
    ms = System.currentTimeMillis();
    for(KEY_ITERATOR it = (KEY_ITERATOR)m.iterator(); it.hasNext(); it.NEXT_KEY());
    d = 1.0 * n / (System.currentTimeMillis() - ms);
    if (j > 2) totIter += d;
    System.out.print("Iter: " + format(d) + " K/s ");

    /* We delete pairs not in m. */
    ms = System.currentTimeMillis();
    for(i = 0; i < n; i++) m.remove(nk[i]);
    d = 1.0 * n / (System.currentTimeMillis() - ms);
    if (j > 2) totRemNo += d;
    System.out.print("RemNo: " + format(d) + " K/s ");

    /* We delete pairs in m. */
    ms = System.currentTimeMillis();
    for(i = 0; i < n; i++) m.remove(k[i]);
    d = 1.0 * n / (System.currentTimeMillis() - ms);
    if (j > 2) totRemYes += d;
    System.out.print("RemYes: " + format(d) + " K/s ");
}

```

```
System.out.println();
}
```

```
System.out.println();
System.out.println("fastutil Add: " + format(totAdd/(j-3)) + " K/s Yes: " + format(totYes/(j-3)) + " K/s No: " +
format(totNo/(j-3)) + " K/s Iter: " + format(totIter/(j-3)) + " K/s RemNo: " + format(totRemNo/(j-3)) + " K/s
RemYes: " + format(totRemYes/(j-3)) + " K/s");
```

```
System.out.println();
}
```

```
private static void fatal(String msg) {
    System.out.println(msg);
    System.exit(1);
}
```

```
private static void ensure(boolean cond, String msg) {
    if (cond) return;
    fatal(msg);
}
```

```
private static void printProbes(OPEN_HASH_BIG_SET m) {
    long totProbes = 0;
    double totSquareProbes = 0;
    int maxProbes = 0;
    final double f = (double)m.size / m.n;
    for(int i = 0, c = 0; i < m.n; i++) {
        if (! KEY_IS_NULL(BIG_ARRAYS.get(m.key, i))) c++;
        else {
            if (c != 0) {
                final long p = (c + 1) * (c + 2) / 2;
                totProbes += p;
                totSquareProbes += (double)p * p;
            }
            maxProbes = Math.max(c, maxProbes);
            c = 0;
            totProbes++;
            totSquareProbes++;
        }
    }
}
```

```
final double expected = (double)totProbes / m.n;
System.err.println("Expected probes: " + (
    3 * Math.sqrt(3) * (f / ((1 - f) * (1 - f))) + 4 / (9 * f) - 1
) + "; actual: " + expected + "; stddev: " + Math.sqrt(totSquareProbes / m.n - expected * expected) + "; max probes:
```

```

" + maxProbes);
}
private static void runTest(int n, float f) {
    int c;
    OPEN_HASH_BIG_SET m = new OPEN_HASH_BIG_SET(Hash.DEFAULT_INITIAL_SIZE, f);
    java.util.Set t = new java.util.HashSet();

    /* First of all, we fill t with random data. */

    for(int i=0; i<f * n; i++) t.add(KEY2OBJ(genKey()));

    /* Now we add to m the same data */

    m.addAll(t);

    if (!m.equals(t)) System.out.println("Error (" + seed + "): !m.equals(t) after insertion");
    if (!t.equals(m)) System.out.println("Error (" + seed + "): !t.equals(m) after insertion");
    printProbes(m);

    /* Now we check that m actually holds that data. */

    for(java.util.Iterator i=t.iterator(); i.hasNext();) {
        Object e = i.next();
        if (!m.contains(e)) {
            System.out.println("Error (" + seed + "): m and t differ on a key (" + e + ") after insertion (iterating on t)");
            System.exit(1);
        }
    }

    /* Now we check that m actually holds that data, but iterating on m. */

    c = 0;
    for(java.util.Iterator i=m.iterator(); i.hasNext();) {
        Object e = i.next();
        c++;
        if (!t.contains(e)) {
            System.out.println("Error (" + seed + "): m and t differ on a key (" + e + ") after insertion (iterating on m)");
            System.exit(1);
        }
    }

    if (c != t.size()) {
        System.out.println("Error (" + seed + "): m has only " + c + " keys instead of " + t.size() + " after insertion (iterating on m)");
        System.exit(1);
    }

    /* Now we check that inquiries about random data give the same answer in m and t. For
    m we use the polymorphic method. */

```

```

for(int i=0; i<n; i++) {
    KEY_TYPE T = genKey();
    if (m.contains(T) != t.contains(KEY2OBJ(T))) {
        System.out.println("Error (" + seed + "): divergence in keys between t and m (polymorphic method)");
        System.exit(1);
    }
}

/* Again, we check that inquiries about random data give the same answer in m and t, but
   for m we use the standard method. */

for(int i=0; i<n; i++) {
    KEY_TYPE T = genKey();
    if (m.contains(KEY2OBJ(T)) != t.contains(KEY2OBJ(T))) {
        System.out.println("Error (" + seed + "): divergence between t and m (standard method)");
        System.exit(1);
    }
}

/* Now we put and remove random data in m and t, checking that the result is the same. */

for(int i=0; i<20*n; i++) {
    KEY_TYPE T = genKey();
    if (m.add(KEY2OBJ(T)) != t.add(KEY2OBJ(T))) {
        System.out.println("Error (" + seed + "): divergence in add() between t and m");
        System.exit(1);
    }
    T = genKey();
    if (m.remove(KEY2OBJ(T)) != t.remove(KEY2OBJ(T))) {
        System.out.println("Error (" + seed + "): divergence in remove() between t and m");
        System.exit(1);
    }
}

if (!m.equals(t)) System.out.println("Error (" + seed + "): !m.equals(t) after removal");
if (!t.equals(m)) System.out.println("Error (" + seed + "): !t.equals(m) after removal");

/* Now we check that m actually holds that data. */

for(java.util.Iterator i=t.iterator(); i.hasNext();) {
    Object e = i.next();
    if (!m.contains(e)) {
        System.out.println("Error (" + seed + "): m and t differ on a key (" + e + ") after removal (iterating on t)");
        System.exit(1);
    }
}

```

```

/* Now we check that m actually holds that data, but iterating on m. */

for(java.util.Iterator i=m.iterator(); i.hasNext();) {
    Object e = i.next();
    if (!t.contains(e)) {
        System.out.println("Error (" + seed + "): m and t differ on a key (" + e + ") after removal (iterating on m)");
        System.exit(1);
    }
}

printProbes(m);

/* Now we make m into an array, make it again a set and check it is OK. */
KEY_TYPE a[] = m.TO_KEY_ARRAY();

if (!new OPEN_HASH_BIG_SET(a).equals(m))
    System.out.println("Error (" + seed + "): toArray() output (or array-based constructor) is not OK");

/* Now we check cloning. */

ensure(m.equals(((OPEN_HASH_BIG_SET)m).clone()), "Error (" + seed + "): m does not equal m.clone()");
ensure(((OPEN_HASH_BIG_SET)m).clone().equals(m), "Error (" + seed + "): m.clone() does not equal m");

int h = m.hashCode();

/* Now we save and read m. */

try {
    java.io.File ff = new java.io.File("it.unimi.dsi.fastutil.test");
    java.io.OutputStream os = new java.io.FileOutputStream(ff);
    java.io.ObjectOutputStream oos = new java.io.ObjectOutputStream(os);

    oos.writeObject(m);
    oos.close();

    java.io.InputStream is = new java.io.FileInputStream(ff);
    java.io.ObjectInputStream ois = new java.io.ObjectInputStream(is);

    m = (OPEN_HASH_BIG_SET)ois.readObject();
    ois.close();
    ff.delete();
}
catch(Exception e) {
    e.printStackTrace();
    System.exit(1);
}

```

```

#if !KEY_CLASS_Reference
if (m.hashCode() != h) System.out.println("Error (" + seed + "): hashCode() changed after save/read");

printProbes(m);

/* Now we check that m actually holds that data, but iterating on m. */

for(java.util.Iterator i=m.iterator(); i.hasNext();) {
    Object e = i.next();
    if (!t.contains(e)) {
        System.out.println("Error (" + seed + "): m and t differ on a key (" + e + ") after save/read");
        System.exit(1);
    }
}
#else
m.clear();
m.addAll(t);
#endif

/* Now we put and remove random data in m and t, checking that the result is the same. */

for(int i=0; i<20*n; i++) {
    KEY_TYPE T = genKey();
    if (m.add(KEY2OBJ(T)) != t.add(KEY2OBJ(T))) {
        System.out.println("Error (" + seed + "): divergence in add() between t and m after save/read");
        System.exit(1);
    }
    T = genKey();
    if (m.remove(KEY2OBJ(T)) != t.remove(KEY2OBJ(T))) {
        System.out.println("Error (" + seed + "): divergence in remove() between t and m after save/read");
        System.exit(1);
    }
}

if (!m.equals(t)) System.out.println("Error (" + seed + "): !m.equals(t) after post-save/read removal");
if (!t.equals(m)) System.out.println("Error (" + seed + "): !t.equals(m) after post-save/read removal");

/* Now we take out of m everything, and check that it is empty. */

for(java.util.Iterator i=m.iterator(); i.hasNext();) { i.next(); i.remove();}

if (!m.isEmpty()) {
    System.out.println("Error (" + seed + "): m is not empty (as it should be)");
    System.exit(1);
}

#if KEY_CLASS_Integer || KEY_CLASS_Long

```

```

m = new OPEN_HASH_BIG_SET(n, f);
t.clear();
int x;

/* Now we torture-test the hash table. This part is implemented only for integers and longs. */

int p = m.key.length - 1;

for(int i=0; i<p; i++) {
    for (int j=0; j<20; j++) {
        m.add(i+(r.nextInt() % 10)*p);
        m.remove(i+(r.nextInt() % 10)*p);
    }

    for (int j=-10; j<10; j++) m.remove(i+j*p);
}

t.addAll(m);

/* Now all table entries are REMOVED. */

int k = 0;
for(int i=0; i<(p*f)/10; i++) {
    for (int j=0; j<10; j++) {
        k++;
        x = i+(r.nextInt() % 10)*p;
        if (m.add(x) != t.add(KEY2OBJ(x)))
            System.out.println("Error (" + seed + "): m and t differ on a key during torture-test insertion.");
    }
}

if (!m.equals(t)) System.out.println("Error (" + seed + "): !m.equals(t) after torture-test insertion");
if (!t.equals(m)) System.out.println("Error (" + seed + "): !t.equals(m) after torture-test insertion");

for(int i=0; i<(p*f)/10; i++) {
    for (int j=0; j<10; j++) {
        x = i+(r.nextInt() % 10)*p;
        if (m.remove(x) != t.remove(KEY2OBJ(x)))
            System.out.println("Error (" + seed + "): m and t differ on a key during torture-test removal.");
    }
}

if (!m.equals(t)) System.out.println("Error (" + seed + "): !m.equals(t) after torture-test removal");
if (!t.equals(m)) System.out.println("Error (" + seed + "): !t.equals(m) after torture-test removal");

if (!m.equals(m.clone())) System.out.println("Error (" + seed + "): !m.equals(m.clone()) after torture-test removal");
if (!(new OPEN_HASH_BIG_SET(m.clone()).equals(m)) System.out.println("Error (" + seed + "): !m.clone().equals(m)
after torture-test removal");

```

```

m.trim();

if (!m.equals(t)) System.out.println("Error (" + seed + "): !m.equals(t) after trim()");
if (!t.equals(m)) System.out.println("Error (" + seed + "): !t.equals(m) after trim()");

#endif

System.out.println("Test OK");
return;
}

public static void main(String args[]) {
float f = Hash.DEFAULT_LOAD_FACTOR;
int n = Integer.parseInt(args[1]);
if (args.length>2) f = Float.parseFloat(args[2]);
if (args.length > 3) r = new java.util.Random(seed = Long.parseLong(args[3]));

try {
if ("speedTest".equals(args[0]) || "speedComp".equals(args[0])) speedTest(n, f, "speedComp".equals(args[0]));
else if ("test".equals(args[0])) runTest(n, f);
} catch(Throwable e) {
e.printStackTrace(System.err);
System.err.println("seed: " + seed);
}
}

#endif

}

```

Found in path(s):

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-a775574/drv/OpenHashBigSet.drv

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2002-2017 Sebastiano Vigna

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

```
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
* See the License for the specific language governing permissions and  
* limitations under the License.  
*/
```

```
package PACKAGE;
```

```
import java.util.Iterator;  
import java.util.NoSuchElementException;
```

```
/** A class providing static methods and objects that do useful things with type-specific iterators.  
*  
* @see Iterator  
*/
```

```
public final class BIG_LIST_ITERATORS {
```

```
    private BIG_LIST_ITERATORS() {}
```

```
    /** A class returning no elements and a type-specific big list iterator interface.  
    *  
    * <p>This class may be useful to implement your own in case you subclass  
    * a type-specific iterator.  
    */
```

```
    public static class EmptyBigListIterator KEY_GENERIC implements KEY_BIG_LIST_ITERATOR  
        KEY_GENERIC, java.io.Serializable, Cloneable {
```

```
        private static final long serialVersionUID = -7046029254386353129L;
```

```
        protected EmptyBigListIterator() {}
```

```
        @Override  
        public boolean hasNext() { return false; }
```

```
        @Override  
        public boolean hasPrevious() { return false; }
```

```
        @Override  
        public KEY_GENERIC_TYPE NEXT_KEY() { throw new NoSuchElementException(); }
```

```
        @Override  
        public KEY_GENERIC_TYPE PREV_KEY() { throw new NoSuchElementException(); }
```

```
        @Override  
        public long nextIndex() { return 0; }
```

```

@Override
public long previousIndex() { return -1; }

@Override
public long skip(long n) { return 0; };

@Override
public long back(long n) { return 0; };

@Override
public Object clone() { return EMPTY_BIG_LIST_ITERATOR; }

private Object readResolve() { return EMPTY_BIG_LIST_ITERATOR; }
}

/** An empty iterator (immutable). It is serializable and cloneable.
 *
 * <p>The class of this objects represent an abstract empty iterator
 * that can iterate as a type-specific (list) iterator.
 */

SUPPRESS_WARNINGS_KEY_RAWTYPES
public static final EmptyBigListIterator EMPTY_BIG_LIST_ITERATOR = new EmptyBigListIterator();

/** An iterator returning a single element. */

private static class SingletonBigListIterator KEY_GENERIC implements KEY_BIG_LIST_ITERATOR
KEY_GENERIC {
    private final KEY_GENERIC_TYPE element;
    private int curr;

    public SingletonBigListIterator(final KEY_GENERIC_TYPE element) {
        this.element = element;
    }

    @Override
    public boolean hasNext() { return curr == 0; }

    @Override
    public boolean hasPrevious() { return curr == 1; }

    @Override
    public KEY_GENERIC_TYPE NEXT_KEY() {
        if (!hasNext()) throw new NoSuchElementException();
        curr = 1;
        return element;
    }
}

```

```

@Override
public KEY_GENERIC_TYPE PREV_KEY() {
    if (!hasPrevious()) throw new NoSuchElementException();
    curr = 0;
    return element;
}

@Override
public long nextIndex() {
    return curr;
}

@Override
public long previousIndex() {
    return curr - 1;
}
}

/** Returns an iterator that iterates just over the given element.
 *
 * @param element the only element to be returned by a type-specific list iterator.
 * @return an iterator that iterates just over {@code element}.
 */
public static KEY_GENERIC KEY_BIG_LIST_ITERATOR KEY_GENERIC singleton(final
KEY_GENERIC_TYPE element) {
    return new SingletonBigListIterator KEY_GENERIC_DIAMOND(element);
}

/** An unmodifiable wrapper class for big list iterators. */

public static class UnmodifiableBigListIterator KEY_GENERIC implements KEY_BIG_LIST_ITERATOR
KEY_GENERIC {
    protected final KEY_BIG_LIST_ITERATOR KEY_GENERIC i;

    public UnmodifiableBigListIterator(final KEY_BIG_LIST_ITERATOR KEY_GENERIC i) {
        this.i = i;
    }

    @Override
    public boolean hasNext() { return i.hasNext(); }

    @Override
    public boolean hasPrevious() { return i.hasPrevious(); }

    @Override

```

```

public KEY_GENERIC_TYPE NEXT_KEY() { return i.NEXT_KEY(); }

@Override
public KEY_GENERIC_TYPE PREV_KEY() { return i.PREV_KEY(); }

@Override
public long nextIndex() { return i.nextIndex(); }

@Override
public long previousIndex() { return i.previousIndex(); }
}

/** Returns an unmodifiable list iterator backed by the specified list iterator.
 *
 * @param i the list iterator to be wrapped in an unmodifiable list iterator.
 * @return an unmodifiable view of the specified list iterator.
 */
public static KEY_GENERIC KEY_BIG_LIST_ITERATOR KEY_GENERIC unmodifiable(final
KEY_BIG_LIST_ITERATOR KEY_GENERIC i) { return new UnmodifiableBigListIterator
KEY_GENERIC_DIAMOND(i); }

/** A class exposing a list iterator as a big-list iterator.. */

public static class BigListIteratorListIterator KEY_GENERIC implements KEY_BIG_LIST_ITERATOR
KEY_GENERIC {
protected final KEY_LIST_ITERATOR KEY_GENERIC i;

protected BigListIteratorListIterator(final KEY_LIST_ITERATOR KEY_GENERIC i) {
this.i = i;
}

private int intDisplacement(long n) {
if (n < Integer.MIN_VALUE || n > Integer.MAX_VALUE) throw new IndexOutOfBoundsException("This big
iterator is restricted to 32-bit displacements");
return (int)n;
}

@Override
public void set(KEY_GENERIC_TYPE ok) { i.set(ok); }

@Override
public void add(KEY_GENERIC_TYPE ok) { i.add(ok); }

@Override
public int back(int n) { return i.back(n); }

@Override

```

```

public long back(long n) { return i.back(intDisplacement(n)); }

@Override
public void remove() { i.remove(); }

@Override
public int skip(int n) { return i.skip(n); }

@Override
public long skip(long n) { return i.skip(intDisplacement(n)); }

@Override
public boolean hasNext() { return i.hasNext(); }

@Override
public boolean hasPrevious() { return i.hasPrevious(); }

@Override
public KEY_GENERIC_TYPE NEXT_KEY() { return i.NEXT_KEY(); }

@Override
public KEY_GENERIC_TYPE PREV_KEY() { return i.PREV_KEY(); }

@Override
public long nextIndex() { return i.nextIndex(); }

@Override
public long previousIndex() { return i.previousIndex(); }
}

/** Returns a big-list iterator backed by the specified list iterator.
 *
 * @param i the list iterator to adapted to the big-list-iterator interface.
 * @return a big-list iterator backed by the specified list iterator.
 */
public static KEY_GENERIC KEY_BIG_LIST_ITERATOR KEY_GENERIC asBigListIterator(final
KEY_LIST_ITERATOR KEY_GENERIC i) { return new BigListIteratorListIterator
KEY_GENERIC_DIAMOND(i); }
}

```

Found in path(s):

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-
a775574/drv/BigListIterators.drv

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2002-2017 Sebastiano Vigna

*

* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

package PACKAGE;

/** An abstract class providing basic methods for priority queues implementing a type-specific interface.
* @deprecated As of fastutil 8 this class is no longer necessary, as its previous abstract
* methods are now default methods of the type-specific interface.
*/

@Deprecated

```
public abstract class ABSTRACT_PRIORITY_QUEUE KEY_GENERIC extends  
it.unimi.dsi.fastutil.AbstractPriorityQueue<KEY_GENERIC_CLASS> implements java.io.Serializable,  
PRIORITY_QUEUE KEY_GENERIC {  
    private static final long serialVersionUID = 1L;  
}
```

Found in path(s):

* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-
a775574/drv/AbstractPriorityQueue.drv

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2003-2017 Paolo Boldi and Sebastiano Vigna

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

```

package PACKAGE;

#if KEY_CLASS_Object
import java.util.Comparator;
#endif

import java.util.Arrays;

/** A class providing static methods and objects that do useful things with indirect heaps.
 *
 * <p>An indirect heap is an extension of a semi-indirect heap using also an
 * <em>inversion array</em> of the same length as the reference array,
 * satisfying the relation { @code heap[inv[i]]==i } when
 * { @code inv[i]&gt;=0}, and { @code inv[heap[i]]==i } for all elements in the heap.
 */

public final class INDIRECT_HEAPS {

    private INDIRECT_HEAPS() {}

    /** Moves the given element down into the indirect heap until it reaches the lowest possible position.
     *
     * @param refArray the reference array.
     * @param heap the indirect heap (starting at 0).
     * @param inv the inversion array.
     * @param size the number of elements in the heap.
     * @param i the index in the heap of the element to be moved down.
     * @param c a type-specific comparator, or { @code null } for the natural order.
     * @return the new position in the heap of the element of heap index { @code i }.
     */

    SUPPRESS_WARNINGS_KEY_UNCHECKED
    public static KEY_GENERIC int downHeap(final KEY_GENERIC_TYPE[] refArray, final int[] heap, final int[]
    inv, final int size, int i, final KEY_COMPARATOR KEY_GENERIC c) {
        assert i < size;

        final int e = heap[i];
        final KEY_GENERIC_TYPE E = refArray[e];
        int child;

        if (c == null)
            while ((child = (i << 1) + 1) < size) {
                int t = heap[child];
                final int right = child + 1;
                if (right < size && KEY_LESS(refArray[heap[right]], refArray[t])) t = heap[child = right];
                if (KEY_LESSEQ(E, refArray[t])) break;
                heap[i] = t;

```

```

    inv[heap[i]] = i;
    i = child;
}
else
while ((child = (i << 1) + 1) < size) {
    int t = heap[child];
    final int right = child + 1;
    if (right < size && c.compare(refArray[heap[right]], refArray[t]) < 0) t = heap[child = right];
    if (c.compare(E, refArray[t]) <= 0) break;
    heap[i] = t;
    inv[heap[i]] = i;
    i = child;
}

heap[i] = e;
inv[e] = i;
return i;
}

/** Moves the given element up in the indirect heap until it reaches the highest possible position.
 *
 * Note that in principle after this call the heap property may be violated.
 *
 * @param refArray the reference array.
 * @param heap the indirect heap (starting at 0).
 * @param inv the inversion array.
 * @param size the number of elements in the heap.
 * @param i the index in the heap of the element to be moved up.
 * @param c a type-specific comparator, or { @code null } for the natural order.
 * @return the new position in the heap of the element of heap index { @code i }.
 */

SUPPRESS_WARNINGS_KEY_UNCHECKED
public static KEY_GENERIC int upHeap(final KEY_GENERIC_TYPE[] refArray, final int[] heap, final int[] inv,
final int size, int i, final KEY_COMPARATOR KEY_GENERIC c) {
    assert i < size;

    final int e = heap[i];
    final KEY_GENERIC_TYPE E = refArray[e];

    if (c == null)
        while (i != 0) {
            final int parent = (i - 1) >>> 1;
            final int t = heap[parent];
            if (KEY_LESSEQ(refArray[t], E)) break;
            heap[i] = t;
            inv[heap[i]] = i;
            i = parent;

```

```

    }
    else
    while (i != 0) {
        final int parent = (i - 1) >>> 1;
        final int t = heap[parent];
        if (c.compare(refArray[t], E) <= 0) break;
        heap[i] = t;
        inv[heap[i]] = i;
        i = parent;
    }

    heap[i] = e;
    inv[e] = i;

    return i;
}

/** Creates an indirect heap in the given array.
 *
 * @param refArray the reference array.
 * @param offset the first element of the reference array to be put in the heap.
 * @param length the number of elements to be put in the heap.
 * @param heap the array where the heap is to be created.
 * @param inv the inversion array.
 * @param c a type-specific comparator, or { @code null } for the natural order.
 */

public static KEY_GENERIC void makeHeap(final KEY_GENERIC_TYPE[] refArray, final int offset, final int
length, final int[] heap, final int[] inv, final KEY_COMPARATOR KEY_GENERIC c) {
    ARRAYS.ensureOffsetLength(refArray, offset, length);
    if (heap.length < length) throw new IllegalArgumentException("The heap length (" + heap.length + ") is smaller
than the number of elements (" + length + ")");
    if (inv.length < refArray.length) throw new IllegalArgumentException("The inversion array length (" + heap.length
+ ") is smaller than the length of the reference array (" + refArray.length + ")");

    Arrays.fill(inv, 0, refArray.length, -1);

    int i = length;
    while(i-- != 0) inv[heap[i] = offset + i] = i;

    i = length >>> 1;
    while(i-- != 0) downHeap(refArray, heap, inv, length, i, c);
}

/** Creates an indirect heap from a given index array.
 *
 * @param refArray the reference array.

```

```
* @param heap an array containing indices into {@code refArray}.
* @param inv the inversion array.
* @param size the number of elements in the heap.
* @param c a type-specific comparator, or {@code null} for the natural order.
*/
```

```
public static KEY_GENERIC void makeHeap(final KEY_GENERIC_TYPE[] refArray, final int[] heap, final int[]
inv, final int size, final KEY_COMPARATOR KEY_GENERIC c) {
    int i = size >>> 1;
    while(i-- != 0) downHeap(refArray, heap, inv, size, i, c);
}
}
```

Found in path(s):

```
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-
a775574/drv/IndirectHeaps.drv
```

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2002-2017 Sebastiano Vigna
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

```
package PACKAGE;
```

```
import java.util.SortedSet;
import java.util.Collection;
```

```
/** A type-specific {@link SortedSet}; provides some additional methods that use polymorphism to avoid
(un)boxing.
```

```
*
* <p>Additionally, this interface strengthens {@link #iterator()},
* {@link #comparator()} (for primitive types), {@link SortedSet#subSet(Object, Object)},
* {@link SortedSet#headSet(Object)} and {@link SortedSet#tailSet(Object)}.
*
* @see SortedSet
```

*/

public interface SORTED_SET KEY_GENERIC extends SET KEY_GENERIC,
SortedSet<KEY_GENERIC_CLASS>, KEY_BIDI_ITERABLE KEY_GENERIC {

/** Returns a type-specific {@link it.unimi.dsi.fastutil.BidirectionalIterator} on the elements in
* this set, starting from a given element of the domain (optional operation).
*
* <p>This method returns a type-specific bidirectional iterator with given
* starting point. The starting point is any element comparable to the
* elements of this set (even if it does not actually belong to the
* set). The next element of the returned iterator is the least element of
* the set that is greater than the starting point (if there are no
* elements greater than the starting point, {@link
* it.unimi.dsi.fastutil.BidirectionalIterator#hasNext() hasNext()} will return
* {@code false}). The previous element of the returned iterator is
* the greatest element of the set that is smaller than or equal to the
* starting point (if there are no elements smaller than or equal to the
* starting point, {@link it.unimi.dsi.fastutil.BidirectionalIterator#hasPrevious()
* hasPrevious()} will return {@code false}).
*
* <p>Note that passing the last element of the set as starting point and
* calling {@link it.unimi.dsi.fastutil.BidirectionalIterator#previous() previous()} you can traverse the
* entire set in reverse order.
*
* @param fromElement an element to start from.
* @return a bidirectional iterator on the element in this set, starting at the given element.
* @throws UnsupportedOperationException if this set does not support iterators with a starting point.
*/

KEY_BIDI_ITERATOR KEY_GENERIC iterator(KEY_GENERIC_TYPE fromElement);

/** Returns a type-specific {@link it.unimi.dsi.fastutil.BidirectionalIterator} on the elements in
* this set.
*
* <p>Note that this specification strengthens the one given in the corresponding type-specific
* {@link Collection}.
*
* <p>This method returns a parameterised bidirectional iterator. The iterator
* can be moreover safely cast to a type-specific iterator.
*
* @return a bidirectional iterator on the element in this set.
*/

@Override

KEY_BIDI_ITERATOR KEY_GENERIC iterator();

```

/** Returns a view of the portion of this sorted set whose elements range from {@code fromElement}, inclusive, to
{@code toElement}, exclusive.
 * <p>Note that this specification strengthens the one given in {@link SortedSet#subSet(Object,Object)}.
 * @see SortedSet#subSet(Object,Object)
 */

#if KEYS_REFERENCE
@Override
#endif
SORTED_SET KEY_GENERIC subSet(KEY_GENERIC_TYPE fromElement, KEY_GENERIC_TYPE
toElement) ;

/** Returns a view of the portion of this sorted set whose elements are strictly less than {@code toElement}.
 * <p>Note that this specification strengthens the one given in {@link SortedSet#headSet(Object)}.
 * @see SortedSet#headSet(Object)
 */

#if KEYS_REFERENCE
@Override
#endif
SORTED_SET KEY_GENERIC headSet(KEY_GENERIC_TYPE toElement);

/** Returns a view of the portion of this sorted set whose elements are greater than or equal to {@code
fromElement}.
 * <p>Note that this specification strengthens the one given in {@link SortedSet#headSet(Object)}.
 * @see SortedSet#tailSet(Object)
 */

#if KEYS_REFERENCE
@Override
#endif
SORTED_SET KEY_GENERIC tailSet(KEY_GENERIC_TYPE fromElement);

#if KEYS_PRIMITIVE
/** {@inheritDoc}
 * <p>Note that this specification strengthens the one given in {@link SortedSet#comparator()}.
 */
@Override
KEY_COMPARATOR comparator();

/** Returns the first (lowest) element currently in this set.
 * @see SortedSet#first()
 */
KEY_TYPE FIRST();

/** Returns the last (highest) element currently in this set.
 * @see SortedSet#last()
 */

```

```

KEY_TYPE LAST();

/** { @inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead.
 */
@Deprecated
@Override
default SORTED_SET KEY_GENERIC subSet(final KEY_GENERIC_CLASS from, final
KEY_GENERIC_CLASS to) {
    return subSet(from.KEY_VALUE(), to.KEY_VALUE());
}

/** { @inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead.
 */
@Deprecated
@Override
default SORTED_SET KEY_GENERIC headSet(final KEY_GENERIC_CLASS to) {
    return headSet(to.KEY_VALUE());
}

/** { @inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead.
 */
@Deprecated
@Override
default SORTED_SET KEY_GENERIC tailSet(final KEY_GENERIC_CLASS from) {
    return tailSet(from.KEY_VALUE());
}

/** { @inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead.
 */
@Deprecated
@Override
default KEY_GENERIC_CLASS first() {
    return KEY2OBJ(FIRST());
}

/** { @inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead.
 */
@Deprecated
@Override
default KEY_CLASS last() {
    return KEY2OBJ(LAST());
}

```

```
#endif
```

```
}
```

Found in path(s):

```
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-  
a775574/drv/SortedSet.drv
```

No license file was found, but licenses were detected in source scan.

```
/*
```

```
* Copyright (C) 2002-2017 Sebastiano Vigna
```

```
*
```

```
* Licensed under the Apache License, Version 2.0 (the "License");
```

```
* you may not use this file except in compliance with the License.
```

```
* You may obtain a copy of the License at
```

```
*
```

```
* http://www.apache.org/licenses/LICENSE-2.0
```

```
*
```

```
* Unless required by applicable law or agreed to in writing, software
```

```
* distributed under the License is distributed on an "AS IS" BASIS,
```

```
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
```

```
* See the License for the specific language governing permissions and
```

```
* limitations under the License.
```

```
*/
```

```
package PACKAGE;
```

```
import it.unimi.dsi.fastutil.Function;
```

```
/** A type-specific {@link Function}; provides some additional methods that use polymorphism to avoid  
(un)boxing.
```

```
*
```

```
* <p>Type-specific versions of {@code get()}, {@code put()} and
```

```
* {@code remove()} cannot rely on {@code null} to denote absence of
```

```
* a key. Rather, they return a {@linkplain #defaultReturnValue() default
```

```
* return value}, which is set to 0/false at creation, but can be changed using
```

```
* the {@code defaultReturnValue()} method.
```

```
*
```

```
* <p>For uniformity reasons, even functions returning objects implement the default
```

```
* return value (of course, in this case the default return value is
```

```
* initialized to {@code null}).
```

```
*
```

```
* <p>The default implementation of optional operations just throw an {@link
```

```
* UnsupportedOperationException}, except for the type-specific {@code
```

```
* containsKey()}, which return true. Generic versions of accessors delegate to
```

```
* the corresponding type-specific counterparts following the interface rules.
```

```
*
```

```

* <p><strong>Warning:</strong> to fall in line as much as possible with the
* { @linkplain java.util.Map standard map interface}, it is required that
* standard versions of { @code get()}, { @code put()} and
* { @code remove()} for maps with primitive-type keys or values <em>return
* { @code null} to denote missing keys </em> rather than wrap the default
* return value in an object. In case both keys and values are reference
* types, the default return value must be returned instead, thus violating
* the { @linkplain java.util.Map standard map interface} when the default
* return value is not { @code null}.
*
* @see Function
*/

```

```
@FunctionalInterface
```

```
#ifdef JDK_PRIMITIVE_FUNCTION
```

```
public interface FUNCTION KEY_VALUE_GENERIC extends Function<KEY_GENERIC_CLASS,
VALUE_GENERIC_CLASS>, JDK_PRIMITIVE_FUNCTION KEY_GENERIC VALUE_GENERIC {
#else
```

```
public interface FUNCTION KEY_VALUE_GENERIC extends Function<KEY_GENERIC_CLASS,
VALUE_GENERIC_CLASS> {
#endif

```

```
#ifdef JDK_PRIMITIVE_FUNCTION
```

```
#if KEY_WIDENED
```

```
/** { @inheritDoc}
```

```
*
```

```

* <p>In this default implementation, the key gets narrowed down to the
* actual key type, throwing an exception if the given key can't be
* represented in the restricted domain. This is done for interoperability
* with the Java 8 function environment. Its use is discouraged, as
* unexpected errors can occur. Instead, the corresponding classes should be
* used (e.g., { @link it.unimi.dsi.fastutil.ints.Int2IntFunction} instead of
* { @link it.unimi.dsi.fastutil.shorts.Short2IntFunction}).

```

```
*
```

```
* @throws IllegalArgumentException If the given operand is not an element of the key domain.
```

```
* @since 8.0.0
```

```
* @deprecated Please use primitive types which don't have to be widened as keys.
```

```
*/
```

```
@Deprecated
```

```
#else
```

```
/** { @inheritDoc}
```

```
* @since 8.0.0
```

```
*/
```

```
#endif
```

```
@Override
```

```
default VALUE_GENERIC_TYPE_WIDENED
```

```
JDK_PRIMITIVE_FUNCTION_APPLY(KEY_GENERIC_TYPE_WIDENED operand) { return
GET_VALUE(KEY_NARROWING(operand)); }
```

```

#endif

/** Adds a pair to the map (optional operation).
 *
 * @param key the key.
 * @param value the value.
 * @return the old value, or the { @linkplain #defaultReturnValue() default return value } if no value was present for
the given key.
 * @see Function#put(Object,Object)
 */

default VALUE_GENERIC_TYPE put(final KEY_GENERIC_TYPE key, final VALUE_GENERIC_TYPE value)
{
    throw new UnsupportedOperationException();
}

/** Returns the value to which the given key is mapped.
 *
 * @param key the key.
 * @return the corresponding value, or the { @linkplain #defaultReturnValue() default return value } if no value was
present for the given key.
 * @see Function#get(Object)
 */

VALUE_GENERIC_TYPE GET_VALUE(KEY_TYPE key);

/** Removes the mapping with the given key (optional operation).
 * @param key the key.
 * @return the old value, or the { @linkplain #defaultReturnValue() default return value } if no value was present for
the given key.
 * @see Function#remove(Object)
 */

default VALUE_GENERIC_TYPE REMOVE_VALUE(final KEY_TYPE key) {
    throw new UnsupportedOperationException();
}

#if KEYS_PRIMITIVE || VALUES_PRIMITIVE

/** { @inheritDoc }
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default VALUE_GENERIC_CLASS put(final KEY_GENERIC_CLASS key, final VALUE_GENERIC_CLASS
value) {
    final KEY_GENERIC_TYPE k = KEY_CLASS2TYPE(key);
    final boolean containsKey = containsKey(k);
    final VALUE_GENERIC_TYPE v = put(k, VALUE_CLASS2TYPE(value));

```

```

return containsKey ? VALUE2OBJ(v) : null;
}

/** { @inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default VALUE_GENERIC_CLASS get(final Object key) {
#if KEYS_PRIMITIVE
    if (key == null) return null;
#endif
    final KEY_TYPE k = KEY_OBJ2TYPE(key);
    final VALUE_GENERIC_TYPE v = GET_VALUE(k);
    return (v != defaultReturnValue() || containsKey(k)) ? VALUE2OBJ(v) : null;
}

/** { @inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */
@Deprecated
@Override
default VALUE_GENERIC_CLASS remove(final Object key) {
#if KEYS_PRIMITIVE
    if (key == null) return null;
#endif
    final KEY_TYPE k = KEY_OBJ2TYPE(key);
    return containsKey(k) ? VALUE2OBJ(REMOVE_VALUE(k)) : null;
}

#if KEYS_PRIMITIVE

/** Returns true if this function contains a mapping for the specified key.
 *
 * <p>Note that for some kind of functions (e.g., hashes) this method will
 * always return true. In particular, this default implementation always
 * returns true.
 *
 * @param key the key.
 * @return true if this function associates a value to { @code key}.
 * @see Function#containsKey(Object)
 */
default boolean containsKey(KEY_TYPE key) {
    return true;
}

/** { @inheritDoc}
 * @deprecated Please use the corresponding type-specific method instead. */

@Deprecated

```

```
@Override
default boolean containsKey(final Object key) {
    return key == null ? false : containsKey(KEY_OBJ2TYPE(key));
}
```

```
#endif
```

```
#endif
```

```
/** Sets the default return value (optional operation).
 *
 * This value must be returned by type-specific versions of
 * { @code get()}, { @code put()} and { @code remove()} to
 * denote that the map does not contain the specified key. It must be
 * 0/{ @code false}/{ @code null} by default.
 *
 * @param rv the new default return value.
 * @see #defaultReturnValue()
 */
```

```
default void defaultReturnValue(VALUE_GENERIC_TYPE rv) {
    throw new UnsupportedOperationException();
}
```

```
/** Gets the default return value.
 *
 * <p>This default implementation just return the default null value
 * of the type ({ @code null} for objects, 0 for scalars, false for Booleans).
 *
 * @return the current default return value.
 */
```

```
default VALUE_GENERIC_TYPE defaultReturnValue() {
    return VALUE_NULL;
}
}
```

Found in path(s):

```
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-
a775574/drv/Function.drv
```

No license file was found, but licenses were detected in source scan.

```
/*
```

```
* Copyright (C) 2002-2017 Sebastiano Vigna
```

```
*
```

```
* Licensed under the Apache License, Version 2.0 (the "License");
```

```
* you may not use this file except in compliance with the License.
```

```
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

```
package PACKAGE;
```

```
/** An abstract class facilitating the creation of type-specific { @linkplain java.util.Comparator comparators }.
*
* @see java.util.Comparator
* @deprecated As of fastutil 8 this class is no longer necessary, as its only previous abstract
* method is now a default method of the type-specific interface.
*/
```

```
@Deprecated
```

```
public abstract class KEY_ABSTRACT_COMPARATOR KEY_GENERIC implements KEY_COMPARATOR
KEY_GENERIC, java.io.Serializable {
    private static final long serialVersionUID = 0L;
    protected KEY_ABSTRACT_COMPARATOR() {}
}
```

```
Found in path(s):
```

```
* /opt/cola/permits/1473460152_1668478175.1613455/0/vigna-fastutil-8-2-3-0-ga775574-1-tar-gz/vigna-fastutil-
a775574/drv/AbstractComparator.drv
```

1.226 org.apache.groovy:groovy-json 4.0.1

1.226.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2019 the original author or authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
```

- * distributed under the License is distributed on an "AS IS" BASIS,
- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.
- */

Found in path(s):

- * /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/filter/FilterContextImpl.groovy
- * /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/RestAssuredResponseOptionsImpl.java
- * /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/matcher/xml/LoadFromClasspathSupport.groovy
- * /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/AuthenticationSpecificationImpl.groovy
- * /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-jar/io/restassured/specification/AuthenticationSpecification.java
- * /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/LogRequestAndResponseOnFailListener.groovy
- * /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/matcher/xml/XmlXsdMatcher.groovy
- * /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/http/ContentEncoding.java
- * /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/http/ContentEncodingRegistry.java
- * /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-jar/io/restassured/authentication/NTLMAuthScheme.groovy
- * /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/http/HttpResponseDecorator.java
- * /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-jar/io/restassured/specification/ResponseSpecification.java
- * /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/MultiValueEntity.java
- * /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/PreemptiveAuthSpecImpl.groovy
- * /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-jar/io/restassured/specification/QueryableRequestSpecification.java
- * /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-jar/io/restassured/config/HeaderConfig.java
- * /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-jar/io/restassured/authentication/ExplicitNoAuthScheme.groovy
- * /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-jar/io/restassured/mapper/ObjectMapperSerializationContext.java
- * /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-jar/io/restassured/specification/LogSpecification.java
- * /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-jar/io/restassured/internal/support/FileReader.groovy
- * /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-

jar/io/restassured/response/ExtractableResponse.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/http/BoundaryExtractor.groovy
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/config/LogConfig.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/matcher/ResponseAwareMatcherComposer.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/assertion/StreamVerifier.groovy
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/builder/ResponseSpecBuilder.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/mapping/Jackson2Mapper.groovy
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/mapper/ObjectMapperType.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/http/ResponseParseException.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/support/PathSupport.groovy
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/ResponseSpecificationImpl.groovy
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/response/ResponseOptions.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/builder/MultiPartSpecBuilder.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/mapping/JohnzonMapper.groovy
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/RequestLogSpecificationImpl.groovy
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/ContentParser.groovy
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/matcher/RestAssuredMatchers.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/specification/RequestSender.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/ValidatableResponseOptionsImpl.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/specification/RequestSpecification.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/specification/RequestSenderOptions.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/support/Prettifier.groovy
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/authentication/BasicAuthScheme.groovy
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/RequestSpecificationImpl.groovy
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-

jar/io/restassured/internal/LogSpecificationImpl.groovy
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/specification/PreemptiveAuthSpec.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/specification/ProxySpecification.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/mapping/ObjectMapperSerializationContextImpl.groovy
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/specification/FilterableRequestSpecification.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/mapping/Jackson1Mapper.groovy
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/http/Header.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/ResponseLogSpecificationImpl.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/ValidatableResponseImpl.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/authentication/CertAuthScheme.groovy
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/log/LogRepository.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/http/Status.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/print/ResponsePrinter.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/authentication/PreemptiveBasicAuthScheme.groovy
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/filter/time/TimingFilter.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/http/HttpContextDecorator.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/response/ResponseBodyData.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/util/IOUtils.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/config/ObjectMapperConfig.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/authentication/OAuthScheme.groovy
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/config/OAuthConfig.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/config/DecoderConfig.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/mapping/JakartaEEMapper.groovy
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/config/MultiPartConfig.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-

jar/io/restassured/config/FailureConfig.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/matcher/ResponseAwareMatcher.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/UriValidator.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/support/CloseHttpClientConnectionInputStreamWrapper.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/filter/log/ErrorLoggingFilter.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/authentication/NoAuthScheme.groovy
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/filter/log/LogDetail.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/http/DeflateEncoding.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/mapping/JaxbMapper.groovy
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/http/CustomHttpMethod.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/assertion/HeaderMatcher.groovy
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/http/Headers.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/TrustAndKeystoreSpecImpl.groovy
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/RestAssuredResponseOptionsGroovyImpl.groovy
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/mapper/ObjectMapper.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/authentication/FormAuthScheme.groovy
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/mapper/ObjectMapperDeserializationContext.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/config/XmlConfig.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/spi/AuthFilter.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/support/ParameterUpdater.groovy
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/multipart/RestAssuredMultiPartEntity.groovy
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/filter/log/RequestLoggingFilter.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/http/CharsetExtractor.groovy
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/NoParameterValue.groovy
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-

jar/io/restassured/internal/RestAssuredResponseImpl.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/filter/OrderedFilter.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/MapCreator.groovy
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/http/HttpResponseContentTypeFinder.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/specification/RequestLogSpecification.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/assertion/ResponseTimeMatcher.groovy
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/config/MatcherConfig.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/authentication/PreemptiveAuthProvider.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/util/MatcherErrorMessageBuilder.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/matcher/xml/XmlDtdMatcher.groovy
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/config/RedirectConfig.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/response/Validatable.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/authentication/OAuth2Scheme.groovy
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/config/EncoderConfig.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/filter/cookie/CookieFilter.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/config/SessionConfig.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/http/ContentType.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/print/RequestPrinter.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/RestAssured.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/listener/ResponseValidationFailureListener.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/filter/Filter.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/specification/ResponseLogSpecification.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/mapping/ObjectMapperDeserializationContextImpl.groovy
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/filter/FormAuthFilter.groovy
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-

jar/io/restassured/internal/filter/SendRequestFilter.groovy
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/specification/MultiPartSpecification.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/builder/RequestSpecBuilder.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/config/RestAssuredConfig.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/filter/log/ResponseLoggingFilter.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/http/HttpResponseException.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/http/StringHashMap.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/http/URIBuilder.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/response/ValidatableResponseLogSpec.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/http/HTTPBuilder.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/config/SSLConfig.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/authentication/OAuthSignature.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/serialization/SerializationSupport.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/RedirectSpecificationImpl.groovy
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/util/SafeExceptionRethrower.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/config/ParamConfig.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/response/ValidatableResponse.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/config/HttpClientConfig.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/response/ValidatableResponseOptions.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/proxy/RestAssuredProxySelector.groovy
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/TrustAndKeystoreSpec.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/specification/SpecificationQuerier.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/matcher/DetailedCookieMatcher.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/http/HttpRequestFactory.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-

jar/io/restassured/assertion/DetailedCookieAssertion.groovy
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/builder/ResponseBuilder.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/filter/FilterContext.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/NameAndValue.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/http/ContentTypeExtractor.groovy
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/authentication/FormAuthConfig.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/SpecificationMerger.groovy
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/specification/Argument.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/authentication/AuthenticationScheme.groovy
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/http/EncoderRegistry.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/response/Response.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/config/ConnectionConfig.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/proxy/RestAssuredProxySelectorRoutePlanner.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/mapping/GsonMapper.groovy
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/TestSpecificationImpl.groovy
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/config/JsonConfig.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/http/ContentTypeSubTypeExtractor.groovy
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/multipart/MultiPartSpecificationImpl.groovy
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/http/Method.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/http/GZIPEncoding.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/authentication/CertificateAuthSettings.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/authentication/PreemptiveOAuth2HeaderScheme.groovy
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/mapping/ObjectMapping.groovy
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/config/Config.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-

jar/io/restassured/http/Cookie.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/response/ResponseBody.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/http/Cookies.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/filter/log/UrlDecoder.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/multipart/MultiPartInternal.groovy
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/specification/FilterableResponseSpecification.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/specification/RedirectSpecification.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/filter/log/StatusCodeBasedLoggingFilter.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/filter/session/SessionFilter.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/http/BasicNameValuePairWithNoValueSupport.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/parsing/Parser.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/http/AuthConfig.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/response/ResponseBodyExtractionOptions.java
No license file was found, but licenses were detected in source scan.

```
/*  
* Copyright 2022 the original author or authors.  
*  
* Licensed under the Apache License, Version 2.0 (the "License");  
* you may not use this file except in compliance with the License.  
* You may obtain a copy of the License at  
*  
* http://www.apache.org/licenses/LICENSE-2.0  
*  
* Unless required by applicable law or agreed to in writing, software  
* distributed under the License is distributed on an "AS IS" BASIS,  
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
* See the License for the specific language governing permissions and  
* limitations under the License.  
*/
```

Found in path(s):

* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/RestAssuredHttpBuilder.java
No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2020 the original author or authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/ResponseParserRegistrar.java
* /opt/cola/permits/1473460248_1668570766.8452125/0/rest-assured-5-1-0-sources-
jar/io/restassured/internal/mapping/JsonbMapper.groovy
```

1.227 aop-alliance 1.0

1.227.1 Available under license :

all the source code provided by AOP Alliance is Public Domain.

1.228 java-architecture-for-xml-binding 2.3.2

1.228.1 Available under license :

Copyright (c) 2004 Kohsuke Kawaguchi

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE

WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

/*

* * Copyright (c) YYYY Oracle and/or its affiliates. All rights reserved.

*

* This program and the accompanying materials are made available under the

* terms of the Eclipse Distribution License v. 1.0, which is available at

* <http://www.eclipse.org/org/documents/edl-v10.php>.

*

* SPDX-License-Identifier: BSD-3-Clause

*/

Copyright (c) 2018 Oracle and/or its affiliates. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the Eclipse Foundation, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2017 Oracle and/or its affiliates. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the Eclipse Foundation, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Notices for Eclipse Implementation of JAXB

This content is produced and maintained by the Eclipse Implementation of JAXB project.

* Project home: <https://projects.eclipse.org/projects/ee4j.jaxb-impl>

Trademarks

Eclipse Implementation of JAXB is a trademark of the Eclipse Foundation.

Copyright

All content is the property of the respective authors or their employers. For more information regarding authorship of content, please consult the listed source code repository logs.

Declared Project Licenses

This program and the accompanying materials are made available under the terms of the Eclipse Distribution License v. 1.0 which is available at <http://www.eclipse.org/org/documents/edl-v10.php>.

SPDX-License-Identifier: BSD-3-Clause

Source Code

The project maintains the following source code repositories:

- * <https://github.com/eclipse-ee4j/jaxb-ri>
- * <https://github.com/eclipse-ee4j/jaxb-istack-commons>
- * <https://github.com/eclipse-ee4j/jaxb-dtd-parser>
- * <https://github.com/eclipse-ee4j/jaxb-fi>
- * <https://github.com/eclipse-ee4j/jaxb-stax-ex>
- * <https://github.com/eclipse-ee4j/jax-rpc-ri>

Third-party Content

This project leverages the following third party content.

Apache Ant (1.10.2)

- * License: Apache-2.0 AND W3C AND LicenseRef-Public-Domain

Apache Ant (1.10.2)

- * License: Apache-2.0 AND W3C AND LicenseRef-Public-Domain

Apache Felix (1.2.0)

- * License: Apache License, 2.0

args4j (2.33)

- * License: MIT License

dom4j (1.6.1)

- * License: Custom license based on Apache 1.1

file-management (3.0.0)

- * License: Apache-2.0
- * Project: <https://maven.apache.org/shared/file-management/>
- * Source:
<https://svn.apache.org/viewvc/maven/shared/tags/file-management-3.0.0/>

JUnit (4.12)

- * License: Eclipse Public License

JUnit (4.12)

* License: Eclipse Public License

maven-compat (3.5.2)

* License: Apache-2.0

* Project: <https://maven.apache.org/ref/3.5.2/maven-compat/>

* Source:

<https://mvnrepository.com/artifact/org.apache.maven/maven-compat/3.5.2>

maven-core (3.5.2)

* License: Apache-2.0

* Project: <https://maven.apache.org/ref/3.5.2/maven-core/index.html>

* Source: <https://mvnrepository.com/artifact/org.apache.maven/maven-core/3.5.2>

maven-plugin-annotations (3.5)

* License: Apache-2.0

* Project: <https://maven.apache.org/plugin-tools/maven-plugin-annotations/>

* Source:

<https://github.com/apache/maven-plugin-tools/tree/master/maven-plugin-annotations>

maven-plugin-api (3.5.2)

* License: Apache-2.0

maven-resolver-api (1.1.1)

* License: Apache-2.0

maven-resolver-api (1.1.1)

* License: Apache-2.0

maven-resolver-connector-basic (1.1.1)

* License: Apache-2.0

maven-resolver-impl (1.1.1)

* License: Apache-2.0

maven-resolver-spi (1.1.1)

* License: Apache-2.0

maven-resolver-transport-file (1.1.1)

* License: Apache-2.0
* Project: <https://maven.apache.org/resolver/maven-resolver-transport-file/>
* Source:
<https://github.com/apache/maven-resolver/tree/master/maven-resolver-transport-file>

maven-resolver-util (1.1.1)

* License: Apache-2.0

maven-settings (3.5.2)

* License: Apache-2.0

* Source:

<https://mvnrepository.com/artifact/org.apache.maven/maven-settings/3.5.2>

OSGi Service Platform Core Companion Code (6.0)

* License: Apache License, 2.0

plexus-archiver (3.5)

* License: Apache-2.0

* Project: <https://codehaus-plexus.github.io/plexus-archiver/>

* Source: <https://github.com/codehaus-plexus/plexus-archiver>

plexus-io (3.0.0)

* License: Apache-2.0

plexus-utils (3.1.0)

* License: Apache- 2.0 or Apache- 1.1 or BSD or Public Domain or Indiana University Extreme! Lab Software License V1.1.1 (Apache 1.1 style)

relaxng-datatype (1.0)

* License: New BSD license

Sax (0.2)

* License: SAX-PD

* Project: <http://www.megginson.com/downloads/SAX/>

* Source: http://sourceforge.net/project/showfiles.php?group_id=29449

testng (6.14.2)

* License: Apache-2.0 AND (MIT OR GPL-1.0+)

* Project: <https://testng.org/doc/index.html>

* Source: <https://github.com/cbeust/testng>

wagon-http-lightweight (3.0.0)

* License: Pending

* Project: <https://maven.apache.org/wagon/>

* Source:

<https://mvnrepository.com/artifact/org.apache.maven.wagon/wagon-http-lightweight/3.0.0>

xz for java (1.8)

* License: LicenseRef-Public-Domain

Cryptography

Content may contain encryption software. The country in which you are currently may have restrictions on the import, possession, and use, and/or re-export to another country, of encryption software. BEFORE using any encryption software, please check the country's laws, regulations and policies concerning the import, possession, or use, and re-export of encryption software, to see if this is permitted.

/*

* Copyright (c) YYYY Oracle and/or its affiliates. All rights reserved.

*

* This program and the accompanying materials are made available under the

* terms of the Eclipse Distribution License v. 1.0, which is available at

* <http://www.eclipse.org/org/documents/edl-v10.php>.

*

* SPDX-License-Identifier: BSD-3-Clause

*/

1.229 jackson-module:-guice 2.14.0

1.229.1 Available under license :

Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library.

It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has been in development since 2007.

It is currently developed by a community of developers, as well as supported commercially by FasterXML.com.

Licensing

Jackson core and extension components may licensed under different licenses.

To find the details that apply to this artifact see the accompanying LICENSE file.

For more information, including possible other licensing options, contact

FasterXML.com (<http://fasterxml.com>).

Credits

A list of contributors may be found from CREDITS file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.

This copy of Jackson JSON processor `jackson-module-guice` module is licensed under the Apache (Software) License, version 2.0 ("the License").

See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

<http://www.apache.org/licenses/LICENSE-2.0>

1.230 micronaut-test-resources 1.1.2

1.230.1 Available under license :

Apache License

Version 2.0, January 2004

<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made,

use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability

incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

mcr.microsoft.com/mssql/server:2019-CU16-GDR1-ubuntu-20.04

1.231 maven-artifact 3.8.3

1.231.1 Available under license :

Maven Artifact
Copyright 2001-2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent

to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work,

excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any

risks associated with Your exercise of permissions under this License.

8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.232 resilience4j 0.13.1

1.232.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 *
 * Copyright 2017: Robert Winkler
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 *
 *
 */
```

Found in path(s):

```
* /opt/cola/permits/1274699229_1654840286.013314/0/resilience4j-core-0-13-1-sources-
jar/io/github/resilience4j/core/Event.java
* /opt/cola/permits/1274699229_1654840286.013314/0/resilience4j-core-0-13-1-sources-
jar/io/github/resilience4j/core/EventProcessor.java
* /opt/cola/permits/1274699229_1654840286.013314/0/resilience4j-core-0-13-1-sources-
jar/io/github/resilience4j/core/EventConsumer.java
* /opt/cola/permits/1274699229_1654840286.013314/0/resilience4j-core-0-13-1-sources-
jar/io/github/resilience4j/core/EventPublisher.java
* /opt/cola/permits/1274699229_1654840286.013314/0/resilience4j-core-0-13-1-sources-
jar/io/github/resilience4j/core/StopWatch.java
```

1.233 micronaut-microstream 1.2.0

1.233.1 Available under license :

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner

or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions

of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.234 javax-inject1-as-osgi-bundle 2.6.1

1.234.1 Available under license :

Eclipse Public License - v 2.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial content Distributed under this Agreement, and

b) in the case of each subsequent Contributor:

- i) changes to the Program, and
- ii) additions to the Program;

where such changes and/or additions to the Program originate from and are Distributed by that particular Contributor. A Contribution "originates" from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include changes or additions to the Program that are not Modified Works.

"Contributor" means any person or entity that Distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions Distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

"Derivative Works" shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the

editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

"Modified Works" shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

"Distribute" means the acts of a) distributing or b) making available in any manner that enables the transfer of a copy.

"Source Code" means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Secondary License" means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity.

Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

e) Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

3. REQUIREMENTS

3.1 If a Contributor Distributes the Program in any form, then:

a) the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and

b) the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:

i) effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and

iv) requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements

of this section 3.

3.2 When the Program is Distributed as Source Code:

- a) it must be made available under this Agreement, or if the Program (i) is combined with other material in a separate file or files made available under a Secondary License, and (ii) the initial Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available under the terms of such Secondary Licenses, and
- b) a copy of this Agreement must be included with each copy of the Program.

3.3 Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability ("notices") contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance

claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.

Exhibit A - Form of Secondary Licenses Notice

"This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: {name license(s), version(s), and exceptions or additional permissions here}."

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor
Boston, MA 02110-1335
USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we

want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices

stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost

of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to

these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey

the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1335 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type
`show w'. This is free software, and you are welcome to redistribute
it under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the
program `Gnomovision' (which makes passes at compilers) written by
James Hacker.
```

```
signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you

may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

CLASSPATH EXCEPTION

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

Notices for Eclipse GlassFish

This content is produced and maintained by the Eclipse GlassFish project.

* Project home: <https://projects.eclipse.org/projects/ee4j.glassfish>

Trademarks

Eclipse GlassFish, and GlassFish are trademarks of the Eclipse Foundation.

Copyright

All content is the property of the respective authors or their employers. For more information regarding authorship of content, please consult the listed source code repository logs.

Declared Project Licenses

This program and the accompanying materials are made available under the terms of the Eclipse Public License v. 2.0 which is available at <http://www.eclipse.org/legal/epl-2.0>. This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License v. 2.0 are satisfied: GNU General Public License, version 2 with the GNU Classpath Exception which is

available at <https://www.gnu.org/software/classpath/license.html>.

SPDX-License-Identifier: EPL-2.0 OR GPL-2.0 WITH Classpath-exception-2.0

Source Code

The project maintains the following source code repositories:

- * <https://github.com/eclipse-ee4j/glassfish-ha-api>
- * <https://github.com/eclipse-ee4j/glassfish-logging-annotation-processor>
- * <https://github.com/eclipse-ee4j/glassfish-shoal>
- * <https://github.com/eclipse-ee4j/glassfish-cdi-porting-tck>
- * <https://github.com/eclipse-ee4j/glassfish-jsftemplating>
- * <https://github.com/eclipse-ee4j/glassfish-hk2-extra>
- * <https://github.com/eclipse-ee4j/glassfish-hk2>
- * <https://github.com/eclipse-ee4j/glassfish-fighterfish>

Third-party Content

This project leverages the following third party content.

None

Cryptography

Content may contain encryption software. The country in which you are currently may have restrictions on the import, possession, and use, and/or re-export to another country, of encryption software. BEFORE using any encryption software, please check the country's laws, regulations and policies concerning the import, possession, or use, and re-export of encryption software, to see if this is permitted.

1.235 fabric8-:::kubernetes-model-:::api-extensions 4.13.3

1.235.1 Available under license :

No license file was found, but licenses were detected in source scan.

*

* Copyright (C) 2015 Red Hat, Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

- *
- * Unless required by applicable law or agreed to in writing, software
- * distributed under the License is distributed on an "AS IS" BASIS,
- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.
- *

Found in path(s):

* /opt/cola/permits/1288519788_1647861625.92/0/kubernetes-model-apiextensions-4-13-3-jar/manifest.vm

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0

Bnd-LastModified: 1619068527673

Build-Jdk-Spec: 1.8

Bundle-Description: Java client for Kubernetes and OpenShift

Bundle-DocURL: <http://redhat.com>

Bundle-License: <http://www.apache.org/licenses/LICENSE-2.0.txt>

Bundle-ManifestVersion: 2

Bundle-Name: Fabric8 :: Kubernetes Model :: API Extensions

Bundle-SymbolicName: io.fabric8.kubernetes-model-apiextensions

Bundle-Vendor: Red Hat

Bundle-Version: 4.13.3

Created-By: Apache Maven Bundle Plugin

Export-Package: io.fabric8.kubernetes.api.model.apiextensions.v1;uses:="com.fasterxml.jackson.annotation,com.fasterxml.jackson.core,com.fasterxml.jackson.databind,com.fasterxml.jackson.databind.annotation,io.fabric8.kubernetes.api.builder,io.fabric8.kubernetes.api.model,io.fabric8.kubernetes.model.annotation";version="4.13.3",io.fabric8.kubernetes.api.model.apiextensions.v1beta1;uses:="com.fasterxml.jackson.annotation,com.fasterxml.jackson.core,com.fasterxml.jackson.databind,com.fasterxml.jackson.databind.annotation,io.fabric8.kubernetes.api.builder,io.fabric8.kubernetes.api.model,io.fabric8.kubernetes.model.annotation";version="4.13.3"

Implementation-Title: Fabric8 :: Kubernetes Model :: API Extensions

Implementation-Vendor: Red Hat

Implementation-Version: 4.13.3

Import-Package: com.fasterxml.jackson.annotation;version="[2.11,3)",com.fasterxml.jackson.core;version="[2.11,3)",com.fasterxml.jackson.core.type;version="[2.11,3)",com.fasterxml.jackson.databind;version="[2.11,3)",com.fasterxml.jackson.databind.annotation;version="[2.11,3)",io.fabric8.kubernetes.api.builder;version="[4.13,5)",io.fabric8.kubernetes.api.model,io.fabric8.kubernetes.model.annotation;version="[4.13,5)"

Require-Capability: osgi.ee;filter="(&(osgi.ee=JavaSE)(version=1.8))"

Specification-Title: Fabric8 :: Kubernetes Model :: API Extensions

Specification-Vendor: Red Hat

Specification-Version: 4.13

Tool: Bnd-5.1.1.202006162103

Found in path(s):

* /opt/cola/permits/1288519788_1647861625.92/0/kubernetes-model-apiextensions-4-13-3-jar/META-INF/MANIFEST.MF

No license file was found, but licenses were detected in source scan.

Copyright (C) 2015 Red Hat, Inc.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE>

2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Found in path(s):

* /opt/cola/permits/1288519788_1647861625.92/0/kubernetes-model-apiextensions-4-13-3-jar/META-INF/maven/io.fabric8/kubernetes-model-apiextensions/pom.xml

1.236 io-grpc-grpc-okhttp 1.47.0

1.236.1 Available under license :

Envoy

Copyright The Envoy Project Authors

Licensed under Apache License 2.0. See LICENSE for terms.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special,

incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner].

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

protoc-gen-validate

Copyright 2019 Envoy Project Authors

Licensed under Apache License 2.0. See LICENSE for terms.

zero-allocation-hashing

Copyright 2015 Higher Frequency Trading <http://www.higherfrequencytrading.com>

Licensed under Apache License 2.0. See LICENSE for terms.

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain

separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the

origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

/*

* Copyright 2015 The gRPC Authors

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Copyright 2014 The gRPC Authors

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

This product contains a modified portion of 'OkHttp', an open source
HTTP & SPDY client for Android and Java applications, which can be obtained
at:

- * LICENSE:
 - * [okhttp/third_party/okhttp/LICENSE](#) (Apache License 2.0)
- * HOMEPAGE:
 - * <https://github.com/square/okhttp>
- * LOCATION_IN_GRPC:
 - * [okhttp/third_party/okhttp](#)

This product contains a modified portion of 'Envoy', an open source cloud-native high-performance edge/middle/service proxy, which can be obtained at:

- * LICENSE:
 - * [xds/third_party/envoy/LICENSE](#) (Apache License 2.0)
- * NOTICE:
 - * [xds/third_party/envoy/NOTICE](#)
- * HOMEPAGE:
 - * <https://www.envoyproxy.io>
- * LOCATION_IN_GRPC:
 - * [xds/third_party/envoy](#)

This product contains a modified portion of 'protoc-gen-validate (PGV)', an open source protoc plugin to generate polyglot message validators, which can be obtained at:

- * LICENSE:
 - * [xds/third_party/protoc-gen-validate/LICENSE](#) (Apache License 2.0)
- * NOTICE:
 - * [xds/third_party/protoc-gen-validate/NOTICE](#)
- * HOMEPAGE:
 - * <https://github.com/envoyproxy/protoc-gen-validate>
- * LOCATION_IN_GRPC:
 - * [xds/third_party/protoc-gen-validate](#)

This product contains a modified portion of 'udpa', an open source universal data plane API, which can be obtained at:

- * LICENSE:
 - * [xds/third_party/udpa/LICENSE](#) (Apache License 2.0)
- * HOMEPAGE:
 - * <https://github.com/cncf/udpa>
- * LOCATION_IN_GRPC:
 - * [xds/third_party/udpa](#)

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and

attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the

appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.237 libplexus-utils 3.3.0

1.237.1 Available under license :

Indiana University Extreme! Lab Software License

Version 1.1.1

Copyright (c) 2002 Extreme! Lab, Indiana University. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright notice,
this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in
the documentation and/or other materials provided with the distribution.
3. The end-user documentation included with the redistribution, if any,
must include the following acknowledgment:

"This product includes software developed by the Indiana University
Extreme! Lab (<http://www.extreme.indiana.edu/>)."

Alternately, this acknowledgment may appear in the software itself,
if and wherever such third-party acknowledgments normally appear.

4. The names "Indiana University" and "Indiana University Extreme! Lab"
must not be used to endorse or promote products derived from this
software without prior written permission. For written permission,
please contact <http://www.extreme.indiana.edu/>.

5. Products derived from this software may not use "Indiana University"
name nor may "Indiana University" appear in their name, without prior
written permission of the Indiana University.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHORS, COPYRIGHT HOLDERS OR ITS CONTRIBUTORS

BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

/*

*/

* CruiseControl, a Continuous Integration Toolkit

* Copyright (c) 2001-2003, ThoughtWorks, Inc.

* 651 W Washington Ave. Suite 500

* Chicago, IL 60661 USA

* All rights reserved.

*

* Redistribution and use in source and binary forms, with or without

* modification, are permitted provided that the following conditions

* are met:

*

* + Redistributions of source code must retain the above copyright

* notice, this list of conditions and the following disclaimer.

*

* + Redistributions in binary form must reproduce the above

* copyright notice, this list of conditions and the following

* disclaimer in the documentation and/or other materials provided

* with the distribution.

*

* + Neither the name of ThoughtWorks, Inc., CruiseControl, nor the

* names of its contributors may be used to endorse or promote

* products derived from this software without specific prior

* written permission.

*

* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS

* "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT

* LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR

* A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR

* CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,

* EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,

* PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR

* PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF

* LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING

* NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS

* SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

This product includes software developed by the Indiana University

Extreme! Lab (<http://www.extreme.indiana.edu/>).

This product includes software developed by

The Apache Software Foundation (<http://www.apache.org/>).

This product includes software developed by
ThoughtWorks (<http://www.thoughtworks.com>).

This product includes software developed by
javolution (<http://javolution.org/>).

This product includes software developed by
Rome (<https://rome.dev.java.net/>).
Javolution - Java(TM) Solution for Real-Time and Embedded Systems
Copyright (c) 2006, Javolution (<http://javolution.org>)
All rights reserved.

Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice,
this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON
ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity

on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one

of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a

result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.238 feign-jaxrs 8.18.0

1.238.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2013 Netflix, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

`/opt/cola/permits/1274698784_1645233710.69/0/feign-jaxrs-8-18-0-sources-jar/feign/jaxrs/JAXRSContract.java`

1.239 rest-assured-common 5.2.0

1.239.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2019, 2020 the original author or authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

`/opt/cola/permits/1473460002_1668478089.1930833/0/rest-assured-common-5-2-0-sources-jar/io/restassured/common/mapper/resolver/ObjectMapperResolver.java`

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2019 the original author or authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1473460002_1668478089.1930833/0/rest-assured-common-5-2-0-sources-
jar/io/restassured/internal/common/assertion/PathFragmentEscaper.groovy
* /opt/cola/permits/1473460002_1668478089.1930833/0/rest-assured-common-5-2-0-sources-
jar/io/restassured/internal/common/path/ObjectConverter.groovy
* /opt/cola/permits/1473460002_1668478089.1930833/0/rest-assured-common-5-2-0-sources-
jar/io/restassured/internal/common/assertion/AssertParameter.groovy
* /opt/cola/permits/1473460002_1668478089.1930833/0/rest-assured-common-5-2-0-sources-
jar/io/restassured/internal/common/classpath/ClassPathResolver.java
* /opt/cola/permits/1473460002_1668478089.1930833/0/rest-assured-common-5-2-0-sources-
jar/io/restassured/internal/common/assertion/GetAtPathFragmentEscaper.groovy
* /opt/cola/permits/1473460002_1668478089.1930833/0/rest-assured-common-5-2-0-sources-
jar/io/restassured/internal/common/assertion/HyphenQuoteFragmentEscaper.groovy
* /opt/cola/permits/1473460002_1668478089.1930833/0/rest-assured-common-5-2-0-sources-
jar/io/restassured/common/exception/PathException.java
* /opt/cola/permits/1473460002_1668478089.1930833/0/rest-assured-common-5-2-0-sources-
jar/io/restassured/internal/common/assertion/Assertion.groovy
* /opt/cola/permits/1473460002_1668478089.1930833/0/rest-assured-common-5-2-0-sources-
jar/io/restassured/internal/common/assertion/AssertionSupport.groovy
* /opt/cola/permits/1473460002_1668478089.1930833/0/rest-assured-common-5-2-0-sources-
jar/io/restassured/common/mapper/ObjectDeserializationContext.java
* /opt/cola/permits/1473460002_1668478089.1930833/0/rest-assured-common-5-2-0-sources-
jar/io/restassured/internal/common/assertion/EndToEndQuoteFragmentEscaper.groovy
* /opt/cola/permits/1473460002_1668478089.1930833/0/rest-assured-common-5-2-0-sources-
jar/io/restassured/common/mapper/DataToDeserialize.java
* /opt/cola/permits/1473460002_1668478089.1930833/0/rest-assured-common-5-2-0-sources-
jar/io/restassured/common/mapper/TypeRef.java
* /opt/cola/permits/1473460002_1668478089.1930833/0/rest-assured-common-5-2-0-sources-
jar/io/restassured/common/mapper/factory/ObjectMapperFactory.java
```

1.240 micronaut-rxjava-3 2.3.0

1.240.1 Available under license :

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes

of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You

meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor,

except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.241 abegocore 1.0.3

1.241.1 Available under license :

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0

Bnd-LastModified: 1446804256758

Build-Jdk: 1.8.0_60

Built-By: ub

Bundle-Description: Efficient and customizable TreeLayout Algorithm in J
ava.

Bundle-DocURL: <http://abego-software.de>

Bundle-License: <http://www.abego-software.de/legal/apl-v10.html>

Bundle-ManifestVersion: 2

Bundle-Name: abego TreeLayout Core

Bundle-SymbolicName: org.abego.treelayout.core

Bundle-Vendor: abego Software GmbH, Germany

Bundle-Version: 1.0.3

Created-By: Apache Maven Bundle Plugin

Export-Package: org.abego.treelayout.util;uses:="org.abego.treelayout";v
ersion="1.0.3",org.abego.treelayout;version="1.0.3"

Implementation-Title: abego TreeLayout Core

Implementation-Vendor: abego Software GmbH, Germany

Implementation-Vendor-Id: org.abego.treelayout

Implementation-Version: 1.0.3

Import-Package: sun.misc;resolution:=optional

Specification-Title: abego TreeLayout Core

Specification-Vendor: abego Software GmbH, Germany

Specification-Version: 1.0.3

Tool: Bnd-1.50.0

Found in path(s):

1.242 spock-framework---bill-of-materials 2.1-groovy-3.0

1.242.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2012 the original author or authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *   http://www.apache.org/licenses/LICENSE-2.0
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

package dsld

import org.codehaus.groovy.ast.expr.*

assertVersion(groovyEclipse: "2.7.2") // tested against this version

// with(foo) { /* delegates to foo */ }
// currently only works if foo is statically typed
// with(foo, Bar) { /* delegates to instance of Bar */ }
contribute(
    enclosingClass(subType("spock.lang.Specification")) &
    enclosingCallDeclaringType("spock.lang.Specification") &
    bind(theCalls: enclosingCall(name("with"))) &
    enclosingClosure() {
    def args = theCalls.iterator().next().arguments.expressions
    if (args.size() == 2) {
        delegatesTo(args[0].type)
    } else if (args.size() == 3) {
        if (args[1] instanceof ClassExpression) {
            delegatesTo(args[1])
        }
    }
    }
}
```

```

// Mock(Foo) { /* delegates to instance of type Foo */ }
// works for all mock factory methods and all overloads that include a type
contribute(
  enclosingClass(subType("spock.lang.Specification")) &
  enclosingCallDeclaringType("spock.mock.MockingApi") &
  bind(theCalls: enclosingCall(name("Mock") | name("Stub") | name("Spy") | name("GroovyMock") |
name("GroovyStub") | name("GroovySpy")))) &
  enclosingClosure() {
  def args = theCalls.iterator().next().arguments.expressions
  def mockType = args.find { it instanceof ClassExpression }
  if (mockType) {
    delegatesTo(mockType)
  }
}
}

```

Found in path(s):

```
* /opt/cola/permits/1343420207_1655131745.3800905/0/spock-core-2-1-groovy-3-0-jar/dsld/spk.dsld
```

No license file was found, but licenses were detected in source scan.

```

/*
 * Copyright 2012 the original author or authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *   http://www.apache.org/licenses/LICENSE-2.0
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

```

```
package org.spockframework.idea
```

```
import com.intellij.psi.PsiVariable
```

```
// only tested with 11.1, so play it safe and bail out on earlier version
```

```
if (!supportsVersion("11.1")) return
```

```
def ctx = context(ctype: "spock.lang.Specification", scope: closureScope(isArg: true))
```

```
// Mock(Foo) { /* delegates to instance of type Foo */ }
```

```
// Foo x = Mock { /* delegates to instance of type Foo */ }
```

```
contributor(ctx, {
```

```
  def call = enclosingCall("Mock") ?: enclosingCall("Stub") ?: enclosingCall("Spy") ?:
```

```
    enclosingCall("GroovyMock") ?: enclosingCall("GroovyStub") ?: enclosingCall("GroovySpy")
```

```

if (call) {
  def method = call.bind()
  def clazz = method?.containingClass
  if (clazz?.qualName == "spock.mock.MockingApi") {
    def mockType = call.arguments.find { it.type?.className == "Class" }
    if (mockType) {
      def typeParameter = mockType.type.parameters[0]
      delegatesTo(typeParameter.resolve())
      return
    }
  }
  def callContext = call.context
  if (callContext instanceof PsiVariable) {
    delegatesTo(callContext.typeGroovy.resolve())
  }
}
})

```

Found in path(s):

```

* /opt/cola/permits/1343420207_1655131745.3800905/0/spock-core-2-1-groovy-3-0-
jar/org/spockframework/idea/spock.gdsl

```

1.243 hamcrest 1.3

1.243.1 Available under license :

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
<HEAD>
<TITLE>Common Public License - v 1.0</TITLE>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</HEAD>

<BODY BGCOLOR="#FFFFFF" VLINK="#800000">

<P ALIGN="CENTER"><B>Common Public License - v 1.0</B>
<P><B></B><FONT SIZE="3"></FONT>
<P><FONT SIZE="3"></FONT><FONT SIZE="2">THE ACCOMPANYING PROGRAM IS PROVIDED
UNDER THE TERMS OF THIS COMMON PUBLIC LICENSE ("AGREEMENT"). ANY USE,
REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE
OF THIS AGREEMENT.</FONT>
<P><FONT SIZE="2"></FONT>
<P><FONT SIZE="2"><B>1. DEFINITIONS</B></FONT>
<P><FONT SIZE="2">"Contribution" means:</FONT>

<UL><FONT SIZE="2">a) in the case of the initial Contributor, the initial code and documentation distributed
under this Agreement, and<BR CLEAR="LEFT">

```

b) in the case of each subsequent Contributor:

i) changes to the Program, and

ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents " mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the

Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

<P>3. REQUIREMENTS

<P>A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:

i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and

iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

<P>When the Program is made available in source code form:

a) it must be made available under this Agreement; and

b) a copy of this Agreement must be included with each copy of the Program.

<P><STRIKE></STRIKE>
<P><STRIKE></STRIKE>Contributors may not remove or alter any copyright notices contained within the Program.

<P>

<P>Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

<P>

<P>4. COMMERCIAL DISTRIBUTION

<P>Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

<P>

<P>For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

<P>

<P>5. NO WARRANTY

<P>EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

<P>

<P>6. DISCLAIMER OF LIABILITY

<P>EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

<P>

<P>7. GENERAL

<P>If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

<P>

<P>If Recipient institutes patent litigation against a Contributor with respect to a patent applicable to software (including a cross-claim or counterclaim in a lawsuit), then any patent licenses granted by that Contributor to such Recipient under this Agreement shall terminate as of the date such litigation is filed. In addition, if Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

<P>

<P>All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

<P>

<P>

<P>Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. IBM is the initial Agreement Steward. IBM may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. <FONT

Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

BSD License

Copyright (c) 2000-2006, www.hamcrest.org

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of Hamcrest nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN

ANY

WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2000-2003, jMock.org

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of jMock nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```
<?xml version="1.0" encoding="iso-8859-1"?>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

```
<title>EasyMock License</title>
```

```
<link rel="stylesheet" href="easymock.css" />
```

```
</head>
```

```
<body><div class="bodywidth">
```

```
<h2>
```

```
EasyMock 2 License (MIT License)
```

</h2>

Copyright (c) 2001-2006 OFFIS, Tammo Freese.

<p>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

</p>

<p>

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

</p>

<p>

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

</p>

</div>

</body>

</html>

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of,

publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution

notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing

the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.244 micronaut-coherence 3.5.1

1.244.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
#  
# Copyright 2017-2020 original authors  
#  
# Licensed under the Apache License, Version 2.0 (the "License");  
# you may not use this file except in compliance with the License.
```

```
# You may obtain a copy of the License at
#
# https://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#
```

io.micronaut.coherence.grpc.GrpcServerConfigurer\$GrpcServerConfigurationBean

Found in path(s):

```
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/META-INF/services/com.oracle.coherence.grpc.proxy.GrpcServerConfiguration
No license file was found, but licenses were detected in source scan.
```

```
#
# Copyright 2017-2020 original authors
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
# https://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#
```

io.micronaut.coherence.config.MicronautSystemPropertyResolver

Found in path(s):

```
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/META-INF/services/com.tangosol.coherence.config.SystemPropertyResolver
No license file was found, but licenses were detected in source scan.
```

```
/*
* Copyright 2017-2020 original authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
```

*
* <https://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/config/CoherenceDefaultProperties.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/namespace/BeanProcessor.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/InjectorImpl.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/grpc/package-info.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/messaging/package-info.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/SessionType.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/AbstractSessionConfigurationBean.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/grpc/GrpcServerBuilder.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/JsonSerializerFactory.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/SessionConfigurationProvider.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/namespace/MicronautNamespaceHandler.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/namespace/package-info.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/CoherenceContext.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/SessionConfigurationBean.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/messaging/SubscriberExecutorServiceConfig.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/SerializerFactories.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/grpc/GrpcServerConfigurer.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/package-info.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-

jar/io/micronaut/coherence/config/MicronautSystemPropertyResolver.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-
jar/io/micronaut/coherence/messaging/TopicKey.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-
jar/io/micronaut/coherence/namespace/BeanBuilder.java
No license file was found, but licenses were detected in source scan.

```
#  
# Copyright 2017-2020 original authors  
#  
# Licensed under the Apache License, Version 2.0 (the "License");  
# you may not use this file except in compliance with the License.  
# You may obtain a copy of the License at  
#  
# https://www.apache.org/licenses/LICENSE-2.0  
#  
# Unless required by applicable law or agreed to in writing, software  
# distributed under the License is distributed on an "AS IS" BASIS,  
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
# See the License for the specific language governing permissions and  
# limitations under the License.  
#
```

io.micronaut.coherence.grpc.GrpcServerBuilder\$GrpcServerBuilderProviderBean

Found in path(s):

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/META-INF/services/com.oracle.coherence.grpc.proxy.GrpcServerBuilderProvider
No license file was found, but licenses were detected in source scan.

```
/*  
* Copyright 2017-2022 original authors  
*  
* Licensed under the Apache License, Version 2.0 (the "License");  
* you may not use this file except in compliance with the License.  
* You may obtain a copy of the License at  
*  
* https://www.apache.org/licenses/LICENSE-2.0  
*  
* Unless required by applicable law or agreed to in writing, software  
* distributed under the License is distributed on an "AS IS" BASIS,  
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
* See the License for the specific language governing permissions and  
* limitations under the License.  
*/
```

Found in path(s):

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-

jar/io/micronaut/coherence/event/AnnotatedMapListener.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-
jar/io/micronaut/coherence/annotation/AbstractArrayLiteral.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-
jar/io/micronaut/coherence/messaging/CoherencePublisherIntroductionAdvice.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-
jar/io/micronaut/coherence/FactoryQualifier.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-
jar/io/micronaut/coherence/annotation/AbstractNamedLiteral.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2017-2021 original authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <https://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-
jar/io/micronaut/coherence/annotation/Lost.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-
jar/io/micronaut/coherence/annotation/SessionName.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-
jar/io/micronaut/coherence/event/EventObserverSupport.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-
jar/io/micronaut/coherence/annotation/Deleted.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-
jar/io/micronaut/coherence/messaging/binders/ElementArgumentBinderRegistry.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-
jar/io/micronaut/coherence/annotation/CoherenceTopicListener.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-
jar/io/micronaut/coherence/annotation/WhereFilter.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-
jar/io/micronaut/coherence/annotation/Starting.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-
jar/io/micronaut/coherence/annotation/Executed.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-
jar/io/micronaut/coherence/annotation/Departing.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/Recovered.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/Destroyed.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/MapName.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/messaging/binders/TypedElementBinder.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/Syncing.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/Synchronous.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/Updated.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/PropertyExtractor.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/ServiceName.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/CommittingLocal.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/Disposing.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/ConfigUri.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/messaging/CoherenceTopicListenerProcessor.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/ParticipantName.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/Executing.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/Utils.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/Activated.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/NamedCacheFactories.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/CoherencePublisher.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/Error.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/Removing.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/Replicating.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/event/ExecutableMethodMapListener.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/Processor.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/Backlog.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/Scope.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/NamedTopicFactories.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/SubscriberGroup.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/CacheName.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/Topic.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/Truncated.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/Connecting.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/event/package-info.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/Updating.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/Committing.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/FilterFactory.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/Inserting.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/Arrived.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/MapEventTransformerBinding.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/event/EventArgumentBinder.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/PofExtractor.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/ExtractorFactories.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/Lite.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/View.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/AnnotationLiteral.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/CoherenceEventListener.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/messaging/binders/AnnotatedElementBinder.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/SerializerFormat.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/FilterBinding.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/FilterFactories.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/Synced.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/messaging/exceptions/CoherenceSubscriberException.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/CoherenceFactory.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/event/ExecutableMethodEventObserver.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/Departed.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/event/EventArgumentBinderRegistry.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/messaging/binders/ElementBinder.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/Name.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/messaging/binders/DefaultTopicBinder.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/messaging/SubscriberExceptionHandler.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/ExtractorBinding.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/event/BaseExecutableMethodObserver.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/MapEventTransformerFactories.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/MapEventTransformerFactory.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/Started.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/Assigned.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/Committed.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/ExtractorFactory.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/Rollback.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/AlwaysFilter.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/Activating.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/Created.java

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/Topics.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/Inserted.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/Disconnected.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/ScopeName.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/CommitStrategy.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/messaging/binders/ElementValueBinder.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/Removed.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/CommittingRemote.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/ChainedExtractor.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/Stopped.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/event/CoherenceEventListenerProcessor.java
* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/io/micronaut/coherence/annotation/Stopping.java

No license file was found, but licenses were detected in source scan.

```
#  
# Copyright 2017-2020 original authors  
#  
# Licensed under the Apache License, Version 2.0 (the "License");  
# you may not use this file except in compliance with the License.  
# You may obtain a copy of the License at  
#  
# https://www.apache.org/licenses/LICENSE-2.0  
#  
# Unless required by applicable law or agreed to in writing, software  
# distributed under the License is distributed on an "AS IS" BASIS,  
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
# See the License for the specific language governing permissions and  
# limitations under the License.  
#
```

io.micronaut.coherence.InjectorImpl

Found in path(s):

* /opt/cola/permits/1425207060_1663952768.0147243/0/micronaut-coherence-3-5-1-sources-jar/META-INF/services/com.oracle.coherence.inject.Injector

1.245 prometheus-java-span-context-supplier-opentelemetry-agent 0.15.0

1.245.1 Available under license :

No license file was found, but licenses were detected in source scan.

<url><http://www.apache.org/licenses/LICENSE-2.0.txt></url>

Found in path(s):

* /opt/cola/permits/1341561705_1654803143.8356545/0/simpleclient-tracer-otel-agent-0-15-0-sources-jar/META-INF/maven/io.prometheus/simpleclient_tracer_otel_agent/pom.xml

1.246 jackson-dataformat-csv 2.14.0

1.246.1 Available under license :

This copy of Jackson JSON processor databind module is licensed under the Apache (Software) License, version 2.0 ("the License").

See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

<http://www.apache.org/licenses/LICENSE-2.0>

Jackson is licensed under [Apache License 2.0](<http://www.apache.org/licenses/LICENSE-2.0.txt>).

The TOML grammar and some parts of the test suite are adapted from the TOML project, which is licensed under the MIT license:

The MIT License

Copyright (c) Tom Preston-Werner

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The JFlex lexer generator is licensed under a 3-clause BSD-style license:

JFlex - Copying, Warranty & License

=====

JFlex is free software, since version 1.5 published under the terms of this [3-clause BSD-style license](<https://opensource.org/licenses/BSD-3-Clause>).

There is absolutely NO WARRANTY for JFlex, its code and its documentation.

Copyright (c) Gerwin Klein, Steve Rowe, Rgis Dcamps, Google LLC.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the names of the authors nor the names of JFlex contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER

IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF

THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This copy of Jackson JSON processor YAML module is licensed under the Apache (Software) License, version 2.0 ("the License").

See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

<http://www.apache.org/licenses/LICENSE-2.0>

Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library. It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has been in development since 2007.

It is currently developed by a community of developers, as well as supported commercially by FasterXML.com.

Licensing

Jackson core and extension components may be licensed under different licenses. To find the details that apply to this artifact see the accompanying LICENSE file. For more information, including possible other licensing options, contact FasterXML.com (<http://fasterxml.com>).

Credits

A list of contributors may be found from CREDITS file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common

control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or

documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill,

work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.247 jackson-bom 2.14.0

1.247.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or

Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/Function4.java
* /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/CheckedConsumer.java
* /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/CheckedFunction4.java
* /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/collection/IndexedSeq.java
* /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/CheckedFunction8.java
* /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/collection/Multimaps.java
* /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/control/Try.java
* /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/Tuple3.java
* /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/collection/ArrayType.java
* /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/collection/TreeSet.java
* /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/Tuple8.java
* /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/collection/HashMap.java
* /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/control/Option.java
* /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/collection/AbstractIterator.java
* /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/collection/HashMapMappedTrie.java
* /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/collection/CharSeq.java
* /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/concurrent/Promise.java
* /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/Function2.java
* /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/concurrent/GuardedBy.java
* /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/Predicates.java
* /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/Function6.java
* /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/collection/HashSet.java
* /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/collection/Iterator.java
* /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/CheckedFunction7.java
* /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/Function8.java
* /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/Tuple7.java
* /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/CheckedRunnable.java
* /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/Value.java
* /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/collection/SortedMap.java
* /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/collection/HashMultimap.java
* /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/control/Validation.java
* /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/collection/SortedMultimap.java
* /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/collection/Collections.java
* /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/control/Either.java
* /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/Tuple2.java
* /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/CheckedFunction0.java
* /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/collection/Traversable.java
* /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/\$.java
* /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/collection/Map.java
* /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/Tuple5.java
* /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/collection/Queue.java
* /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/CheckedFunction1.java
* /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/collection/TreeMap.java

* /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/CheckedFunction2.java
 * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/PartialFunction.java
 * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/Tuple4.java
 * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/collection/JavaConverters.java
 * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/Function3.java
 * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/collection/TreeMultimap.java
 * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/collection/BitMappedTrie.java
 * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/collection/LinkedHashSet.java
 * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/collection/Array.java
 * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/GwtIncompatible.java
 * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/CheckedFunction6.java
 * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/collection/Multimap.java
 * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/Tuple1.java
 * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/collection/Set.java
 * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/collection/BitSet.java
 * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/collection/GwtIncompatible.java
 * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/CheckedPredicate.java
 * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/collection/Tree.java
 * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/Lazy.java
 * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/collection/Maps.java
 * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/CheckedFunction5.java
 * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/Function5.java
 * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/Function1.java
 * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/concurrent/GwtIncompatible.java
 * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/collection/List.java
 * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/collection/AbstractQueue.java
 * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/collection/PriorityQueue.java
 * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/Tuple6.java
 * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/collection/Stream.java
 * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/collection/SortedSet.java
 * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/MatchError.java
 * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/collection/Vector.java
 * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/Function0.java
 * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/Tuple.java
 * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/collection/Seq.java
 * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/collection/LinkedHashMap.java
 * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/Tuple0.java
 * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/collection/LinearSeq.java
 * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/Function7.java
 * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/API.java
 * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/collection/Ordered.java
 * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/collection/Foldable.java
 * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/Lambda.java
 * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/concurrent/FutureImpl.java
 * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/collection/LinkedHashMultimap.java
 * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/collection/RedBlackTree.java
 * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/CheckedFunction3.java

- * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/collection/AbstractMultimap.java
- * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/collection/Comparators.java
- * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/NotImplementedError.java
- * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/concurrent/Future.java
- * /opt/cola/permits/1274705121_1645234458.01/0/vavr-0-9-2-sources-jar/io/vavr/control/GwtIncompatible.java

1.250 jakarta-annotations-api 2.1.1

1.250.1 Available under license :

Apache License

Version 2.0, January 2004

<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate

as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify

the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include

the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.251 fabric8-::-kubernetes-model-::-metrics

4.13.3

1.251.1 Available under license :

No license file was found, but licenses were detected in source scan.

Copyright (C) 2015 Red Hat, Inc.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE>

2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Found in path(s):

* /opt/cola/permits/1288519885_1647861811.17/0/kubernetes-model-metrics-4-13-3-jar/META-INF/maven/io.fabric8/kubernetes-model-metrics/pom.xml

No license file was found, but licenses were detected in source scan.

*
 * Copyright (C) 2015 Red Hat, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * <http://www.apache.org/licenses/LICENSE-2.0>
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 *

Found in path(s):

* /opt/cola/permits/1288519885_1647861811.17/0/kubernetes-model-metrics-4-13-3-jar/manifest.vm

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0

Bnd-LastModified: 1619068629071

Build-Jdk-Spec: 1.8

Bundle-Description: Java client for Kubernetes and OpenShift

Bundle-DocURL: <http://redhat.com>

Bundle-License: <http://www.apache.org/licenses/LICENSE-2.0.txt>

Bundle-ManifestVersion: 2

Bundle-Name: Fabric8 :: Kubernetes Model :: Metrics

Bundle-SymbolicName: io.fabric8.kubernetes-model-metrics

Bundle-Vendor: Red Hat

Bundle-Version: 4.13.3

Created-By: Apache Maven Bundle Plugin

Export-Package: io.fabric8.kubernetes.api.model.metrics.v1beta1;uses:=
 "com.fasterxml.jackson.annotation,com.fasterxml.jackson.databind,com.
 fasterxml.jackson.databind.annotation,io.fabric8.kubernetes.api.build
 er,io.fabric8.kubernetes.api.model,io.fabric8.kubernetes.model.annota
 tion";version="4.13.3"

Implementation-Title: Fabric8 :: Kubernetes Model :: Metrics

Implementation-Vendor: Red Hat

Implementation-Version: 4.13.3

Import-Package: com.fasterxml.jackson.annotation;version="[2.11,3)",co
 m.fasterxml.jackson.databind;version="[2.11,3)",com.fasterxml.jackson
 .databind.annotation;version="[2.11,3)",io.fabric8.kubernetes.api.bui
 lder;version="[4.13,5)",io.fabric8.kubernetes.api.model,io.fabric8.ku
 bernetes.model.annotation;version="[4.13,5)"

Require-Capability: osgi.ee;filter="(&(osgi.ee=JavaSE)(version=1.8))"

Specification-Title: Fabric8 :: Kubernetes Model :: Metrics

Specification-Vendor: Red Hat

Specification-Version: 4.13
Tool: Bnd-5.1.1.202006162103

Found in path(s):
* /opt/cola/permits/1288519885_1647861811.17/0/kubernetes-model-metrics-4-13-3-jar/META-INF/MANIFEST.MF

1.252 fabric8-::-kubernetes-model-::- autoscaling 4.13.3

1.252.1 Available under license :

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0
Bnd-LastModified: 1619068504373
Build-Jdk-Spec: 1.8
Bundle-Description: Java client for Kubernetes and OpenShift
Bundle-DocURL: <http://redhat.com>
Bundle-License: <http://www.apache.org/licenses/LICENSE-2.0.txt>
Bundle-ManifestVersion: 2
Bundle-Name: Fabric8 :: Kubernetes Model :: Autoscaling
Bundle-SymbolicName: io.fabric8.kubernetes-model-autoscaling
Bundle-Vendor: Red Hat
Bundle-Version: 4.13.3
Created-By: Apache Maven Bundle Plugin
Export-Package: io.fabric8.kubernetes.api.model.autoscaling.v1;uses:="com.fasterxml.jackson.annotation,com.fasterxml.jackson.databind,com.fasterxml.jackson.databind.annotation,io.fabric8.kubernetes.api.builder,io.fabric8.kubernetes.api.model,io.fabric8.kubernetes.model.annotation";version="4.13.3",io.fabric8.kubernetes.api.model.autoscaling.v2beta1;uses:="com.fasterxml.jackson.annotation,com.fasterxml.jackson.databind,com.fasterxml.jackson.databind.annotation,io.fabric8.kubernetes.api.builder,io.fabric8.kubernetes.api.model,io.fabric8.kubernetes.model.annotation";version="4.13.3",io.fabric8.kubernetes.api.model.autoscaling.v2beta2;uses:="com.fasterxml.jackson.annotation,com.fasterxml.xml.jackson.databind,com.fasterxml.xml.jackson.databind.annotation,io.fabric8.kubernetes.api.builder,io.fabric8.kubernetes.api.model,io.fabric8.kubernetes.model.annotation";version="4.13.3"
Implementation-Title: Fabric8 :: Kubernetes Model :: Autoscaling
Implementation-Vendor: Red Hat
Implementation-Version: 4.13.3
Import-Package: com.fasterxml.jackson.annotation;version="[2.11,3)",com.fasterxml.jackson.databind;version="[2.11,3)",com.fasterxml.jackson.databind.annotation;version="[2.11,3)",io.fabric8.kubernetes.api.builder;version="[4.13,5)",io.fabric8.kubernetes.api.model,io.fabric8.kubernetes.model.annotation;version="[4.13,5)"

Require-Capability: osgi.ee;filter:="(&(osgi.ee=JavaSE)(version=1.8))"

Specification-Title: Fabric8 :: Kubernetes Model :: Autoscaling

Specification-Vendor: Red Hat

Specification-Version: 4.13

Tool: Bnd-5.1.1.202006162103

Found in path(s):

* /opt/cola/permits/1288520050_1647861661.06/0/kubernetes-model-autoscaling-4-13-3-jar/META-INF/MANIFEST.MF

No license file was found, but licenses were detected in source scan.

*

* Copyright (C) 2015 Red Hat, Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*

Found in path(s):

* /opt/cola/permits/1288520050_1647861661.06/0/kubernetes-model-autoscaling-4-13-3-jar/manifest.vm

No license file was found, but licenses were detected in source scan.

Copyright (C) 2015 Red Hat, Inc.

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE>

2.0

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

Found in path(s):

* /opt/cola/permits/1288520050_1647861661.06/0/kubernetes-model-autoscaling-4-13-3-jar/META-

1.253 feign-slf4j 8.18.0

1.253.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2013 Netflix, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1274703681_1645234024.72/0/feign-slf4j-8-18-0-sources-jar/feign/slf4j/Slf4jLogger.java
```

1.254 jakarta-expression-language-3.0-implementation 5.0.0-M1

1.254.1 Available under license :

Notices for Jakarta Expression Language

This content is produced and maintained by the Jakarta Expression Language project.

* Project home: <https://projects.eclipse.org/projects/ee4j.el>

Trademarks

Jakarta Expression Language is a trademark of the Eclipse Foundation.

Copyright

All content is the property of the respective authors or their employers. For more information regarding authorship of content, please consult the listed

source code repository logs.

Declared Project Licenses

This program and the accompanying materials are made available under the terms of the Eclipse Public License v. 2.0 which is available at <http://www.eclipse.org/legal/epl-2.0>. This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License v. 2.0 are satisfied: GNU General Public License, version 2 with the GNU Classpath Exception which is available at <https://www.gnu.org/software/classpath/license.html>.

SPDX-License-Identifier: EPL-2.0 OR GPL-2.0 WITH Classpath-exception-2.0

Source Code

The project maintains the following source code repositories:

* <https://github.com/eclipse-ee4j/el-ri>

Third-party Content

Cryptography

Content may contain encryption software. The country in which you are currently may have restrictions on the import, possession, and use, and/or re-export to another country, of encryption software. BEFORE using any encryption software, please check the country's laws, regulations and policies concerning the import, possession, or use, and re-export of encryption software, to see if this is permitted.

Eclipse Public License - v 2.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial content Distributed under this Agreement, and

b) in the case of each subsequent Contributor:

- i) changes to the Program, and
- ii) additions to the Program;

where such changes and/or additions to the Program originate from and are Distributed by that particular Contributor. A Contribution

"originates" from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include changes or additions to the Program that are not Modified Works.

"Contributor" means any person or entity that Distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions Distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

"Derivative Works" shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

"Modified Works" shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

"Distribute" means the acts of a) distributing or b) making available in any manner that enables the transfer of a copy.

"Source Code" means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Secondary License" means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display,

publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

e) Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

3. REQUIREMENTS

3.1 If a Contributor Distributes the Program in any form, then:

a) the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily

used for software exchange; and

b) the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:

i) effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and

iv) requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.

3.2 When the Program is Distributed as Source Code:

a) it must be made available under this Agreement, or if the Program (i) is combined with other material in a separate file or files made available under a Secondary License, and (ii) the initial Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available under the terms of such Secondary Licenses, and

b) a copy of this Agreement must be included with each copy of the Program.

3.3 Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability ("notices") contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor

("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)

ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended

to be enforceable by any entity that is not a Contributor or Recipient.
No third-party beneficiary rights are created under this Agreement.

Exhibit A - Form of Secondary Licenses Notice

"This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: {name license(s), version(s), and exceptions or additional permissions here}."

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor
Boston, MA 02110-1335
USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new

free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program

(independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest

your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program

except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to

the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN

WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1335 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands ``show w'` and ``show c'` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than ``show w'` and ``show c'`; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program ``Gnomovision'` (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

CLASSPATH EXCEPTION

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

1.255 kafka-schema-registry-client 5.5.5

1.255.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
<project xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://maven.apache.org/POM/4.0.0"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">

  <modelVersion>4.0.0</modelVersion>

  <parent>
    <groupId>io.confluent</groupId>
    <artifactId>kafka-schema-registry-parent</artifactId>
    <version>5.5.5</version>
  </parent>

  <licenses>
    <license>
      <name>Apache License 2.0</name>
      <url>http://www.apache.org/licenses/LICENSE-2.0.html</url>
      <distribution>repo</distribution>
    </license>
  </licenses>

  <artifactId>kafka-schema-registry-client</artifactId>
  <packaging>jar</packaging>
  <name>kafka-schema-registry-client</name>

  <dependencies>
    <dependency>
      <groupId>org.apache.kafka</groupId>
      <artifactId>kafka-clients</artifactId>
    </dependency>
    <dependency>
      <groupId>io.confluent</groupId>
      <artifactId>common-config</artifactId>
    </dependency>

    <dependency>
      <groupId>org.apache.avro</groupId>
      <artifactId>avro</artifactId>
    </dependency>
    <dependency>
      <groupId>com.fasterxml.jackson.core</groupId>
      <artifactId>jackson-databind</artifactId>
    </dependency>
    <dependency>
      <groupId>jakarta.ws.rs</groupId>
      <artifactId>jakarta.ws.rs-api</artifactId>
```

```
</dependency>
<dependency>
  <groupId>org.glassfish.jersey.core</groupId>
  <artifactId>jersey-common</artifactId>
  <version>${jersey.version}</version>
</dependency>
<dependency>
  <groupId>org.easymock</groupId>
  <artifactId>easymock</artifactId>
  <scope>test</scope>
</dependency>
<dependency>
  <groupId>org.powermock</groupId>
  <artifactId>powermock-module-junit4</artifactId>
  <scope>test</scope>
</dependency>
<dependency>
  <groupId>org.powermock</groupId>
  <artifactId>powermock-api-easymock</artifactId>
  <scope>test</scope>
</dependency>
<dependency>
  <groupId>junit</groupId>
  <artifactId>junit</artifactId>
  <scope>test</scope>
</dependency>
<dependency>
  <groupId>io.swagger</groupId>
  <artifactId>swagger-annotations</artifactId>
</dependency>
<dependency>
  <groupId>com.google.guava</groupId>
  <artifactId>guava</artifactId>
  <scope>test</scope>
</dependency>
</dependencies>
</project>
```

Found in path(s):

* /opt/cola/permits/1341639824_1654785850.9168322/0/kafka-schema-registry-client-5-5-5-jar/META-INF/maven/io.confluent/kafka-schema-registry-client/pom.xml

1.256 micrometer-registry-prometheus 1.9.4

1.256.1 Available under license :

Micrometer

Copyright (c) 2017-Present VMware, Inc. All Rights Reserved.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<https://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

This product contains a modified portion of 'io.netty.util.internal.logging',
in the Netty/Common library distributed by The Netty Project:

- * Copyright 2013 The Netty Project
- * License: Apache License v2.0
- * Homepage: <https://netty.io>

This product contains a modified portion of 'StringUtils.isBlank()',
in the Commons Lang library distributed by The Apache Software Foundation:

- * Copyright 2001-2019 The Apache Software Foundation
- * License: Apache License v2.0
- * Homepage: <https://commons.apache.org/proper/commons-lang/>

This product contains a modified portion of 'JsonUtf8Writer',
in the Moshi library distributed by Square, Inc:

- * Copyright 2010 Google Inc.
- * License: Apache License v2.0
- * Homepage: <https://github.com/square/moshi>

This product contains a modified portion of the 'org.springframework.lang'
package in the Spring Framework library, distributed by VMware, Inc:

- * Copyright 2002-2019 the original author or authors.
- * License: Apache License v2.0
- * Homepage: <https://spring.io/projects/spring-framework>

Apache License
Version 2.0, January 2004

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally

submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or

implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<https://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.257 io-swagger-swagger-annotations 1.6.2

1.257.1 Available under license :

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0
Bnd-LastModified: 1593602041104
Build-Jdk: 1.8.0_151
Built-By: frantuma
Bundle-Description: Sonatype helps open source projects to set up Maven repositories on <https://oss.sonatype.org/>
Bundle-License: <http://www.apache.org/licenses/LICENSE-2.0.html>
Bundle-ManifestVersion: 2
Bundle-Name: swagger-annotations
Bundle-SymbolicName: io.swagger.annotations
Bundle-Version: 1.6.2
Created-By: Apache Maven Bundle Plugin
Export-Package: io.swagger.annotations;version="1.6.2"
implementation-version: 1.6.2
mode: development
package: io.swagger
Require-Capability: osgi.ee;filter="(&(osgi.ee=JavaSE)(version=1.7))"
Tool: Bnd-3.3.0.201609221906
url: <https://github.com/swagger-api/swagger-core/modules/swagger-annotations>

Found in path(s):

* /opt/cola/permits/1170799104_1654737045.2729578/0/swagger-annotations-1-6-2-jar/META-INF/MANIFEST.MF

1.258 curator-framework 5.2.1

1.258.1 Available under license :

Curator Framework
Copyright 2011-2022 The Apache Software Foundation

This product includes software developed at

The Apache Software Foundation (<http://www.apache.org/>).

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or

Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.259 kotlinx-coroutines-bom 1.5.1

1.259.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but

not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their

Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with

the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 2000-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

The MIT License (MIT)

Copyright (c) 2016 Parker Moore

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=====
== NOTICE file corresponding to the section 4 d of ==
== the Apache License, Version 2.0, ==
== in this case for the kotlinx.coroutines library. ==
=====

kotlinx.coroutines library.

Copyright 2016-2019 JetBrains s.r.o and respective authors and developers

1.260 micronaut-tracing 4.4.0

1.260.1 Available under license :

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or

Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work

or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work

by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.261 jakarta-ws-rs-api 2.1.6

1.261.1 Available under license :

Notices for the Jakarta RESTful Web Services Project

This content is produced and maintained by the ****Jakarta RESTful Web Services**** project.

* Project home: <https://projects.eclipse.org/projects/ee4j.jaxrs>

Trademarks

****Jakarta RESTful Web Services**** is a trademark of the Eclipse Foundation.

Copyright

All content is the property of the respective authors or their employers. For more information regarding authorship of content, please consult the listed source code repository logs.

Declared Project Licenses

This program and the accompanying materials are made available under the terms of the Eclipse Public License v. 2.0 which is available at

<http://www.eclipse.org/legal/epl-2.0>. This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License v. 2.0 are satisfied: GNU General Public License, version 2 with the GNU Classpath Exception which is available at <https://www.gnu.org/software/classpath/license.html>.

SPDX-License-Identifier: EPL-2.0 OR GPL-2.0 WITH Classpath-exception-2.0

Source Code

The project maintains the following source code repositories:

* <https://github.com/eclipse-ee4j/jaxrs-api>

Third-party Content

This project leverages the following third party content.

javaee-api (7.0)

* License: Apache-2.0 AND W3C

JUnit (4.11)

* License: Common Public License 1.0

Mockito (2.16.0)

* Project: <http://site.mockito.org>

* Source: <https://github.com/mockito/mockito/releases/tag/v2.16.0>

Cryptography

Content may contain encryption software. The country in which you are currently may have restrictions on the import, possession, and use, and/or re-export to another country, of encryption software. BEFORE using any encryption software, please check the country's laws, regulations and policies concerning the import, possession, or use, and re-export of encryption software, to see if this is permitted.

Eclipse Public License - v 2.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial content

Distributed under this Agreement, and

b) in the case of each subsequent Contributor:

i) changes to the Program, and

ii) additions to the Program;

where such changes and/or additions to the Program originate from and are Distributed by that particular Contributor. A Contribution "originates" from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include changes or additions to the Program that are not Modified Works.

"Contributor" means any person or entity that Distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions Distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

"Derivative Works" shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

"Modified Works" shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

"Distribute" means the acts of a) distributing or b) making available in any manner that enables the transfer of a copy.

"Source Code" means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Secondary License" means either the GNU General Public License,

Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

e) Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

3. REQUIREMENTS

3.1 If a Contributor Distributes the Program in any form, then:

a) the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and

b) the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:

i) effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and

iv) requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.

3.2 When the Program is Distributed as Source Code:

a) it must be made available under this Agreement, or if the Program (i) is combined with other material in a separate file or files made available under a Secondary License, and (ii) the initial Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available under the terms of such Secondary Licenses, and

b) a copy of this Agreement must be included with each copy of the Program.

3.3 Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability ("notices") contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was

received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.

Exhibit A - Form of Secondary Licenses Notice

"This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: {name license(s), version(s), and exceptions or additional permissions here}."

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor
Boston, MA 02110-1335
USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and

to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is

to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and

can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the

executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other

circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW.

EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1335 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type
`show w'. This is free software, and you are welcome to redistribute
it under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the
program `Gnomovision' (which makes passes at compilers) written by
James Hacker.
```

```
signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

CLASSPATH EXCEPTION

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or

based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

1.262 jetty-::-jakarta-servlet-api-and-schemas-for-jpms-and-osgi 5.0.2

1.262.1 Available under license :

Notices for Eclipse Project for Servlet

This content is produced and maintained by the Eclipse Project for Servlet project.

* Project home: <https://projects.eclipse.org/projects/ee4j.servlet>

Trademarks

Eclipse Project for Servlet is a trademark of the Eclipse Foundation.

Copyright

All content is the property of the respective authors or their employers. For more information regarding authorship of content, please consult the listed source code repository logs.

Declared Project Licenses

This program and the accompanying materials are made available under the terms of the Eclipse Public License v. 2.0 which is available at <http://www.eclipse.org/legal/epl-2.0>. This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License v. 2.0 are satisfied: GNU General Public License, version 2 with the GNU Classpath Exception which is available at <https://www.gnu.org/software/classpath/license.html>.

SPDX-License-Identifier: EPL-2.0 OR GPL-2.0 WITH Classpath-exception-2.0

Source Code

The project maintains the following source code repositories:

- * <https://github.com/eclipse-ee4j/servlet-api>
- * <https://github.com/eclipse/jetty.toolchain>

Third-party Content

Jakarta

The following artifacts are EPL 2.0 + GPLv2 with classpath exception.
<https://projects.eclipse.org/projects/ee4j.servlet>

- * jakarta.servlet:jakarta.servlet-api

GlassFish

The following artifacts are CDDL + GPLv2 with classpath exception.
<https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html>

- * org.eclipse.jetty.toolchain:jetty-schemas

This program and the accompanying materials are made available under the terms of the Eclipse Public License 2.0 which is available at <http://www.eclipse.org/legal/epl-2.0>, or the Apache Software License 2.0 which is available at <https://www.apache.org/licenses/LICENSE-2.0>.

Eclipse Public License - v 2.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

- a) in the case of the initial Contributor, the initial content Distributed under this Agreement, and

- b) in the case of each subsequent Contributor:

- i) changes to the Program, and
- ii) additions to the Program;

where such changes and/or additions to the Program originate from and are Distributed by that particular Contributor. A Contribution "originates" from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include changes or additions to the Program that are not Modified Works.

"Contributor" means any person or entity that Distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions Distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

"Derivative Works" shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

"Modified Works" shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

"Distribute" means the acts of a) distributing or b) making available in any manner that enables the transfer of a copy.

"Source Code" means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Secondary License" means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent

license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

e) Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

3. REQUIREMENTS

3.1 If a Contributor Distributes the Program in any form, then:

a) the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and

b) the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:

i) effectively disclaims on behalf of all other Contributors all

warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and

iv) requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.

3.2 When the Program is Distributed as Source Code:

a) it must be made available under this Agreement, or if the Program (i) is combined with other material in a separate file or files made available under a Secondary License, and (ii) the initial Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available under the terms of such Secondary Licenses, and

b) a copy of this Agreement must be included with each copy of the Program.

3.3 Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability ("notices") contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such

Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.

Exhibit A - Form of Secondary Licenses Notice

"This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: {name license(s), version(s), and exceptions or additional permissions here}."

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Apache License - v 2.0

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical

transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable

by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use,

reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor
Boston, MA 02110-1335
USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term

"modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you

distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access

to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any

patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED

WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1335 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this

when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type
`show w'. This is free software, and you are welcome to redistribute
it under certain conditions; type `show c' for details.
```

The hypothetical commands ``show w'` and ``show c'` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than ``show w'` and ``show c'`; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the
program `Gnomovision' (which makes passes at compilers) written by
James Hacker.
```

```
signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

CLASSPATH EXCEPTION

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

1.263 jackson-bom 2.13.4

1.263.1 Available under license :

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0

Bundle-License: <http://www.apache.org/licenses/LICENSE-2.0.txt>

Bundle-SymbolicName: com.fasterxml.jackson.dataformat.jackson-dataformat-cbor

Bnd-LastModified: 1662244895850

Implementation-Vendor-Id: com.fasterxml.jackson.dataformat

Specification-Title: Jackson dataformat: CBOR

Bundle-DocURL: <http://github.com/FasterXML/jackson-dataformats-binary>

Import-Package: com.fasterxml.jackson.core;version="[2.13,3)",com.fasterxml.jackson.core.base;version="[2.13,3)",com.fasterxml.jackson.core.format;version="[2.13,3)",com.fasterxml.jackson.core.io;version="[2.13,3)",com.fasterxml.jackson.core.json;version="[2.13,3)",com.fasterxml.jackson.core.sym;version="[2.13,3)",com.fasterxml.jackson.core.util;version="[2.13,3)",com.fasterxml.jackson.databind;version="[2.13,3)",com.fasterxml.jackson.databind.cfg;version="[2.13,3)",com.fasterxml.jackson.dataformat.cbor;version="[2.13,3)"

Require-Capability: osgi.ee;filter="(&(osgi.ee=JavaSE)(version=1.8))"

Implementation-Build-Date: 2022-09-03 22:41:29+0000

Export-Package: com.fasterxml.jackson.dataformat.cbor;uses="com.fasterxml.jackson.core,com.fasterxml.jackson.core.base,com.fasterxml.jackson.core.format,com.fasterxml.jackson.core.io,com.fasterxml.jackson.core.json,com.fasterxml.jackson.core.sym,com.fasterxml.jackson.core.util";version="2.13.4",com.fasterxml.jackson.databind;uses="com.fasterxml.jackson.core,com.fasterxml.jackson.databind,com.fasterxml.jackson.databind.cfg,com.fasterxml.jackson.dataformat.cbor";version="2.13.4"

Bundle-Name: Jackson dataformat: CBOR

Multi-Release: true

Build-Jdk-Spec: 1.8

Bundle-Description: Support for reading and writing Concise Binary Object Representation([CBOR](<https://www.rfc-editor.org/info/rfc7049>)) encoded data using Jackson abstractions (streaming API, data binding, tree model)

Implementation-Title: Jackson dataformat: CBOR

Implementation-Version: 2.13.4

Bundle-ManifestVersion: 2

Specification-Vendor: FasterXML

Bundle-Vendor: FasterXML

Tool: Bnd-5.1.1.202006162103

Implementation-Vendor: FasterXML

Bundle-Version: 2.13.4

X-Compile-Target-JDK: 1.8
X-Compile-Source-JDK: 1.8
Created-By: Apache Maven Bundle Plugin
Specification-Version: 2.13.4

Found in path(s):

* /opt/cola/permits/1424168264_1663881850.9813635/0/jackson-dataformat-cbor-2-13-4-jar/META-INF/MANIFEST.MF

1.264 apache-zookeeper-jute 3.8.0

1.264.1 Available under license :

This program and the accompanying materials are made available under the terms of the Eclipse Public License 2.0 which is available at <http://www.eclipse.org/legal/epl-2.0>, or the Apache Software License 2.0 which is available at <https://www.apache.org/licenses/LICENSE-2.0>.

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

- a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
- b) in the case of each subsequent Contributor:
 - i) changes to the Program, and
 - ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when

combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

- a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.
- b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.
- c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.
- d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

- a) it complies with the terms and conditions of this Agreement; and
- b) its license agreement:
 - i) effectively disclaims on behalf of all Contributors all warranties

- and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
- ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
 - iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
 - iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

- a) it must be made available under this Agreement; and
 - b) a copy of this Agreement must be included with each copy of the Program.
- Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must:

- a) promptly notify the Commercial Contributor in writing of such claim, and
- b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product

offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to

comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity

on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one

of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a

result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of

the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works

that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A

PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 1999-2005 The Apache Software Foundation

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Copyright (c) 2002, 2004, Christopher Clark

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the original author; nor the names of any contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Net::ZooKeeper - Perl extension for Apache ZooKeeper

Copyright 2009 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to

communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of

the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 2010-2012 Coda Hale and Yammer, Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

This contrib module includes software developed under the COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

This contrib depends on binary only jar libraries developed at:

<https://jersey.dev.java.net/>

<https://grizzly.dev.java.net/>

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate

as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify

the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include

the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

This distribution bundles javacc, which is available under the 3-clause BSD License. For details, see a copy of the license in lib/javacc.LICENSE.txt

This distribution bundles jline 2.14.6, which is available under the 2-clause BSD License. For details, see a copy of the license in lib/jline-2.14.6.LICENSE.txt

This distribution bundles SLF4J 1.7.30, which is available under the MIT License. For details, see a copy of the license in lib/slf4j-1.7.30.LICENSE.txt

Apache ZooKeeper
Copyright 2009-2011 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

include/winstdint.h is included only for Windows Client support, as follows:

```
// ISO C9x compliant stdint.h for Microsoft Visual Studio
// Based on ISO/IEC 9899:TC2 Committee draft (May 6, 2005) WG14/N1124
//
// Copyright (c) 2006-2008 Alexander Chemeris
//
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are met:
```

```
//
// 1. Redistributions of source code must retain the above copyright notice,
//    this list of conditions and the following disclaimer.
//
// 2. Redistributions in binary form must reproduce the above copyright
//    notice, this list of conditions and the following disclaimer in the
//    documentation and/or other materials provided with the distribution.
//
// 3. The name of the author may be used to endorse or promote products
//    derived from this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED
// WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
// MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO
// EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
// SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
// PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS;
// OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
// WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
// OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF
// ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
//
//
//////////////////////////////////////////////////////////////////
```

src/java/com/nitido/utills/toaster/Toaster.java:

This java file is copyright by Daniele Piras ("danielepiras80", no email known) released under the Apache Software License 2.0

It has been downloaded in december 2009 from the CVS web interface of the sourceforge project <http://sourceforge.net/projects/jtoaster/> . The web interface to CVS is not available anymore on sourceforge.

The icons in src/main/resources/icons are taken from the Tango project downloaded from <http://tango.freedesktop.org/releases> on 2011-09-06.

The Tango project is public domain.

Distribution packagers should not include the icons in the package but rather depend on tango-icon-theme (Debian package name). ZooInspector will then try to get the icons from /usr/share/icons/Tango rather than from its jar file.

Apache ZooKeeper

Copyright 2009-2022 The Apache Software Foundation

This product includes software developed at

The Apache Software Foundation (<http://www.apache.org/>).

This product includes software components originally developed for Airlift (<https://github.com/airlift/airlift>), licensed under the Apache 2.0 license. The licensing terms for Airlift code can be found at:

<https://github.com/airlift/airlift/blob/master/LICENSE>

This project also includes some files with the following licenses.

These BSD licensed files:

`./zookeeper-client/zookeeper-client-c/src/hashtable/hashtable.c`
`./zookeeper-client/zookeeper-client-c/src/hashtable/hashtable.h`
`./zookeeper-client/zookeeper-client-c/src/hashtable/hashtable_itr.c`
`./zookeeper-client/zookeeper-client-c/src/hashtable/hashtable_itr.h`
`./zookeeper-client/zookeeper-client-c/src/hashtable/hashtable_private.h`
`./zookeeper-contrib/zookeeper-contrib-loggraph/src/main/resources/webapp/org/apache/zookeeper/graph/resources/yui-min.js`
`./zookeeper-docs/src/main/resources/markdown/skin/prototype.js`

These MIT licensed files:

`./zookeeper-contrib/zookeeper-contrib-loggraph/src/main/resources/webapp/org/apache/zookeeper/graph/resources/date.format.js`
`./zookeeper-contrib/zookeeper-contrib-loggraph/src/main/resources/webapp/org/apache/zookeeper/graph/resources/g.bar.js`
`./zookeeper-contrib/zookeeper-contrib-loggraph/src/main/resources/webapp/org/apache/zookeeper/graph/resources/g.dot.js`
`./zookeeper-contrib/zookeeper-contrib-loggraph/src/main/resources/webapp/org/apache/zookeeper/graph/resources/g.line.js`
`./zookeeper-contrib/zookeeper-contrib-loggraph/src/main/resources/webapp/org/apache/zookeeper/graph/resources/g.pie.js`
`./zookeeper-contrib/zookeeper-contrib-loggraph/src/main/resources/webapp/org/apache/zookeeper/graph/resources/g.raphael.js`
`./zookeeper-contrib/zookeeper-contrib-loggraph/src/main/resources/webapp/org/apache/zookeeper/graph/resources/raphael.js`

This Apache 2.0 licensed file:

`./zookeeper-contrib/zookeeper-contrib-zooinpector/src/main/java/com/nitido/utils/toaster/Toaster.java`
Apache ZooKeeper
Copyright 2009-2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

This product includes software components originally
developed for Airlift (<https://github.com/airlift/airlift>),
licensed under the Apache 2.0 license. The licensing terms
for Airlift code can be found at:
<https://github.com/airlift/airlift/blob/master/LICENSE>

This product includes software developed by
The Netty Project (<http://netty.io/>)
Copyright 2011 The Netty Project

The Netty NOTICE file contains the following items:

This product contains the extensions to Java Collections Framework which has been derived from the works by JSR-166 EG, Doug Lea, and Jason T. Greene:

* LICENSE:

* license/LICENSE.jsr166y.txt (Public Domain)

* HOMEPAGE:

* <http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/>

* <http://viewvc.jboss.org/cgi-bin/viewvc.cgi/jboss/cache/experimental/jsr166/>

This product contains a modified version of Robert Harder's Public Domain Base64 Encoder and Decoder, which can be obtained at:

* LICENSE:

* license/LICENSE.base64.txt (Public Domain)

* HOMEPAGE:

* <http://iharder.sourceforge.net/current/java/base64/>

This product contains a modified version of 'JZlib', a re-implementation of zlib in pure Java, which can be obtained at:

* LICENSE:

* license/LICENSE.jzlib.txt (BSD Style License)

* HOMEPAGE:

* <http://www.jcraft.com/jzlib/>

This product contains a modified version of 'Webbit', a Java event based WebSocket and HTTP server:

* LICENSE:

* license/LICENSE.webbit.txt (BSD License)

* HOMEPAGE:

* <https://github.com/joewalnes/webbit>

This product optionally depends on 'Protocol Buffers', Google's data interchange format, which can be obtained at:

* LICENSE:

* license/LICENSE.protobuf.txt (New BSD License)

* HOMEPAGE:

* <http://code.google.com/p/protobuf/>

This product optionally depends on 'Bouncy Castle Crypto APIs' to generate a temporary self-signed X.509 certificate when the JVM does not provide the equivalent functionality. It can be obtained at:

* LICENSE:

* license/LICENSE.bouncycastle.txt (MIT License)

* HOMEPAGE:

* <http://www.bouncycastle.org/>

This product optionally depends on 'SLF4J', a simple logging facade for Java, which can be obtained at:

* LICENSE:

* [license/LICENSE.slf4j.txt](#) (MIT License)

* HOMEPAGE:

* <http://www.slf4j.org/>

This product optionally depends on 'Apache Commons Logging', a logging framework, which can be obtained at:

* LICENSE:

* [license/LICENSE.commons-logging.txt](#) (Apache License 2.0)

* HOMEPAGE:

* <http://commons.apache.org/logging/>

This product optionally depends on 'Apache Logback', a logging framework, which can be obtained at:

* LICENSE:

* [license/LICENSE.logback.txt](#) (Eclipse Public License 1.0)

* HOMEPAGE:

* <https://logback.qos.ch/>

This product optionally depends on 'JBoss Logging', a logging framework, which can be obtained at:

* LICENSE:

* [license/LICENSE.jboss-logging.txt](#) (GNU LGPL 2.1)

* HOMEPAGE:

* <http://anonsvn.jboss.org/repos/common/common-logging-spi/>

This product optionally depends on 'Apache Felix', an open source OSGi framework implementation, which can be obtained at:

* LICENSE:

* [license/LICENSE.felix.txt](#) (Apache License 2.0)

* HOMEPAGE:

* <http://felix.apache.org/>

The bundled library Metrics Core NOTICE file reports the following items

Metrics

Copyright 2010-2013 Coda Hale and Yammer, Inc.

This product includes software developed by Coda Hale and Yammer, Inc.

This product includes code derived from the JSR-166 project (ThreadLocalRandom, Striped64, LongAdder), which was released with the following comments:

Written by Doug Lea with assistance from members of JCP JSR-166 Expert Group and released to the public domain, as explained at <http://creativecommons.org/publicdomain/zero/1.0/>

The Nappy Java NOTICE file reports the following items:

This product includes software developed by Google
Snappy: <http://code.google.com/p/snappy/> (New BSD License)

This product includes software developed by Apache
PureJavaCrc32C from apache-hadoop-common <http://hadoop.apache.org/>
(Apache 2.0 license)

This library contained statically linked libstdc++. This inclusion is allowed by "GCC Runtime Library Exception"
<http://gcc.gnu.org/onlinedocs/libstdc++/manual/license.html>

== Contributors ==

- * Tatu Saloranta
 - * Providing benchmark suite
- * Alec Wysoker
 - * Performance and memory usage improvement

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of,

publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution

notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing

the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

```
=====  
=====  
=== The following part contains the dual OpenSSL and SSLeay license ===  
=== for OpenSSL versions 1.1.1, 1.1.0, 1.0.2 and all prior releases ===  
=== (see https://www.openssl.org/source/license.html) ===  
=====  
=====
```

LICENSE ISSUES

=====

The OpenSSL toolkit stays under a double license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. See below for the actual license texts.

OpenSSL License

/* =====

* Copyright (c) 1998-2019 The OpenSSL Project. All rights reserved.

*

* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:

*

* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.

*

* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in
* the documentation and/or other materials provided with the
* distribution.

*

* 3. All advertising materials mentioning features or use of this
* software must display the following acknowledgment:
* "This product includes software developed by the OpenSSL Project
* for use in the OpenSSL Toolkit. (<http://www.openssl.org/>)"

*

* 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
* endorse or promote products derived from this software without
* prior written permission. For written permission, please contact
* openssl-core@openssl.org.

*

* 5. Products derived from this software may not be called "OpenSSL"
* nor may "OpenSSL" appear in their names without prior written
* permission of the OpenSSL Project.

*

* 6. Redistributions of any form whatsoever must retain the following
* acknowledgment:
* "This product includes software developed by the OpenSSL Project
* for use in the OpenSSL Toolkit (<http://www.openssl.org/>)"

*

* THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY
* EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR

* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
* STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
* OF THE POSSIBILITY OF SUCH DAMAGE.

* =====

*

* This product includes cryptographic software written by Eric Young
* (eay@cryptsoft.com). This product includes software written by Tim
* Hudson (tjh@cryptsoft.com).

*

*/

Original SSLeay License

/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)

* All rights reserved.

*

* This package is an SSL implementation written

* by Eric Young (eay@cryptsoft.com).

* The implementation was written so as to conform with Netscapes SSL.

*

* This library is free for commercial and non-commercial use as long as

* the following conditions are aheared to. The following conditions

* apply to all code found in this distribution, be it the RC4, RSA,

* lhash, DES, etc., code; not just the SSL code. The SSL documentation

* included with this distribution is covered by the same copyright terms

* except that the holder is Tim Hudson (tjh@cryptsoft.com).

*

* Copyright remains Eric Young's, and as such any Copyright notices in

* the code are not to be removed.

* If this package is used in a product, Eric Young should be given attribution

* as the author of the parts of the library used.

* This can be in the form of a textual message at program startup or

* in documentation (online or textual) provided with the package.

*

* Redistribution and use in source and binary forms, with or without

* modification, are permitted provided that the following conditions

* are met:

* 1. Redistributions of source code must retain the copyright

* notice, this list of conditions and the following disclaimer.

* 2. Redistributions in binary form must reproduce the above copyright

* notice, this list of conditions and the following disclaimer in the

* documentation and/or other materials provided with the distribution.

```

* 3. All advertising materials mentioning features or use of this software
* must display the following acknowledgement:
* "This product includes cryptographic software written by
* Eric Young (eay@cryptsoft.com)"
* The word 'cryptographic' can be left out if the rouines from the library
* being used are not cryptographic related :-).
* 4. If you include any Windows specific code (or a derivative thereof) from
* the apps directory (application code) you must include an acknowledgement:
* "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
*
* THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS" AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*
* The licence and distribution terms for any publically available version or
* derivative of this code cannot be changed. i.e. this code cannot simply be
* copied and put under another distribution licence
* [including the GNU Public Licence.]
*/

```

```

=====
====
=== The following part contains the license for the Cyrus SASL 2.x library ===
=== used for optional SASL support ===
=====
=====

```

```

/* CMU libsasl
* Tim Martin
* Rob Earhart
* Rob Siemborski
*/
/*
* Copyright (c) 1998-2003 Carnegie Mellon University. All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright

```

- * notice, this list of conditions and the following disclaimer.
- *
- * 2. Redistributions in binary form must reproduce the above copyright
- * notice, this list of conditions and the following disclaimer in
- * the documentation and/or other materials provided with the
- * distribution.
- *
- * 3. The name "Carnegie Mellon University" must not be used to
- * endorse or promote products derived from this software without
- * prior written permission. For permission or any other legal
- * details, please contact
- * Office of Technology Transfer
- * Carnegie Mellon University
- * 5000 Forbes Avenue
- * Pittsburgh, PA 15213-3890
- * (412) 268-4387, fax: (412) 268-7395
- * tech-transfer@andrew.cmu.edu
- *
- * 4. Redistributions of any form whatsoever must retain the following
- * acknowledgment:
- * "This product includes software developed by Computing Services
- * at Carnegie Mellon University (<http://www.cmu.edu/computing/>)."
- *
- * CARNEGIE MELLON UNIVERSITY DISCLAIMS ALL WARRANTIES WITH REGARD TO
- * THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY
- * AND FITNESS, IN NO EVENT SHALL CARNEGIE MELLON UNIVERSITY BE LIABLE
- * FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
- * WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN
- * AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING
- * OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
- */

Copyright (c) 2002-2012, the original author or authors.
All rights reserved.

<http://www.opensource.org/licenses/bsd-license.php>

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of JLine nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2004-2017 QOS.ch

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,

and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the

Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,

whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.265 opentest4j-opentest4j 1.2.0

1.265.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes

of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You

meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor,

except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.266 feign-jax-rs 10.12

1.266.1 Available under license :

No license file was found, but licenses were detected in source scan.

<!--

Copyright 2012-2020 The Feign Authors

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License
is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
or implied. See the License for the specific language governing permissions and limitations under
the License.

-->

Found in path(s):

* /opt/cola/permits/1418559498_1663305821.3174336/0/feign-jaxrs-10-12-jar/META-
INF/maven/io.github.openfeign/feign-jaxrs/pom.xml

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0

Bundle-Description: Feign JAX-RS

Bundle-License: <http://www.apache.org/licenses/LICENSE-2.0.txt>

Bundle-SymbolicName: io.github.openfeign.feign-jaxrs

Built-By: circleci

Bnd-LastModified: 1609190910214
Bundle-ManifestVersion: 2
Bundle-DocURL: <https://github.com/openfeign>
Bundle-Vendor: OpenFeign
Import-Package: feign;version="[10.12,11)",javax.ws.rs;version="[1.1,2)"
Require-Capability: osgi.ee;filter="(&(osgi.ee=JavaSE)(version=1.8))"
Tool: Bnd-4.0.0.201805111645
Export-Package: feign.jaxrs;uses:=feign;version="10.12.0"
Bundle-Name: Feign JAX-RS
Bundle-Version: 10.12.0
Created-By: Apache Maven Bundle Plugin
Build-Jdk: 1.8.0_265

Found in path(s):

* /opt/cola/permits/1418559498_1663305821.3174336/0/feign-jaxrs-10-12-jar/META-INF/MANIFEST.MF

1.267 micronaut-aws 3.9.2

1.267.1 Available under license :

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation

source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable

(except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and

may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify,

defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.268 metrics-core-library 2.2.0

1.268.1 Available under license :

Public Domain

1.269 zjsonpatch 0.3.0

1.269.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
```

```
<groupId>com.flipkart.zjsonpatch</groupId>
```

```
<artifactId>zjsonpatch</artifactId>
<version>0.3.0</version>
<packaging>jar</packaging>

<name>zjsonpatch</name>
<description>Java Library to find / apply JSON Patches according to RFC 6902</description>
<url>https://github.com/flipkart-incubator/zjsonpatch/</url>

<scm>
  <connection>scm:git:https://github.com/flipkart-incubator/zjsonpatch.git</connection>
  <developerConnection>scm:git:https://github.com/flipkart-incubator/zjsonpatch.git</developerConnection>
  <tag>HEAD</tag>
  <url>https://github.com/flipkart-incubator/zjsonpatch/</url>
</scm>

<developers>
  <developer>
    <id>vishwakarma</id>
    <name>Gopi Vishwakarma</name>
    <email>vishwakarma.iita@gmail.com</email>
  </developer>
</developers>

<licenses>
  <license>
    <name>The Apache Software License, Version 2.0</name>
    <url>http://www.apache.org/licenses/LICENSE-2.0.txt</url>
    <distribution>repo</distribution>
  </license>
</licenses>

<properties>
  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  <jackson.version>2.8.5</jackson.version>
</properties>

<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-compiler-plugin</artifactId>
      <version>2.3.1</version>
      <configuration>
        <source>1.6</source>
        <target>1.6</target>
        <showDeprecation>>true</showDeprecation>
        <showWarnings>>true</showWarnings>
      </configuration>
    </plugin>
  </plugins>
</build>
```

```

</plugin>
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-source-plugin</artifactId>
  <version>2.2.1</version>
  <executions>
    <execution>
      <id>attach-sources</id>
      <goals>
        <goal>jar-no-fork</goal>
      </goals>
    </execution>
  </executions>
</plugin>
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-javadoc-plugin</artifactId>
  <version>2.9.1</version>
  <executions>
    <execution>
      <id>attach-javadocs</id>
      <goals>
        <goal>jar</goal>
      </goals>
    </execution>
  </executions>
</plugin>
<plugin>
  <groupId>org.sonatype.plugins</groupId>
  <artifactId>nexus-staging-maven-plugin</artifactId>
  <version>1.6.3</version>
  <extensions>>true</extensions>
  <configuration>
    <serverId>ossrh</serverId>
    <nexusUrl>https://oss.sonatype.org/</nexusUrl>
    <autoReleaseAfterClose>>true</autoReleaseAfterClose>
  </configuration>
</plugin>
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-gpg-plugin</artifactId>
  <version>1.4</version>
  <executions>
    <execution>
      <id>sign-artifacts</id>
      <phase>verify</phase>
      <goals>
        <goal>sign</goal>
      </goals>
    </execution>
  </executions>

```

```

        </goals>
    </execution>
</executions>
</plugin>
<plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-release-plugin</artifactId>
    <version>2.5</version>
    <configuration>
        <autoVersionSubmodules>true</autoVersionSubmodules>
        <useReleaseProfile>>false</useReleaseProfile>
        <releaseProfiles>release</releaseProfiles>
        <goals>deploy</goals>
    </configuration>
</plugin>
</plugins>
</build>
<dependencies>
    <dependency>
        <groupId>com.fasterxml.jackson.core</groupId>
        <artifactId>jackson-databind</artifactId>
        <version>${jackson.version}</version>
    </dependency>
    <dependency>
        <groupId>com.fasterxml.jackson.core</groupId>
        <artifactId>jackson-core</artifactId>
        <version>${jackson.version}</version>
    </dependency>
    <dependency>
        <groupId>com.google.guava</groupId>
        <artifactId>guava</artifactId>
        <version>18.0</version>
    </dependency>
    <dependency>
        <scope>test</scope>
        <groupId>commons-io</groupId>
        <artifactId>commons-io</artifactId>
        <version>2.4</version>
    </dependency>
    <dependency>
        <scope>test</scope>
        <groupId>junit</groupId>
        <artifactId>junit</artifactId>
        <version>4.12</version>
    </dependency>
    <dependency>
        <groupId>org.apache.commons</groupId>
        <artifactId>commons-collections4</artifactId>

```

```

    <version>4.1 </version>
  </dependency>
</dependencies>
<distributionManagement>
  <snapshotRepository>
    <id>ossrh</id>
    <url>https://oss.sonatype.org/content/repositories/snapshots</url>
  </snapshotRepository>
  <repository>
    <id>ossrh</id>
    <url>https://oss.sonatype.org/service/local/staging/deploy/maven2/</url>
  </repository>
</distributionManagement>
</project>

```

Found in path(s):

```

* /opt/cola/permits/1288519755_1647351836.02/0/zjsonpatch-0-3-0-jar/META-
INF/maven/com.flipkart.zjsonpatch/zjsonpatch/pom.xml

```

1.270 netty/tomcatnative-[openssl---classes]

2.0.54.Final

1.270.1 Available under license :

LICENSE ISSUES

=====

The OpenSSL toolkit stays under a double license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. See below for the actual license texts.

OpenSSL License

/* =====

* Copyright (c) 1998-2019 The OpenSSL Project. All rights reserved.

*

* Redistribution and use in source and binary forms, with or without

* modification, are permitted provided that the following conditions

* are met:

*

* 1. Redistributions of source code must retain the above copyright

* notice, this list of conditions and the following disclaimer.

*

* 2. Redistributions in binary form must reproduce the above copyright

* notice, this list of conditions and the following disclaimer in

* the documentation and/or other materials provided with the
 * distribution.
 *

* 3. All advertising materials mentioning features or use of this
 * software must display the following acknowledgment:
 * "This product includes software developed by the OpenSSL Project
 * for use in the OpenSSL Toolkit. (<http://www.openssl.org/>)"
 *

* 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
 * endorse or promote products derived from this software without
 * prior written permission. For written permission, please contact
 * openssl-core@openssl.org.
 *

* 5. Products derived from this software may not be called "OpenSSL"
 * nor may "OpenSSL" appear in their names without prior written
 * permission of the OpenSSL Project.
 *

* 6. Redistributions of any form whatsoever must retain the following
 * acknowledgment:
 * "This product includes software developed by the OpenSSL Project
 * for use in the OpenSSL Toolkit (<http://www.openssl.org/>)"
 *

* THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS" AND ANY
 * EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
 * PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR
 * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
 * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
 * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
 * OF THE POSSIBILITY OF SUCH DAMAGE.
 * =====
 *

* This product includes cryptographic software written by Eric Young
 * (ey@cryptsoft.com). This product includes software written by Tim
 * Hudson (tjh@cryptsoft.com).
 *
 */

Original SSLeay License

/* Copyright (C) 1995-1998 Eric Young (ey@cryptsoft.com)
 * All rights reserved.
 *

* This package is an SSL implementation written
 * by Eric Young (eay@cryptsoft.com).
 * The implementation was written so as to conform with Netscapes SSL.
 *
 * This library is free for commercial and non-commercial use as long as
 * the following conditions are aheared to. The following conditions
 * apply to all code found in this distribution, be it the RC4, RSA,
 * lhash, DES, etc., code; not just the SSL code. The SSL documentation
 * included with this distribution is covered by the same copyright terms
 * except that the holder is Tim Hudson (tjh@cryptsoft.com).
 *
 * Copyright remains Eric Young's, and as such any Copyright notices in
 * the code are not to be removed.
 * If this package is used in a product, Eric Young should be given attribution
 * as the author of the parts of the library used.
 * This can be in the form of a textual message at program startup or
 * in documentation (online or textual) provided with the package.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * 3. All advertising materials mentioning features or use of this software
 * must display the following acknowledgement:
 * "This product includes cryptographic software written by
 * Eric Young (eay@cryptsoft.com)"
 * The word 'cryptographic' can be left out if the rouines from the library
 * being used are not cryptographic related :-).
 * 4. If you include any Windows specific code (or a derivative thereof) from
 * the apps directory (application code) you must include an acknowledgement:
 * "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
 *
 * THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS" AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 *

- * The licence and distribution terms for any publically available version or
- * derivative of this code cannot be changed. i.e. this code cannot simply be
- * copied and put under another distribution licence
- * [including the GNU Public Licence.]
- */

BoringSSL is a fork of OpenSSL. As such, large parts of it fall under OpenSSL licensing. Files that are completely new have a Google copyright and an ISC license. This license is reproduced at the bottom of this file.

Contributors to BoringSSL are required to follow the CLA rules for Chromium:
<https://cla.developers.google.com/cla>

Files in third_party/ have their own licenses, as described therein. The MIT license, for third_party/fiat, which, unlike other third_party directories, is compiled into non-test libraries, is included below.

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. See below for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL please contact openssl-core@openssl.org.

The following are Google-internal bug numbers where explicit permission from some authors is recorded for use of their work. (This is purely for our own record keeping.)

27287199

27287880

27287883

OpenSSL License

/* =====

* Copyright (c) 1998-2011 The OpenSSL Project. All rights reserved.

*

* Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:

*

* 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.

*

* 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in
 * the documentation and/or other materials provided with the
 * distribution.

*

* 3. All advertising materials mentioning features or use of this
 * software must display the following acknowledgment:
 * "This product includes software developed by the OpenSSL Project
 * for use in the OpenSSL Toolkit. (<http://www.openssl.org/>)"

*

* 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
 * endorse or promote products derived from this software without

* prior written permission. For written permission, please contact
 * openssl-core@openssl.org.
 *

* 5. Products derived from this software may not be called "OpenSSL"
 * nor may "OpenSSL" appear in their names without prior written
 * permission of the OpenSSL Project.
 *

* 6. Redistributions of any form whatsoever must retain the following
 * acknowledgment:
 * "This product includes software developed by the OpenSSL Project
 * for use in the OpenSSL Toolkit (<http://www.openssl.org/>)"
 *

* THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS" AND ANY
 * EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
 * PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR
 * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
 * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
 * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
 * OF THE POSSIBILITY OF SUCH DAMAGE.
 * =====
 *

* This product includes cryptographic software written by Eric Young
 * (eay@cryptsoft.com). This product includes software written by Tim
 * Hudson (tjh@cryptsoft.com).
 *

*/
 Original SSLeay License

 /* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
 * All rights reserved.
 *

* This package is an SSL implementation written
 * by Eric Young (eay@cryptsoft.com).
 * The implementation was written so as to conform with Netscapes SSL.
 *

* This library is free for commercial and non-commercial use as long as
 * the following conditions are aheared to. The following conditions
 * apply to all code found in this distribution, be it the RC4, RSA,
 * lhash, DES, etc., code; not just the SSL code. The SSL documentation
 * included with this distribution is covered by the same copyright terms
 * except that the holder is Tim Hudson (tjh@cryptsoft.com).
 *

* Copyright remains Eric Young's, and as such any Copyright notices in

* the code are not to be removed.

* If this package is used in a product, Eric Young should be given attribution

* as the author of the parts of the library used.

* This can be in the form of a textual message at program startup or

* in documentation (online or textual) provided with the package.

*

* Redistribution and use in source and binary forms, with or without

* modification, are permitted provided that the following conditions

* are met:

* 1. Redistributions of source code must retain the copyright

* notice, this list of conditions and the following disclaimer.

* 2. Redistributions in binary form must reproduce the above copyright

* notice, this list of conditions and the following disclaimer in the

* documentation and/or other materials provided with the distribution.

* 3. All advertising materials mentioning features or use of this software

* must display the following acknowledgement:

* "This product includes cryptographic software written by

* Eric Young (eay@cryptsoft.com)"

* The word 'cryptographic' can be left out if the routines from the library

* being used are not cryptographic related :-).

* 4. If you include any Windows specific code (or a derivative thereof) from

* the apps directory (application code) you must include an acknowledgement:

* "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"

*

* THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS" AND

* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE

* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE

* ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE

* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL

* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS

* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)

* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT

* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY

* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF

* SUCH DAMAGE.

*

* The licence and distribution terms for any publically available version or

* derivative of this code cannot be changed. i.e. this code cannot simply be

* copied and put under another distribution licence

* [including the GNU Public Licence.]

*/

ISC license used for completely new code in BoringSSL:

/* Copyright (c) 2015, Google Inc.

*

* Permission to use, copy, modify, and/or distribute this software for any

* purpose with or without fee is hereby granted, provided that the above

* copyright notice and this permission notice appear in all copies.

*

* THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
* WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
* MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY
* SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
* WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION
* OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN
* CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. */

The code in third_party/ fiat carries the MIT license:

Copyright (c) 2015-2016 the fiat-crypto authors (see
<https://github.com/mit-plv/ fiat-crypto/blob/master/AUTHORS>).

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

Licenses for support code

Parts of the TLS test suite are under the Go license. This code is not included
in BoringSSL (i.e. libcrypto and libssl) when compiled, however, so
distributing code linked against BoringSSL does not trigger this license:

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.

* Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,

SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

BoringSSL uses the Chromium test infrastructure to run a continuous build, trybots etc. The scripts which manage this, and the script for generating build metadata, are under the Chromium license. Distributing code linked against BoringSSL does not trigger this license.

Copyright 2015 The Chromium Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity

on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one

of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a

result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

The Netty Project

=====

Please visit the Netty web site for more information:

* <http://netty.io/>

Copyright 2016 The Netty Project

The Netty Project licenses this file to you under the Apache License, version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at:

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

This product contains a forked and modified version of Tomcat Native

- * LICENSE:
 - * [license/LICENSE.tomcat-native.txt](#) (Apache License 2.0)
- * HOMEPAGE:
 - * <http://tomcat.apache.org/native-doc/>
 - * <https://svn.apache.org/repos/asf/tomcat/native/>

This product contains the Maven wrapper scripts from 'Maven Wrapper', that provides an easy way to ensure a user has everything necessary to run the Maven build.

- * LICENSE:
 - * [license/LICENSE.mvn-wrapper.txt](#) (Apache License 2.0)
- * HOMEPAGE:
 - * <https://github.com/takari/maven-wrapper>

This product contains small piece of code to support AIX, taken from netbsd.

- * LICENSE:
 - * [license/LICENSE.aix-netbsd.txt](#) (OpenSSL License)
- * HOMEPAGE:
 - * <https://ftp.netbsd.org/pub/NetBSD/NetBSD-current/src/crypto/external/bsd/openssl/dist>

This product contains code from boringssl.

- * LICENSE (Combination ISC and OpenSSL license)
 - * [license/LICENSE.boringssl.txt](#) (Combination ISC and OpenSSL license)
- * HOMEPAGE:
 - * <https://boringssl.googlesource.com/boringssl/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or

agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.271 fabric8-::-kubernetes-model-::-common

4.13.3

1.271.1 Available under license :

No license file was found, but licenses were detected in source scan.

Copyright (C) 2015 Red Hat, Inc.

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE>

2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Found in path(s):

* /opt/cola/permits/1288520151_1647861705.13/0/kubernetes-model-common-4-13-3-jar/META-INF/maven/io.fabric8/kubernetes-model-common/pom.xml

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0

Bnd-LastModified: 1619068339611

Build-Jdk-Spec: 1.8

Bundle-Description: Java client for Kubernetes and OpenShift

Bundle-DocURL: <http://redhat.com>

Bundle-License: <http://www.apache.org/licenses/LICENSE-2.0.txt>

Bundle-ManifestVersion: 2

Bundle-Name: Fabric8 :: Kubernetes Model :: Common

Bundle-SymbolicName: io.fabric8.kubernetes-model-common

Bundle-Vendor: Red Hat
Bundle-Version: 4.13.3
Created-By: Apache Maven Bundle Plugin
Export-Package: io.fabric8.kubernetes.model.annotation;version="4.13.3",io.fabric8.kubernetes.model.util;version="4.13.3"
Implementation-Title: Fabric8 :: Kubernetes Model :: Common
Implementation-Vendor: Red Hat
Implementation-Version: 4.13.3
Import-Package: io.fabric8.kubernetes.model.annotation
Require-Capability: osgi.ee;filter="(&(osgi.ee=JavaSE)(version=1.8))"
Specification-Title: Fabric8 :: Kubernetes Model :: Common
Specification-Vendor: Red Hat
Specification-Version: 4.13
Tool: Bnd-5.1.1.202006162103

Found in path(s):

* /opt/cola/permits/1288520151_1647861705.13/0/kubernetes-model-common-4-13-3-jar/META-INF/MANIFEST.MF

1.272 metrics-health-checks 4.0.5

1.272.1 Available under license :

Apache-2.0

1.273 micronaut-rxjava-2 1.2.1

1.273.1 Available under license :

Apache License

Version 2.0, January 2004

<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or

otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents

of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.274 openapi/swagger-support 4.0.1

1.274.1 Available under license :

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and

attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the

appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.275 xml-path 5.2.0

1.275.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2019 the original author or authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1473459925_1668478029.9454896/0/xml-path-5-2-0-sources-
jar/io/restassured/internal/path/xml/mapping/XmlPathJakartaEEOBJECTDeserializ
er.groovy
* /opt/cola/permits/1473459925_1668478029.9454896/0/xml-path-5-2-0-sources-
jar/io/restassured/internal/path/xml/XmlRenderer.groovy
* /opt/cola/permits/1473459925_1668478029.9454896/0/xml-path-5-2-0-sources-
jar/io/restassured/internal/path/xml/GroovyNodeSerializer.groovy
* /opt/cola/permits/1473459925_1668478029.9454896/0/xml-path-5-2-0-sources-
jar/io/restassured/path/xml/config/XmlParserType.java
* /opt/cola/permits/1473459925_1668478029.9454896/0/xml-path-5-2-0-sources-
jar/io/restassured/internal/path/xml/XMLAssertion.groovy
* /opt/cola/permits/1473459925_1668478029.9454896/0/xml-path-5-2-0-sources-
jar/io/restassured/path/xml/mapper/factory/JakartaEEOBJECTMapperFactory.java
* /opt/cola/permits/1473459925_1668478029.9454896/0/xml-path-5-2-0-sources-
jar/io/restassured/path/xml/XmlPath.java
* /opt/cola/permits/1473459925_1668478029.9454896/0/xml-path-5-2-0-sources-
jar/io/restassured/internal/path/xml/mapping/XmlPathJaxbOBJECTDeserializ
er.groovy
* /opt/cola/permits/1473459925_1668478029.9454896/0/xml-path-5-2-0-sources-
jar/io/restassured/path/xml/element/NodeChildren.java
* /opt/cola/permits/1473459925_1668478029.9454896/0/xml-path-5-2-0-sources-
jar/io/restassured/path/xml/mapper/factory/JAXBObjectMapperFactory.java
```

* /opt/cola/permits/1473459925_1668478029.9454896/0/xml-path-5-2-0-sources-jar/io/restassured/internal/path/xml/NodeBase.groovy
* /opt/cola/permits/1473459925_1668478029.9454896/0/xml-path-5-2-0-sources-jar/io/restassured/internal/path/xml/mapping/XmlObjectDeserializer.groovy
* /opt/cola/permits/1473459925_1668478029.9454896/0/xml-path-5-2-0-sources-jar/io/restassured/internal/path/xml/XmlPrettifier.groovy
* /opt/cola/permits/1473459925_1668478029.9454896/0/xml-path-5-2-0-sources-jar/io/restassured/path/xml/mapper/factory/DefaultJakartaEEOBJECTMapperFactory.java
* /opt/cola/permits/1473459925_1668478029.9454896/0/xml-path-5-2-0-sources-jar/io/restassured/path/xml/element/PathElement.java
* /opt/cola/permits/1473459925_1668478029.9454896/0/xml-path-5-2-0-sources-jar/io/restassured/path/xml/element/Node.java
* /opt/cola/permits/1473459925_1668478029.9454896/0/xml-path-5-2-0-sources-jar/io/restassured/internal/path/xml/NodeChildrenImpl.java
* /opt/cola/permits/1473459925_1668478029.9454896/0/xml-path-5-2-0-sources-jar/io/restassured/internal/path/xml/NodeImpl.java
* /opt/cola/permits/1473459925_1668478029.9454896/0/xml-path-5-2-0-sources-jar/io/restassured/path/xml/mapper/factory/DefaultJAXBOBJECTMapperFactory.java
* /opt/cola/permits/1473459925_1668478029.9454896/0/xml-path-5-2-0-sources-jar/io/restassured/path/xml/config/XmlPathConfig.java
* /opt/cola/permits/1473459925_1668478029.9454896/0/xml-path-5-2-0-sources-jar/io/restassured/path/xml/exception/XmlPathException.java
* /opt/cola/permits/1473459925_1668478029.9454896/0/xml-path-5-2-0-sources-jar/io/restassured/path/xml/mapping/XmlPathObjectDeserializer.java

1.276 tagsoup 1.2.1

1.276.1 Available under license :

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf

and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.277 netty/transport/classes/epoll

4.1.84.Final

1.277.1 Available under license :

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0

Implementation-Title: Netty/Codec/HTTP

Bundle-Description: Netty is an asynchronous event-driven network application framework for rapid development of maintainable high performance protocol servers and clients.

Automatic-Module-Name: io.netty.codec.http

Bundle-License: <https://www.apache.org/licenses/LICENSE-2.0>

Bundle-SymbolicName: io.netty.codec-http

Implementation-Version: 4.1.84.Final

Built-By: chris

Bnd-LastModified: 1665536154725

Bundle-ManifestVersion: 2

Implementation-Vendor-Id: io.netty

Bundle-DocURL: <https://netty.io/>

Bundle-Vendor: The Netty Project

Import-Package: com.aayushatharva.brotli4j.encoder;resolution:=optional,com.jcraft.jzlib;resolution:=optional,io.netty.buffer;version="[4.1,5)",io.netty.channel;version="[4.1,5)",io.netty.channel.embedded;version="[4.1,5)",io.netty.handler.codec,io.netty.handler.codec.base64;version="[4.1,5)",io.netty.handler.codec.compression;version="[4.1,5)",io.netty.handler.ssl;version="[4.1,5)",io.netty.handler.stream;version="[4.1,5)",io.netty.util;version="[4.1,5)",io.netty.util.concurrent;version="[4.1,5)",io.netty.util.internal;version="[4.1,5)",io.netty.util.internal.logging;version="[4.1,5)",sun.nio.ch;resolution:=optional,org.eclipse.jetty.npn;version="[1,2)";resolution:=optional,org.eclipse.jetty.alpn;version="[1,2)";resolution:=optional

Require-Capability: osgi.ee;filter="(&(osgi.ee=JavaSE)(version=1.6))"

Tool: Bnd-2.4.1.201501161923

Implementation-Vendor: The Netty Project

Export-Package: io.netty.handler.codec.http;uses:="io.netty.buffer,io.netty.channel,io.netty.channel.embedded,io.netty.handler.codec,io.netty.handler.codec.compression,io.netty.handler.codec.http.cookie,io.netty.handler.stream,io.netty.util";version="4.1.84",io.netty.handler.codec.http.cookie;version="4.1.84",io.netty.handler.codec.http.cors;uses:="io.netty.channel,io.netty.handler.codec.http";version="4.1.84",io.netty.handler.codec.http.multipart;uses:="io.netty.buffer,io.netty.channel,io.netty.handler.codec,io.netty.handler.codec.http,io.netty.handler.stream,io.netty.util";version="4.1.84",io.netty.handler.codec.http.websocketx;uses:="io.netty.buffer,io.netty.channel,io.netty.handler.codec,io.netty.handler.codec.http,io.netty.handler.stream,io.net

y.util,io.netty.util.internal.logging";version="4.1.84",io.netty.handler.codec.http.websocketx.extensions;uses:="io.netty.channel,io.netty.handler.codec,io.netty.handler.codec.http.websocketx";version="4.1.84",io.netty.handler.codec.http.websocketx.extensions.compression;uses:="io.netty.channel,io.netty.handler.codec.http.websocketx.extensions";version="4.1.84",io.netty.handler.codec.rtsp;uses:="io.netty.buffer,io.netty.channel,io.netty.handler.codec.http,io.netty.util";version="4.1.84",io.netty.handler.codec.spdy;uses:="io.netty.buffer,io.netty.channel,io.netty.handler.codec,io.netty.handler.codec.http,io.netty.util";version="4.1.84"

Bundle-Name: Netty/Codec/HTTP

Bundle-Version: 4.1.84.Final

Created-By: Apache Maven Bundle Plugin

Build-Jdk: 1.8.0_312

Implementation-URL: <https://netty.io/netty-codec-http/>

Found in path(s):

* /opt/cola/permits/1470280885_1668115952.2421944/0/netty-codec-http-4-1-84-final-jar/META-INF/MANIFEST.MF

No license file was found, but licenses were detected in source scan.

<!--

~ Copyright 2012 The Netty Project

~

~ The Netty Project licenses this file to you under the Apache License,
~ version 2.0 (the "License"); you may not use this file except in compliance
~ with the License. You may obtain a copy of the License at:

~

~ <https://www.apache.org/licenses/LICENSE-2.0>

~

~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
~ License for the specific language governing permissions and limitations
~ under the License.

-->

Found in path(s):

* /opt/cola/permits/1470280885_1668115952.2421944/0/netty-codec-http-4-1-84-final-jar/META-INF/maven/io.netty/netty-codec-http/pom.xml

No license file was found, but licenses were detected in source scan.

The Netty Project licenses this file to you under the Apache License,
version 2.0 (the "License"); you may not use this file except in compliance
with the License. You may obtain a copy of the License at:
distributed under the License is distributed on an "AS IS" BASIS, WITHOUT

Found in path(s):

* /opt/cola/permits/1470280885_1668115952.2421944/0/netty-codec-http-4-1-84-final-jar/META-INF/native-image/io.netty.codec-http/native-image.properties

1.278 fabric8-::-kubernetes-model-::-node

4.13.3

1.278.1 Available under license :

No license file was found, but licenses were detected in source scan.

Copyright (C) 2015 Red Hat, Inc.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE>

2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Found in path(s):

* /opt/cola/permits/1288520098_1647861833.26/0/kubernetes-model-node-4-13-3-jar/META-INF/maven/io.fabric8/kubernetes-model-node/pom.xml

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0

Bnd-LastModified: 1619068770964

Build-Jdk-Spec: 1.8

Bundle-Description: Java client for Kubernetes and OpenShift

Bundle-DocURL: <http://redhat.com>

Bundle-License: <http://www.apache.org/licenses/LICENSE-2.0.txt>

Bundle-ManifestVersion: 2

Bundle-Name: Fabric8 :: Kubernetes Model :: Node

Bundle-SymbolicName: io.fabric8.kubernetes-model-node

Bundle-Vendor: Red Hat

Bundle-Version: 4.13.3

Created-By: Apache Maven Bundle Plugin

Export-Package: io.fabric8.kubernetes.api.model.node.v1alpha1;uses:="com.fasterxml.xml.jackson.annotation,com.fasterxml.xml.jackson.databind,com.fasterxml.xml.jackson.databind.annotation,io.fabric8.kubernetes.api.builder,io.fabric8.kubernetes.api.model,io.fabric8.kubernetes.model.annotation";version="4.13.3",io.fabric8.kubernetes.api.model.node.v1beta1;use

s:="com.fasterxml.jackson.annotation,com.fasterxml.jackson.databind,com.fasterxml.jackson.databind.annotation,io.fabric8.kubernetes.api.builder,io.fabric8.kubernetes.api.model,io.fabric8.kubernetes.model.annotation";version="4.13.3"

Implementation-Title: Fabric8 :: Kubernetes Model :: Node

Implementation-Vendor: Red Hat

Implementation-Version: 4.13.3

Import-Package: com.fasterxml.jackson.annotation;version="[2.11,3)",com.fasterxml.jackson.databind;version="[2.11,3)",com.fasterxml.jackson.databind.annotation;version="[2.11,3)",io.fabric8.kubernetes.api.builder;version="[4.13,5)",io.fabric8.kubernetes.api.model,io.fabric8.kubernetes.model.annotation;version="[4.13,5)"

Require-Capability: osgi.ee;filter:="(&(osgi.ee=JavaSE)(version=1.8))"

Specification-Title: Fabric8 :: Kubernetes Model :: Node

Specification-Vendor: Red Hat

Specification-Version: 4.13

Tool: Bnd-5.1.1.202006162103

Found in path(s):

* /opt/cola/permits/1288520098_1647861833.26/0/kubernetes-model-node-4-13-3-jar/META-INF/MANIFEST.MF

No license file was found, but licenses were detected in source scan.

*

* Copyright (C) 2015 Red Hat, Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*

Found in path(s):

* /opt/cola/permits/1288520098_1647861833.26/0/kubernetes-model-node-4-13-3-jar/manifest.vm

1.279 protobuf-java-format 1.2

1.279.1 Available under license :

Copyright (c) 2009, Orbitz World Wide

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of the Orbitz World Wide nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2009, Orbitz LLC
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of the Orbitz LLC nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY

THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.280 junit-jupiter-junit-jupiter-api 5.9.1

1.280.1 Available under license :

Eclipse Public License - v 2.0

=====

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE (AGREEMENT). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. Definitions

Contribution means:

* **a)*** in the case of the initial Contributor, the initial content Distributed under this Agreement, and

* **b)*** in the case of each subsequent Contributor:

* **i)*** changes to the Program, and

* **j)*** additions to the Program;

where such changes and/or additions to the Program originate from and are Distributed by that particular Contributor. A Contribution originates from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include changes or additions to the Program that are not Modified Works.

Contributor means any person or entity that Distributes the Program.

Licensed Patents mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

Program means the Contributions Distributed in accordance with this Agreement.

Recipient means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

Derivative Works shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

Modified Works shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

Distribute means the acts of **a)*** distributing or **b)*** making available in any manner that enables the transfer

of a copy.

Source Code means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

Secondary License means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.

2. Grant of Rights

****a)**** Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

****b)**** Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

****c)**** Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

****d)**** Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

****e)**** Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

3. Requirements

****3.1)**** If a Contributor Distributes the Program in any form, then:

****a)**** the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and

****b)**** the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:

****i)**** effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied,

including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

* **ii)*** effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

* **iii)*** does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and

* **iv)*** requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.

3.2 When the Program is Distributed as Source Code:

* **a)*** it must be made available under this Agreement, or if the Program **(i)** is combined with other material in a separate file or files made available under a Secondary License, and **(ii)** the initial Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available under the terms of such Secondary Licenses, and

* **b)*** a copy of this Agreement must be included with each copy of the Program.

3.3 Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability (notices) contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

4. Commercial Distribution

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor (Commercial Contributor) hereby agrees to defend and indemnify every other Contributor (Indemnified Contributor) against any losses, damages and costs (collectively Losses) arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: **a)** promptly notify the Commercial Contributor in writing of such claim, and **b)** allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. No Warranty

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN AS IS BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION,

ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. Disclaimer of Liability

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. General

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are

created under this Agreement.

Exhibit A - Form of Secondary Licenses Notice

> This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: {name license(s), version(s), and exceptions or additional permissions here}.

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Open Source Licenses

=====

This product may include a number of subcomponents with separate copyright notices and license terms. Your use of the source code for these subcomponents is subject to the terms and conditions of the subcomponent's license, as noted in the LICENSE-<subcomponent>.md files.

1.281 snappy-java 1.1.8.4

1.281.1 Available under license :

This product includes software developed by Google
Snappy: <http://code.google.com/p/snappy/> (New BSD License)

This product includes software developed by Apache
PureJavaCrc32C from apache-hadoop-common <http://hadoop.apache.org/>
(Apache 2.0 license)

This library contained statically linked libstdc++. This inclusion is allowed by
"GCC Runtime Library Exception"
<http://gcc.gnu.org/onlinedocs/libstdc++/manual/license.html>

== Contributors ==

- * Tatu Saloranta
- * Providing benchmark suite
- * Alec Wysoker
- * Performance and memory usage improvement

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner

or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions

of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.282 micronaut-email 1.3.2

1.282.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright 2017-2021 original authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):

```
* /opt/cola/permits/1425206436_1663952955.879292/0/micronaut-email-1-3-2-sources-
jar/io/micronaut/email/Email.java
* /opt/cola/permits/1425206436_1663952955.879292/0/micronaut-email-1-3-2-sources-
jar/io/micronaut/email/DefaultEmailSender.java
* /opt/cola/permits/1425206436_1663952955.879292/0/micronaut-email-1-3-2-sources-
jar/io/micronaut/email/FromDecorator.java
* /opt/cola/permits/1425206436_1663952955.879292/0/micronaut-email-1-3-2-sources-
jar/io/micronaut/email/StringBody.java
* /opt/cola/permits/1425206436_1663952955.879292/0/micronaut-email-1-3-2-sources-
jar/io/micronaut/email/AsyncEmailSender.java
* /opt/cola/permits/1425206436_1663952955.879292/0/micronaut-email-1-3-2-sources-
jar/io/micronaut/email/EmailComposer.java
* /opt/cola/permits/1425206436_1663952955.879292/0/micronaut-email-1-3-2-sources-
jar/io/micronaut/email/Body.java
* /opt/cola/permits/1425206436_1663952955.879292/0/micronaut-email-1-3-2-sources-
jar/io/micronaut/email/AsyncTransactionalEmailSender.java
* /opt/cola/permits/1425206436_1663952955.879292/0/micronaut-email-1-3-2-sources-
jar/io/micronaut/email/configuration/FromConfigurationProperties.java
```

* /opt/cola/permits/1425206436_1663952955.879292/0/micronaut-email-1-3-2-sources-jar/io/micronaut/email/validation/AnyRecipient.java
* /opt/cola/permits/1425206436_1663952955.879292/0/micronaut-email-1-3-2-sources-jar/io/micronaut/email/configuration/FromConfiguration.java
* /opt/cola/permits/1425206436_1663952955.879292/0/micronaut-email-1-3-2-sources-jar/io/micronaut/email/TransactionalEmailSender.java
* /opt/cola/permits/1425206436_1663952955.879292/0/micronaut-email-1-3-2-sources-jar/io/micronaut/email/BodyType.java
* /opt/cola/permits/1425206436_1663952955.879292/0/micronaut-email-1-3-2-sources-jar/io/micronaut/email/EmailSender.java
* /opt/cola/permits/1425206436_1663952955.879292/0/micronaut-email-1-3-2-sources-jar/io/micronaut/email/Attachment.java
* /opt/cola/permits/1425206436_1663952955.879292/0/micronaut-email-1-3-2-sources-jar/io/micronaut/email/AbstractTransactionalEmailSender.java
* /opt/cola/permits/1425206436_1663952955.879292/0/micronaut-email-1-3-2-sources-jar/io/micronaut/email/DefaultAsyncEmailSender.java
* /opt/cola/permits/1425206436_1663952955.879292/0/micronaut-email-1-3-2-sources-jar/io/micronaut/email/Contact.java
* /opt/cola/permits/1425206436_1663952955.879292/0/micronaut-email-1-3-2-sources-jar/io/micronaut/email/validation/Recipients.java
* /opt/cola/permits/1425206436_1663952955.879292/0/micronaut-email-1-3-2-sources-jar/io/micronaut/email/EmailDecorator.java
* /opt/cola/permits/1425206436_1663952955.879292/0/micronaut-email-1-3-2-sources-jar/io/micronaut/email/EmailException.java
* /opt/cola/permits/1425206436_1663952955.879292/0/micronaut-email-1-3-2-sources-jar/io/micronaut/email/MultipartBody.java
* /opt/cola/permits/1425206436_1663952955.879292/0/micronaut-email-1-3-2-sources-jar/io/micronaut/email/validation/EmailMessages.java
* /opt/cola/permits/1425206436_1663952955.879292/0/micronaut-email-1-3-2-sources-jar/io/micronaut/email/validation/AnyRecipientConstraintValidatorFactory.java
* /opt/cola/permits/1425206436_1663952955.879292/0/micronaut-email-1-3-2-sources-jar/io/micronaut/email/TrackLinks.java
* /opt/cola/permits/1425206436_1663952955.879292/0/micronaut-email-1-3-2-sources-jar/io/micronaut/email/validation/RecipientsUtils.java

1.283 micronaut-r2dbc 4.0.0

1.283.1 Available under license :

mcr.microsoft.com/mssql/server:2017-CU12

Apache License

Version 2.0, January 2004

<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems,

and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and

limitations under the License.

1.284 antlr 2.7.7

1.284.1 Available under license :

[The BSD License]

Copyright (c) 2012 Terence Parr and Sam Harwell

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the author nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.285 jersey's-jersey 2.35

1.285.1 Available under license :

Notice for Jersey Core Common module

This content is produced and maintained by the Eclipse Jersey project.

* <https://projects.eclipse.org/projects/ee4j.jersey>

Trademarks

Eclipse Jersey is a trademark of the Eclipse Foundation.

Copyright

All content is the property of the respective authors or their employers. For more information regarding authorship of content, please consult the listed source code repository logs.

Declared Project Licenses

This program and the accompanying materials are made available under the terms of the Eclipse Public License v. 2.0 which is available at <http://www.eclipse.org/legal/epl-2.0>. This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License v. 2.0 are satisfied: GNU General Public License, version 2 with the GNU Classpath Exception which is available at <https://www.gnu.org/software/classpath/license.html>.

SPDX-License-Identifier: EPL-2.0 OR GPL-2.0 WITH Classpath-exception-2.0

Source Code

The project maintains the following source code repositories:

- * <https://github.com/eclipse-ee4j/jersey>

Third-party Content

Google Guava Version 18.0

- * License: Apache License, 2.0
- * Copyright: (C) 2009 The Guava Authors

JSR-166 Extension - JEP 266

- * License: Creative Commons 1.0 (CC0)
- * No copyright
- * Written by Doug Lea with assistance from members of JCP JSR-166
- * Expert Group and released to the public domain, as explained at
- * <http://creativecommons.org/publicdomain/zero/1.0/>

Notice for Jersey

This content is produced and maintained by the Eclipse Jersey project.

- * Project home: <https://projects.eclipse.org/projects/ee4j.jersey>

Trademarks

Eclipse Jersey is a trademark of the Eclipse Foundation.

Copyright

All content is the property of the respective authors or their employers. For more information regarding authorship of content, please consult the listed source code repository logs.

Declared Project Licenses

This program and the accompanying materials are made available under the terms of the Eclipse Public License v. 2.0 which is available at <http://www.eclipse.org/legal/epl-2.0>. This Source Code may also be made

available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License v. 2.0 are satisfied: GNU General Public License, version 2 with the GNU Classpath Exception which is available at <https://www.gnu.org/software/classpath/license.html>.

SPDX-License-Identifier: EPL-2.0 OR GPL-2.0 WITH Classpath-exception-2.0

Source Code

The project maintains the following source code repositories:

- * <https://github.com/eclipse-ee4j/jersey>

Third-party Content

Angular JS, v1.6.6

- * License MIT (<http://www.opensource.org/licenses/mit-license.php>)
- * Project: <http://angularjs.org>
- * Copyright: (c) 2010-2017 Google, Inc.

aopalliance Version 1

- * License: all the source code provided by AOP Alliance is Public Domain.
- * Project: <http://aopalliance.sourceforge.net>
- * Copyright: Material in the public domain is not protected by copyright

Bean Validation API 2.0.2

- * License: Apache License, 2.0
- * Project: <http://beanvalidation.org/1.1/>
- * Copyright: 2009, Red Hat, Inc. and/or its affiliates, and individual contributors
- * by the @authors tag.

Hibernate Validator CDI, 6.1.2.Final

- * License: Apache License, 2.0
- * Project: <https://beanvalidation.org/>
- * Repackaged in `org.glassfish.jersey.server.validation.internal.hibernate`

Bootstrap v3.3.7

- * License: MIT license (<https://github.com/twbs/bootstrap/blob/master/LICENSE>)
- * Project: <http://getbootstrap.com>
- * Copyright: 2011-2016 Twitter, Inc

Google Guava Version 18.0

- * License: Apache License, 2.0
- * Copyright (C) 2009 The Guava Authors

javax.inject Version: 1

- * License: Apache License, 2.0
- * Copyright (C) 2009 The JSR-330 Expert Group

Javassist Version 3.25.0-GA

- * License: Apache License, 2.0
- * Project: <http://www.javassist.org/>
- * Copyright (C) 1999- Shigeru Chiba. All Rights Reserved.

Jackson JAX-RS Providers Version 2.10.1

- * License: Apache License, 2.0
- * Project: <https://github.com/FasterXML/jackson-jaxrs-providers>
- * Copyright: (c) 2009-2011 FasterXML, LLC. All rights reserved unless otherwise indicated.

jQuery v1.12.4

- * License: jquery.org/license
- * Project: jquery.org
- * Copyright: (c) jQuery Foundation

jQuery Barcode plugin 0.3

- * License: MIT & GPL (<http://www.opensource.org/licenses/mit-license.php> & <http://www.gnu.org/licenses/gpl.html>)
- * Project: <http://www.pasella.it/projects/jquery/barcode>
- * Copyright: (c) 2009 Antonello Pasella antonello.pasella@gmail.com

JSR-166 Extension - JEP 266

- * License: CC0
- * No copyright
- * Written by Doug Lea with assistance from members of JCP JSR-166 Expert Group and released to the public domain, as explained at <http://creativecommons.org/publicdomain/zero/1.0/>

KineticJS, v4.7.1

- * License: MIT license (<http://www.opensource.org/licenses/mit-license.php>)
- * Project: <http://www.kineticjs.com>, <https://github.com/ericdrowell/KineticJS>
- * Copyright: Eric Rowell

org.objectweb.asm Version 8.0

- * License: Modified BSD (<http://asm.objectweb.org/license.html>)
- * Copyright (c) 2000-2011 INRIA, France Telecom. All rights reserved.

org.osgi.core version 6.0.0

- * License: Apache License, 2.0
- * Copyright (c) OSGi Alliance (2005, 2008). All Rights Reserved.

org.glassfish.jersey.server.internal.monitoring.core

- * License: Apache License, 2.0
- * Copyright (c) 2015-2018 Oracle and/or its affiliates. All rights reserved.
- * Copyright 2010-2013 Coda Hale and Yammer, Inc.

W3.org documents

- * License: W3C License
- * Copyright: Copyright (c) 1994-2001 World Wide Web Consortium, (Massachusetts Institute of Technology,

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial content Distributed under this Agreement, and

b) in the case of each subsequent Contributor:

- i) changes to the Program, and
- ii) additions to the Program;

where such changes and/or additions to the Program originate from and are Distributed by that particular Contributor. A Contribution "originates" from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include changes or additions to the Program that are not Modified Works.

"Contributor" means any person or entity that Distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions Distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

"Derivative Works" shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

"Modified Works" shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely

in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

"Distribute" means the acts of a) distributing or b) making available in any manner that enables the transfer of a copy.

"Source Code" means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Secondary License" means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

e) Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

3. REQUIREMENTS

3.1 If a Contributor Distributes the Program in any form, then:

a) the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and

b) the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:

i) effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and

iv) requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.

3.2 When the Program is Distributed as Source Code:

a) it must be made available under this Agreement, or if the Program (i) is combined with other material in a separate file or files made available under a Secondary License, and (ii) the initial Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available

under the terms of such Secondary Licenses, and

b) a copy of this Agreement must be included with each copy of the Program.

3.3 Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability ("notices") contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.

Exhibit A - Form of Secondary Licenses Notice

"This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: {name license(s), version(s), and exceptions or additional permissions here}."

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor
Boston, MA 02110-1335
USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such

interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in

accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute

so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free

programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or

(at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1335 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type
`show w'. This is free software, and you are welcome to redistribute
it under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the
program `Gnomovision' (which makes passes at compilers) written by
James Hacker.
```

```
signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

```
## CLASSPATH EXCEPTION
```

Linking this library statically or dynamically with other modules is

making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

1.286 micronaut-mongodb 4.6.0

1.286.1 Available under license :

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made,

use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability

incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.287 fabric8-::-kubernetes-model-::-rbac

4.13.3

1.287.1 Available under license :

No license file was found, but licenses were detected in source scan.

Copyright (C) 2015 Red Hat, Inc.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE>

2.0

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Found in path(s):

* /opt/cola/permits/1288519942_1647861852.16/0/kubernetes-model-rbac-4-13-3-jar/META-INF/maven/io.fabric8/kubernetes-model-rbac/pom.xml

No license file was found, but licenses were detected in source scan.

*

* Copyright (C) 2015 Red Hat, Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*

Found in path(s):

* /opt/cola/permits/1288519942_1647861852.16/0/kubernetes-model-rbac-4-13-3-jar/manifest.vm

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0

Bnd-LastModified: 1619068440938

Build-Jdk-Spec: 1.8

Bundle-Description: Java client for Kubernetes and OpenShift

Bundle-DocURL: <http://redhat.com>

Bundle-License: <http://www.apache.org/licenses/LICENSE-2.0.txt>

Bundle-ManifestVersion: 2

Bundle-Name: Fabric8 :: Kubernetes Model :: RBAC

Bundle-SymbolicName: io.fabric8.kubernetes-model-rbac

Bundle-Vendor: Red Hat

Bundle-Version: 4.13.3

Created-By: Apache Maven Bundle Plugin

Export-Package: io.fabric8.kubernetes.api.model.rbac;uses:="com.faster

xml.jackson.annotation,com.fasterxml.jackson.databind,com.fasterxml.j

ackson.databind.annotation,io.fabric8.kubernetes.api.builder,io.fabri

c8.kubernetes.api.model,io.fabric8.kubernetes.model.annotation";versi

on="4.13.3"

Implementation-Title: Fabric8 :: Kubernetes Model :: RBAC

Implementation-Vendor: Red Hat
Implementation-Version: 4.13.3
Import-Package: com.fasterxml.jackson.annotation;version="[2.11,3)",com.fasterxml.jackson.databind;version="[2.11,3)",com.fasterxml.jackson.databind.annotation;version="[2.11,3)",io.fabric8.kubernetes.api.builder;version="[4.13,5)",io.fabric8.kubernetes.api.model,io.fabric8.kubernetes.model.annotation;version="[4.13,5)"
Require-Capability: osgi.ee;filter="(&(osgi.ee=JavaSE)(version=1.8))"
Specification-Title: Fabric8 :: Kubernetes Model :: RBAC
Specification-Vendor: Red Hat
Specification-Version: 4.13
Tool: Bnd-5.1.1.202006162103

Found in path(s):

* /opt/cola/permits/1288519942_1647861852.16/0/kubernetes-model-rbac-4-13-3-jar/META-INF/MANIFEST.MF

1.288 config 5.5.5

1.288.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
<!--~
~ Copyright 2015 Confluent Inc.
~
~ Licensed under the Apache License, Version 2.0 (the "License");
~ you may not use this file except in compliance with the License.
~ You may obtain a copy of the License at
~
~ http://www.apache.org/licenses/LICENSE-2.0
~
~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS,
~ WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
~ See the License for the specific language governing permissions and
~ limitations under the License.
~-->
```

Found in path(s):

* /opt/cola/permits/1341639818_1654816582.0388222/0/common-config-5-5-5-jar/META-INF/maven/io.confluent/common-config/pom.xml

1.289 prometheus-java-span-context-supplier-opentelemetry 0.15.0

1.289.1 Available under license :

No license file was found, but licenses were detected in source scan.

<url><http://www.apache.org/licenses/LICENSE-2.0.txt></url>

Found in path(s):

* /opt/cola/permits/1341561651_1654803080.3794436/0/simpleclient-tracer-otel-0-15-0-sources-jar/META-INF/maven/io.prometheus/simpleclient_tracer_otel/pom.xml

1.290 micrometer-registry-jmx 1.9.4

1.290.1 Available under license :

Micrometer

Copyright (c) 2017-Present VMware, Inc. All Rights Reserved.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<https://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

This product contains a modified portion of 'io.netty.util.internal.logging',
in the Netty/Common library distributed by The Netty Project:

- * Copyright 2013 The Netty Project
- * License: Apache License v2.0
- * Homepage: <https://netty.io>

This product contains a modified portion of 'StringUtils.isBlank()',
in the Commons Lang library distributed by The Apache Software Foundation:

- * Copyright 2001-2019 The Apache Software Foundation
- * License: Apache License v2.0
- * Homepage: <https://commons.apache.org/proper/commons-lang/>

This product contains a modified portion of 'JsonUtf8Writer',
in the Moshi library distributed by Square, Inc:

- * Copyright 2010 Google Inc.
- * License: Apache License v2.0
- * Homepage: <https://github.com/square/moshi>

This product contains a modified portion of the 'org.springframework.lang' package in the Spring Framework library, distributed by VMware, Inc:

- * Copyright 2002-2019 the original author or authors.
- * License: Apache License v2.0
- * Homepage: <https://spring.io/projects/spring-framework>
Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate

as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify

the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include

the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<https://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.291 testcontainers-bom 1.17.5

1.291.1 Available under license :

Apache License

Version 2.0, January 2004

<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed

as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this

License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.292 argparse 0.8.1

1.292.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2011 Tatsuhiro Tsujikawa

*

* Permission is hereby granted, free of charge, to any person

* obtaining a copy of this software and associated documentation

* files (the "Software"), to deal in the Software without

* restriction, including without limitation the rights to use, copy,

* modify, merge, publish, distribute, sublicense, and/or sell copies

* of the Software, and to permit persons to whom the Software is
* furnished to do so, subject to the following conditions:
*
* The above copyright notice and this permission notice shall be
* included in all copies or substantial portions of the Software.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
* NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS
* BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN
* ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
* CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
* SOFTWARE.
*/

Found in path(s):

* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/inf/ArgumentType.java
* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/inf/ArgumentParser.java
* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/helper/CJKTextWidthCounter.java
* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/impl/action/HelpArgumentAction.java
* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/helper/PrefixPattern.java
* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/helper/TextWidthCounter.java
* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/internal/UnrecognizedArgumentException.java
* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/annotation/Arg.java
* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/inf/ArgumentParserException.java
* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/impl/action/StoreConstArgumentAction.java
* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/inf/ArgumentAction.java
* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/impl/Arguments.java
* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/inf/ArgumentGroup.java
* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/inf/Subparsers.java
* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/helper/ReflectHelper.java
* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-

jar/net/sourceforge/argparse4j/internal/UnrecognizedCommandException.java
* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/impl/action/StoreArgumentAction.java
* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/inf/Namespace.java
* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/impl/choice/CollectionArgumentChoice.java
* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/ArgumentParsers.java
* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/internal/ArgumentParserImpl.java
* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/impl/action/StoreFalseArgumentAction.java
* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/impl/action/AppendConstArgumentAction.java
* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/internal/ArgumentGroupImpl.java
* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/inf/Argument.java
* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/inf/ArgumentChoice.java
* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/inf/FeatureControl.java
* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/impl/choice/RangeArgumentChoice.java
* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/internal/SubparsersImpl.java
* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/internal/SubparserImpl.java
* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/inf/Subparser.java
* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/impl/type/StringArgumentType.java
* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/impl/action/AppendArgumentAction.java
* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/helper/ASCIITextWidthCounter.java
* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/impl/type/FileVerification.java
* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/helper/TextHelper.java
* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/impl/type/EnumArgumentType.java
* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/impl/action/StoreTrueArgumentAction.java
* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/impl/type/ConstructorArgumentType.java
* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-

jar/net/sourceforge/argparse4j/internal/ParseState.java
* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/internal/ArgumentImpl.java
* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/impl/action/VersionArgumentAction.java
No license file was found, but licenses were detected in source scan.

```
/*  
* Copyright (C) 2015 Andrew January  
*  
* Permission is hereby granted, free of charge, to any person  
* obtaining a copy of this software and associated documentation  
* files (the "Software"), to deal in the Software without  
* restriction, including without limitation the rights to use, copy,  
* modify, merge, publish, distribute, sublicense, and/or sell copies  
* of the Software, and to permit persons to whom the Software is  
* furnished to do so, subject to the following conditions:  
*  
* The above copyright notice and this permission notice shall be  
* included in all copies or substantial portions of the Software.  
*  
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,  
* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF  
* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND  
* NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS  
* BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN  
* ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN  
* CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE  
* SOFTWARE.  
*/
```

Found in path(s):

* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/impl/type/CaseInsensitiveEnumStringArgumentType.java
* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/impl/type/CaseInsensitiveEnumNameArgumentType.java
* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/impl/type/EnumStringArgumentType.java
No license file was found, but licenses were detected in source scan.

```
/*  
* Copyright (C) 2015 Tatsuhiro Tsujikawa  
*  
* Permission is hereby granted, free of charge, to any person  
* obtaining a copy of this software and associated documentation  
* files (the "Software"), to deal in the Software without  
* restriction, including without limitation the rights to use, copy,  
* modify, merge, publish, distribute, sublicense, and/or sell copies
```

* of the Software, and to permit persons to whom the Software is
* furnished to do so, subject to the following conditions:
*
* The above copyright notice and this permission notice shall be
* included in all copies or substantial portions of the Software.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
* NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS
* BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN
* ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
* CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
* SOFTWARE.
*/

Found in path(s):

* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/impl/type/BooleanArgumentType.java
* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/inf/MetavarInference.java
No license file was found, but licenses were detected in source scan.

/*

* Licensed to the Apache Software Foundation (ASF) under one
* or more contributor license agreements. See the NOTICE file
* distributed with this work for additional information
* regarding copyright ownership. The ASF licenses this file
* to you under the Apache License, Version 2.0 (the
* "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing,
* software distributed under the License is distributed on an
* "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
* KIND, either express or implied. See the License for the
* specific language governing permissions and limitations
* under the License.
*/

Found in path(s):

* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/impl/type/FileArgumentType.java
No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2013 Adam Parkin

*

* Permission is hereby granted, free of charge, to any person
* obtaining a copy of this software and associated documentation
* files (the "Software"), to deal in the Software without
* restriction, including without limitation the rights to use, copy,
* modify, merge, publish, distribute, sublicense, and/or sell copies
* of the Software, and to permit persons to whom the Software is
* furnished to do so, subject to the following conditions:

*

* The above copyright notice and this permission notice shall be
* included in all copies or substantial portions of the Software.

*

* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
* NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS
* BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN
* ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
* CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
* SOFTWARE.

*/

Found in path(s):

* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/helper/HelpScreenException.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2013 Tatsuhiro Tsujikawa

*

* Permission is hereby granted, free of charge, to any person
* obtaining a copy of this software and associated documentation
* files (the "Software"), to deal in the Software without
* restriction, including without limitation the rights to use, copy,
* modify, merge, publish, distribute, sublicense, and/or sell copies
* of the Software, and to permit persons to whom the Software is
* furnished to do so, subject to the following conditions:

*

* The above copyright notice and this permission notice shall be
* included in all copies or substantial portions of the Software.

*

* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
* NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS
* BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN
* ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN

* CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
* SOFTWARE.
*/

Found in path(s):

* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/impl/action/CountArgumentAction.java
* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/impl/type/ReflectArgumentType.java
No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2012 Tatsuhiro Tsujikawa

*

* Permission is hereby granted, free of charge, to any person
* obtaining a copy of this software and associated documentation
* files (the "Software"), to deal in the Software without
* restriction, including without limitation the rights to use, copy,
* modify, merge, publish, distribute, sublicense, and/or sell copies
* of the Software, and to permit persons to whom the Software is
* furnished to do so, subject to the following conditions:

*

* The above copyright notice and this permission notice shall be
* included in all copies or substantial portions of the Software.

*

* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
* NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS
* BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN
* ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
* CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
* SOFTWARE.

*/

Found in path(s):

* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-
jar/net/sourceforge/argparse4j/inf/MutuallyExclusiveGroup.java
No license file was found, but licenses were detected in source scan.

/*

* Licensed to the Apache Software Foundation (ASF) under one or more
* contributor license agreements. See the NOTICE file distributed with
* this work for additional information regarding copyright ownership.
* The ASF licenses this file to You under the Apache License, Version 2.0
* (the "License"); you may not use this file except in compliance with
* the License. You may obtain a copy of the License at

*

- * <http://www.apache.org/licenses/LICENSE-2.0>
- *
- * Unless required by applicable law or agreed to in writing, software
- * distributed under the License is distributed on an "AS IS" BASIS,
- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.
- */

Found in path(s):

* /opt/cola/permits/1274701436_1645235201.06/0/argparse4j-0-8-1-sources-jar/net/sourceforge/argparse4j/internal/TerminalWidth.java

1.293 micronaut-sql-libraries 4.7.2

1.293.1 Available under license :

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but

not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their

Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with

the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.294 jackson-protobuf-support 0.9.9-jackson2.9-proto2

1.294.1 Available under license :

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all

other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and

subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed

as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the

Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.295 micrometer-metrics/micrometer 1.9.4

1.295.1 Available under license :

This product includes 'ANTLR', which is released under the following license(s):

BSD-3 <<http://opensource.org/licenses/BSD-3-Clause>>

This product includes 'ASM', which is released under the following license(s):

ASM (BSD-3) <<http://asm.ow2.org/license.html>>

This product includes 'Apache Cassandra', which is released under the following license(s):

Apache License 2.0 <<http://www.apache.org/licenses/LICENSE-2.0>>

This product includes 'Apache Commons CLI', which is released under the following license(s):

Apache License 2.0 <<http://www.apache.org/licenses/LICENSE-2.0>>

This product includes 'Apache Commons Codec', which is released under the following license(s):

Apache License 2.0 <<http://www.apache.org/licenses/LICENSE-2.0>>

This product includes 'Apache Commons Logging', which is released under the following license(s):

Apache License 2.0 <<http://www.apache.org/licenses/LICENSE-2.0>>

This product includes 'GSON', which is released under the following license(s):

Apache License 2.0 <<http://www.apache.org/licenses/LICENSE-2.0>>

This product includes 'Guava', which is released under the following license(s):

Apache License 2.0 <<http://www.apache.org/licenses/LICENSE-2.0>>

This product includes 'HttpClient', which is released under the following license(s):

Apache License 2.0 <<http://www.apache.org/licenses/LICENSE-2.0>>

This product includes 'JAXB', which is released under the following license(s):

CDDL-1.0 <<https://opensource.org/licenses/CDDL-1.0>>

This product includes 'JRegex', which is released under the following license(s):

BSD-3 <<http://opensource.org/licenses/BSD-3-Clause>>

This product includes 'JSON.simple', which is released under the following license(s):

Apache License 2.0 <<http://www.apache.org/licenses/LICENSE-2.0>>

This product includes 'Jackson', which is released under the following license(s):

Apache License 2.0 <<http://www.apache.org/licenses/LICENSE-2.0>>

This product includes 'Javassist', which is released under the following license(s):

Apache License 2.0 <<http://www.apache.org/licenses/LICENSE-2.0>>

This product includes 'LogBack', which is released under the following license(s):

Eclipse Public License 1.0 <<http://www.eclipse.org/legal/epl-v10.html>>

This product includes 'Reflections', which is released under the following license(s):

Do What The Fuck You Want To Public License (WTFPL) <<http://www.wtfpl.net>>

This product includes 'SLF4J', which is released under the following license(s):

The MIT License (MIT) <<http://opensource.org/licenses/MIT>>

This product includes 'SnakeYAML', which is released under the following license(s):

Apache License 2.0 <<http://www.apache.org/licenses/LICENSE-2.0>>

All other components of this product are: Copyright (c) 2010-2017 New Relic, Inc. All rights reserved.

Certain inventions disclosed in this file may be claimed within patents owned or patent applications

filed by New Relic, Inc. or third parties. Subject to the terms of this notice, New Relic grants you a nonexclusive, nontransferable license, without the right to sublicense, to (a) install and execute one copy of these files on any number of workstations owned or controlled by you and (b) distribute verbatim copies of these files to third parties. As a condition to the foregoing grant, you must provide this notice along with each copy you distribute and you must not remove, alter, or obscure this notice.

All other use, reproduction, modification, distribution, or other exploitation of these files is strictly prohibited, except as may be set forth in a separate written license agreement between you and New Relic. The terms of any such license agreement will control over this notice. The license stated above will be automatically terminated and revoked if you exceed its scope or violate any of the terms of this notice.

This License does not grant permission to use the trade names, trademarks, service marks, or product names of New Relic, except as required for reasonable and customary use in describing the origin of this file and reproducing the content of this notice. You may not mark or brand this file with any trade name, trademarks, service marks, or product names other than the original brand (if any) provided by New Relic.

Unless otherwise expressly agreed by New Relic in a separate written license agreement, these files are provided AS IS, WITHOUT WARRANTY OF ANY KIND, including without any implied warranties of MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, or NON-INFRINGEMENT. As a condition to your use of these files, you are solely responsible for such use. New Relic will have no liability to you for direct, indirect, consequential, incidental, special, or punitive damages or for lost profits or data.

Micrometer

Copyright (c) 2017-Present VMware, Inc. All Rights Reserved.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<https://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

This product contains a modified portion of 'io.netty.util.internal.logging',
in the Netty/Common library distributed by The Netty Project:

* Copyright 2013 The Netty Project

* License: Apache License v2.0

* Homepage: <https://netty.io>

This product contains a modified portion of 'StringUtils.isBlank()',
in the Commons Lang library distributed by The Apache Software Foundation:

* Copyright 2001-2019 The Apache Software Foundation

* License: Apache License v2.0

* Homepage: <https://commons.apache.org/proper/commons-lang/>

This product contains a modified portion of 'JsonUtf8Writer',
in the Moshi library distributed by Square, Inc:

* Copyright 2010 Google Inc.

* License: Apache License v2.0

* Homepage: <https://github.com/square/moshi>

This product contains a modified portion of the 'org.springframework.lang'
package in the Spring Framework library, distributed by VMware, Inc:

* Copyright 2002-2019 the original author or authors.

* License: Apache License v2.0

* Homepage: <https://spring.io/projects/spring-framework>

All components of this product are: Copyright (c) 2010-2015 New Relic, Inc. All rights reserved.

Certain inventions disclosed in this file may be claimed within patents owned or patent applications filed by New Relic, Inc. or third parties. Subject to the terms of this notice, New Relic grants you a nonexclusive, nontransferable license, without the right to sublicense, to (a) install and execute one copy of these files on any number of workstations owned or controlled by you and (b) distribute verbatim copies of these files to third parties. As a condition to the foregoing grant, you must provide this notice along with each copy you distribute and you must not remove, alter, or obscure this notice.

All other use, reproduction, modification, distribution, or other exploitation of these files is strictly prohibited, except as may be set forth in a separate written license agreement between you and New Relic. The terms of any such license agreement will control over this notice. The license stated above will be automatically terminated and revoked if you exceed its scope or violate any of the terms of this notice.

This License does not grant permission to use the trade names, trademarks, service marks, or product names of New Relic, except as required for reasonable and customary use in describing the origin of this file and reproducing the content of this notice. You may not mark or brand this file with any trade name, trademarks, service marks, or product names other than the original brand (if any) provided by New Relic.

Unless otherwise expressly agreed by New Relic in a separate written license agreement, these files are provided AS IS, WITHOUT WARRANTY OF ANY KIND, including without any implied warranties of MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, or NON-INFRINGEMENT. As a

condition to your use of these files, you are solely responsible for such use. New Relic will have no liability to you for direct, indirect, consequential, incidental, special, or punitive damages or for lost profits or data.

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain

separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the

origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<https://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.296 osgi-resource-locator-bundle 1.0.3

1.296.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Copyright (c) 2009, 2018 Oracle and/or its affiliates. All rights reserved.
 *
 * This program and the accompanying materials are made available under the
 * terms of the Eclipse Public License v. 2.0, which is available at
 * http://www.eclipse.org/legal/epl-2.0.
 *
 * This Source Code may also be made available under the following Secondary
 * Licenses when the conditions for such availability set forth in the
 * Eclipse Public License v. 2.0 are satisfied: GNU General Public License,
 * version 2 with the GNU Classpath Exception, which is available at
 * https://www.gnu.org/software/classpath/license.html.
 *
 * SPDX-License-Identifier: EPL-2.0 OR GPL-2.0 WITH Classpath-exception-2.0
 */
```

Found in path(s):

```
* /opt/cola/permits/1340031775_1654680102.675777/0/osgi-resource-locator-1-0-3-sources-2-
jar/org/glassfish/hk2/osgiresourcelocator/ServiceLoader.java
* /opt/cola/permits/1340031775_1654680102.675777/0/osgi-resource-locator-1-0-3-sources-2-
jar/org/glassfish/hk2/osgiresourcelocator/ResourceFinder.java
* /opt/cola/permits/1340031775_1654680102.675777/0/osgi-resource-locator-1-0-3-sources-2-
jar/org/glassfish/hk2/osgiresourcelocator/Activator.java
* /opt/cola/permits/1340031775_1654680102.675777/0/osgi-resource-locator-1-0-3-sources-2-
jar/org/glassfish/hk2/osgiresourcelocator/ServiceLoaderImpl.java
* /opt/cola/permits/1340031775_1654680102.675777/0/osgi-resource-locator-1-0-3-sources-2-
jar/org/glassfish/hk2/osgiresourcelocator/ResourceFinderImpl.java
```

1.297 apache-commons-beanutils 1.9.4

1.297.1 Available under license :

Apache Commons BeanUtils
Copyright 2000-2019 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work

(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses

granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]"

replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.298 fabric8-:::-kubernetes-model-::- coordination 4.13.3

1.298.1 Available under license :

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0
Bnd-LastModified: 1619068568928
Build-Jdk-Spec: 1.8
Bundle-Description: Java client for Kubernetes and OpenShift
Bundle-DocURL: <http://redhat.com>
Bundle-License: <http://www.apache.org/licenses/LICENSE-2.0.txt>
Bundle-ManifestVersion: 2
Bundle-Name: Fabric8 :: Kubernetes Model :: Coordination
Bundle-SymbolicName: io.fabric8.kubernetes-model-coordination
Bundle-Vendor: Red Hat
Bundle-Version: 4.13.3
Created-By: Apache Maven Bundle Plugin
Export-Package: io.fabric8.kubernetes.api.model.coordination.v1;uses:=
"com.fasterxml.jackson.annotation,com.fasterxml.jackson.databind,com.
fasterxml.jackson.databind.annotation,io.fabric8.kubernetes.api.build
er,io.fabric8.kubernetes.api.model,io.fabric8.kubernetes.model.annota
tion";version="4.13.3"
Implementation-Title: Fabric8 :: Kubernetes Model :: Coordination
Implementation-Vendor: Red Hat

Implementation-Version: 4.13.3

Import-Package: com.fasterxml.jackson.annotation;version="[2.11,3)",com.fasterxml.jackson.databind;version="[2.11,3)",com.fasterxml.jackson.databind.annotation;version="[2.11,3)",io.fabric8.kubernetes.api.builer;version="[4.13,5)",io.fabric8.kubernetes.api.model,io.fabric8.kubernetes.model.annotation;version="[4.13,5)"

Require-Capability: osgi.ee;filter="(&(osgi.ee=JavaSE)(version=1.8))"

Specification-Title: Fabric8 :: Kubernetes Model :: Coordination

Specification-Vendor: Red Hat

Specification-Version: 4.13

Tool: Bnd-5.1.1.202006162103

Found in path(s):

* /opt/cola/permits/1288519842_1647861722.74/0/kubernetes-model-coordination-4-13-3-jar/META-INF/MANIFEST.MF

No license file was found, but licenses were detected in source scan.

*

* Copyright (C) 2015 Red Hat, Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*

Found in path(s):

* /opt/cola/permits/1288519842_1647861722.74/0/kubernetes-model-coordination-4-13-3-jar/manifest.vm

No license file was found, but licenses were detected in source scan.

Copyright (C) 2015 Red Hat, Inc.

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE>

2.0

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Found in path(s):

* /opt/cola/permits/1288519842_1647861722.74/0/kubernetes-model-coordination-4-13-3-jar/META-INF/maven/io.fabric8/kubernetes-model-coordination/pom.xml

1.299 fabric8-::-kubernetes-model-::-policy

4.13.3

1.299.1 Available under license :

No license file was found, but licenses were detected in source scan.

Copyright (C) 2015 Red Hat, Inc.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE>

2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Found in path(s):

* /opt/cola/permits/1288519814_1647861838.2/0/kubernetes-model-policy-4-13-3-jar/META-INF/maven/io.fabric8/kubernetes-model-policy/pom.xml

No license file was found, but licenses were detected in source scan.

*

* Copyright (C) 2015 Red Hat, Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

- * See the License for the specific language governing permissions and
- * limitations under the License.
- *

Found in path(s):

* /opt/cola/permits/1288519814_1647861838.2/0/kubernetes-model-policy-4-13-3-jar/manifest.vm

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0

Bnd-LastModified: 1619068639575

Build-Jdk-Spec: 1.8

Bundle-Description: Java client for Kubernetes and OpenShift

Bundle-DocURL: <http://redhat.com>

Bundle-License: <http://www.apache.org/licenses/LICENSE-2.0.txt>

Bundle-ManifestVersion: 2

Bundle-Name: Fabric8 :: Kubernetes Model :: Policy

Bundle-SymbolicName: io.fabric8.kubernetes-model-policy

Bundle-Vendor: Red Hat

Bundle-Version: 4.13.3

Created-By: Apache Maven Bundle Plugin

Export-Package: io.fabric8.kubernetes.api.model.policy;uses:="com.fasterxml.jackson.annotation,com.fasterxml.jackson.databind,com.fasterxml.jackson.databind.annotation,io.fabric8.kubernetes.api.builder,io.fabric8.kubernetes.api.model,io.fabric8.kubernetes.model.annotation";version="4.13.3"

Implementation-Title: Fabric8 :: Kubernetes Model :: Policy

Implementation-Vendor: Red Hat

Implementation-Version: 4.13.3

Import-Package: com.fasterxml.jackson.annotation;version="[2.11,3)",com.fasterxml.jackson.databind;version="[2.11,3)",com.fasterxml.jackson.databind.annotation;version="[2.11,3)",io.fabric8.kubernetes.api.builder;version="[4.13,5)",io.fabric8.kubernetes.api.model,io.fabric8.kubernetes.model.annotation;version="[4.13,5)"

Require-Capability: osgi.ee;filter="(&(osgi.ee=JavaSE)(version=1.8))"

Specification-Title: Fabric8 :: Kubernetes Model :: Policy

Specification-Vendor: Red Hat

Specification-Version: 4.13

Tool: Bnd-5.1.1.202006162103

Found in path(s):

* /opt/cola/permits/1288519814_1647861838.2/0/kubernetes-model-policy-4-13-3-jar/META-INF/MANIFEST.MF

1.300 jackson-core 2.14.0

1.300.1 Available under license :

Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library. It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has been in development since 2007. It is currently developed by a community of developers.

Licensing

Jackson 2.x core and extension components are licensed under Apache License 2.0. To find the details that apply to this artifact see the accompanying LICENSE file.

Credits

A list of contributors may be found from CREDITS(-2.x) file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work,

where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or

for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason

of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.301 jakarta-mail 1.6.2

1.301.1 Available under license :

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.1

1. Definitions.

1.1. "Contributor" means each individual or entity that creates or contributes to the creation of Modifications.

1.2. "Contributor Version" means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. "Covered Software" means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. "Executable" means the Covered Software in any form other than Source Code.

1.5. "Initial Developer" means the individual or entity that first makes Original Software available under this License.

1.6. "Larger Work" means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. "License" means this document.

1.8. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. "Modifications" means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. "Original Software" means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. "Source Code" means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than

fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in

combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the

Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Oracle is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section

4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as "Participant") alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. If You assert a patent infringement claim against Participant alleging that the Participant Software directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

6.4. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION

AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and "commercial computer software documentation" as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION
LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor
Boston, MA 02110-1335
USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis

or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other

recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not

signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the

original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1335 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type
`show w'. This is free software, and you are welcome to redistribute
it under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if

necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

#

Certain source files distributed by Oracle America, Inc. and/or its affiliates are subject to the following clarification and special exception to the GPLv2, based on the GNU Project exception for its Classpath libraries, known as the GNU Classpath Exception, but only where Oracle has expressly included in the particular source file's header the words "Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the LICENSE file that accompanied this code."

You should also note that Oracle includes multiple, independent programs in this software package. Some of those programs are provided under licenses deemed incompatible with the GPLv2 by the Free Software Foundation and others. For example, the package includes programs licensed under the Apache License, Version 2.0. Such programs are licensed to you under their original licenses.

Oracle facilitates your further distribution of this package by adding the Classpath Exception to the necessary parts of its GPLv2 code, which permits you to use that code in combination with other independent modules not licensed under the GPLv2. However, note that this would not permit you to commingle code under an incompatible license with Oracle's GPLv2 licensed code by, for example, cutting and pasting such code into a file also containing Oracle's GPLv2 licensed code and then distributing the result. Additionally, if you were to remove the Classpath Exception from any of the files to which it applies and distribute the result, you would likely be required to license some or all of the other code in that distribution under the GPLv2 as well, and since the GPLv2 is incompatible with the license terms of some items included in the distribution by Oracle, removing the Classpath Exception could therefore effectively compromise your ability to further distribute the package.

Proceed with caution and we recommend that you obtain the advice of a lawyer skilled in open source matters before removing the Classpath Exception or making modifications to this package which may subsequently be redistributed and/or involve the use of third party software.

CLASSPATH EXCEPTION

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

1.302 metrics-utility-servlets 4.1.29

1.302.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the

direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of

this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and

wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor

has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 2010-2013 Coda Hale and Yammer, Inc., 2014-2020 Dropwizard Team

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.303 micronaut-crac 1.0.1

1.303.1 Available under license :

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of

the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works

that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A

PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.304 asm 7.2

1.304.1 Available under license :

<OWNER> = Regents of the University of California

<ORGANIZATION> = University of California, Berkeley

<YEAR> = 1998

In the original BSD license, both occurrences of the phrase "COPYRIGHT HOLDERS AND CONTRIBUTORS" in the disclaimer read "REGENTS AND CONTRIBUTORS".

Here is the license template:

Copyright (c) <YEAR>, <OWNER>

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the <ORGANIZATION> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.305 jose4j 0.7.9

1.305.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*

- * Licensed to the Apache Software Foundation (ASF) under one or more
- * contributor license agreements. See the NOTICE file distributed with
- * this work for additional information regarding copyright ownership.
- * The ASF licenses this file to You under the Apache License, Version 2.0
- * (the "License"); you may not use this file except in compliance with
- * the License. You may obtain a copy of the License at
- *
- * <http://www.apache.org/licenses/LICENSE-2.0>
- *
- * Unless required by applicable law or agreed to in writing, software
- * distributed under the License is distributed on an "AS IS" BASIS,
- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.
- */

Found in path(s):

- * /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/base64url/internal/apache/commons/codec/binary/BaseNCodec.java
- * /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/base64url/internal/apache/commons/codec/binary/Base64.java

No license file was found, but licenses were detected in source scan.

/*

- * Copyright 2012-2017 Brian Campbell
- *
- * Licensed under the Apache License, Version 2.0 (the "License");
- * you may not use this file except in compliance with the License.
- * You may obtain a copy of the License at
- *
- * <http://www.apache.org/licenses/LICENSE-2.0>
- *
- * Unless required by applicable law or agreed to in writing, software
- * distributed under the License is distributed on an "AS IS" BASIS,
- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/json/JsonUtil.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwe/SimpleAeadCipher.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/lang/InvalidAlgorithmException.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jws/BaseSignatureAlgorithm.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwt/consumer/JwsCustomizer.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/keys/RsaKeyUtil.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwt/consumer/ErrorCodeValidator.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwe/ContentEncryptionAlgorithm.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/keys/PbkdfKey.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/keys/BigEndianBigInteger.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/keys/resolvers/JwksDecryptionKeyResolver.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwe/AesGcmContentEncryptionAlgorithm.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwa/AlgorithmInfo.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwx/Headers.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/lang/StringUtil.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwt/JwtClaims.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/zip/DeflateRFC1951CompressionAlgorithm.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/keys/AesKey.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwe/KeyManagementAlgorithmIdentifiers.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwe/DirectKeyManagementAlgorithm.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/keys/resolvers/HttpsJwksVerificationKeyResolver.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/lang/HashUtil.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwa/AlgorithmAvailability.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/mac/MacUtil.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwe/EcdhKeyAgreementWithAesKeyWrapAlgorithm.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/base64url/Base64Url.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-

jar/org/jose4j/lang/ExceptionHelp.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-
jar/org/jose4j/keys/EllipticCurves.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-
jar/org/jose4j/jwk/RsaJwkGenerator.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-
jar/org/jose4j/jwe/AesKeyWrapManagementAlgorithm.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-
jar/org/jose4j/lang/DefaultByteGenerator.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-
jar/org/jose4j/jwt/consumer/JwtContext.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwe/CipherUtil.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-
jar/org/jose4j/keys/resolvers/VerificationKeyResolver.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-
jar/org/jose4j/lang/JoseException.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-
jar/org/jose4j/jws/HmacUsingShaAlgorithm.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-
jar/org/jose4j/jwk/JsonWebKeySet.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-
jar/org/jose4j/jws/JsonWebSignatureAlgorithm.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-
jar/org/jose4j/jwe/ContentEncryptionKeyDescriptor.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-
jar/org/jose4j/jwa/AlgorithmConstraints.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-
jar/org/jose4j/jwe/AesCbcHmacSha2ContentEncryptionAlgorithm.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-
jar/org/jose4j/jws/JsonWebSignature.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-
jar/org/jose4j/jwe/KeyManagementAlgorithm.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-
jar/org/jose4j/jwe/EcdhKeyAgreementAlgorithm.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-
jar/org/jose4j/jwe/ContentEncryptionParts.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-
jar/org/jose4j/lang/BouncyCastleProviderHelp.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-
jar/org/jose4j/json/JsonHeaderUtil.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-
jar/org/jose4j/jwt/GeneralJwtException.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-
jar/org/jose4j/jwt/consumer/ErrorCodeValidatorAdapter.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-
jar/org/jose4j/jwa/AlgorithmFactoryFactory.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-
jar/org/jose4j/jwe/ContentEncryptionAlgorithmIdentifiers.java

* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwk/EcJwkGenerator.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwk/Use.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwt/consumer/AudValidator.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/keys/EcKeyUtil.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwx/KeyValidationSupport.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/zip/CompressionAlgorithmIdentifiers.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwt/consumer/IssValidator.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwt/consumer/InvalidJwtException.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/lang/IntegrityException.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwx/HeaderParameterNames.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwk/RsaJsonWebKey.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwk/JsonWebKey.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/keys/X509Util.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwk/OctetSequenceJsonWebKey.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/lang/ByteUtil.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwt/ReservedClaimNames.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/keys/resolvers/X509VerificationKeyResolver.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwt/consumer/SimpleKeyResolver.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwk/EllipticCurveJsonWebKey.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/lang/ByteGenerator.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwx/CompactSerializer.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwt/consumer/JwtConsumer.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwk/PublicJsonWebKey.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwx/CompactSerialization.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwt/NumericDate.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jws/UnsecuredNoneAlgorithm.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/http/SimpleGet.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/lang/JsonHelp.java

* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwx/JsonWebStructure.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/keys/KeyPairUtil.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwt/IntDate.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jws/RsaUsingShaAlgorithm.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/lang/UncheckedJoseException.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwt/consumer/ErrorCodes.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwe/kdf/PasswordBasedKeyDerivationFunction2.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jws/AlgorithmIdentifiers.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwe/JsonWebEncryption.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwa/AlgorithmFactory.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/keys/resolvers/DecryptionKeyResolver.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/keys/resolvers/JwksVerificationKeyResolver.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwt/consumer/NumericDateValidator.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwe/Pbes2HmacShaWithAesKeyWrapAlgorithm.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwe/CipherStrengthSupport.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/base64url/SimplePEMEncoder.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/http/Get.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwe/ContentEncryptionKeys.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwt/consumer/JwtConsumerBuilder.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwt/consumer/JweCustomizer.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwe/ContentEncryptionHelp.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwe/InitializationVectorHelp.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwa/Algorithm.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwe/AesGcmKeyEncryptionAlgorithm.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwt/MalformedClaimException.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/keys/KeyPersuasion.java

* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwk/OctJwkGenerator.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/keys/HmacKey.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwk/KeyOperations.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwt/consumer/Validator.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwt/consumer/JtiValidator.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwk/VerificationJwkSelector.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/lang/UnresolvableKeyException.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/zip/CompressionAlgorithm.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwt/consumer/SubValidator.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwe/RsaKeyManagementAlgorithm.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jws/EcdsaUsingShaAlgorithm.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jca/ProviderContext.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwe/WrappingKeyManagementAlgorithm.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2012-2021 Brian Campbell
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwe/kdf/ConcatenationKeyDerivationFunctionWithSha256.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwe/kdf/ConcatKeyDerivationFunctionFactory.java

* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwe/kdf/ConcatKeyDerivationFunction.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwe/kdf/KdfUtil.java
No license file was found, but licenses were detected in source scan.

```
/*  
* Copyright 2012-2018 Brian Campbell  
*  
* Licensed under the Apache License, Version 2.0 (the "License");  
* you may not use this file except in compliance with the License.  
* You may obtain a copy of the License at  
*  
* http://www.apache.org/licenses/LICENSE-2.0  
*  
* Unless required by applicable law or agreed to in writing, software  
* distributed under the License is distributed on an "AS IS" BASIS,  
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
* See the License for the specific language governing permissions and  
* limitations under the License.  
*/
```

Found in path(s):

* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwk/HttpsJwks.java
No license file was found, but licenses were detected in source scan.

```
/*  
* Copyright 2012-2017 Brian Campbell  
*  
* Licensed under the Apache License, Version 2.0 (the "License");  
* you may not use this file except in compliance with the License.  
* You may obtain a copy of the License at  
*  
* http://www.apache.org/licenses/LICENSE-2.0  
*  
* Unless required by applicable law or agreed to in writing, software  
* distributed under the License is distributed on an "AS IS" BASIS,  
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
* See the License for the specific language governing permissions and  
* limitations under the License.  
*/
```

Found in path(s):

* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/lang/InvalidKeyException.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwk/SimpleJwkFilter.java
* /opt/cola/permits/1425205813_1663952843.717818/0/jose4j-0-7-9-sources-3-jar/org/jose4j/jwk/DecryptionJwkSelector.java

1.306 mockito 2.27.0

1.306.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the

editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the

same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

====

Copyright (c) 2016 Mockito contributors

This program is made available under the terms of the MIT License.

====

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation

source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable

(except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and

may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify,

defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

The MIT License

Copyright (c) 2007 Mockito contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER

LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.307 micronaut-micrometer 4.6.1

1.307.1 Available under license :

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object

form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a

file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.308 profiler 1.0.2

1.308.1 Available under license :

The MIT License (MIT)

Copyright (c) 2013 G4 Code

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.309 curator-testing 5.2.1

1.309.1 Available under license :

Curator Testing

Copyright 2011-2022 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object

form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a

file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.310 junit-platform-junit-platform-commons

1.9.1

1.310.1 Available under license :

Eclipse Public License - v 2.0

=====

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE (AGREEMENT). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. Definitions

Contribution means:

* **a)** in the case of the initial Contributor, the initial content Distributed under this Agreement, and

* **b)** in the case of each subsequent Contributor:

* **i)** changes to the Program, and

* **j)** additions to the Program;

where such changes and/or additions to the Program originate from and are Distributed by that particular Contributor. A Contribution originates from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include changes or additions to the Program that are not Modified Works.

Contributor means any person or entity that Distributes the Program.

Licensed Patents mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

Program means the Contributions Distributed in accordance with this Agreement.

Recipient means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

Derivative Works shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

Modified Works shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

Distribute means the acts of ****a)**** distributing or ****b)**** making available in any manner that enables the transfer of a copy.

Source Code means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

Secondary License means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.

2. Grant of Rights

****a)**** Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

****b)**** Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

****c)**** Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

****d)**** Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

****e)**** Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

3. Requirements

****3.1)**** If a Contributor Distributes the Program in any form, then:

****a)**** the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and

****b)**** the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:

****i)**** effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

****ii)**** effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

****iii)**** does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and

****iv)**** requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.

****3.2)**** When the Program is Distributed as Source Code:

****a)**** it must be made available under this Agreement, or if the Program ****i)**** is combined with other material in a separate file or files made available under a Secondary License, and ****ii)**** the initial Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available under the terms of such Secondary Licenses, and

****b)**** a copy of this Agreement must be included with each copy of the Program.

****3.3)**** Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability (notices) contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

4. Commercial Distribution

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor (Commercial Contributor) hereby agrees to defend and indemnify every other Contributor (Indemnified Contributor) against any losses, damages and costs (collectively Losses) arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: ****a)**** promptly

notify the Commercial Contributor in writing of such claim, and ****b)**** allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. No Warranty

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN AS IS BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. Disclaimer of Liability

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. General

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this

Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.

Exhibit A - Form of Secondary Licenses Notice

> This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: {name license(s), version(s), and exceptions or additional permissions here }.

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Open Source Licenses

=====

This product may include a number of subcomponents with separate copyright notices and license terms. Your use of the source code for these subcomponents is subject to the terms and conditions of the subcomponent's license, as noted in the LICENSE-<subcomponent>.md files.

1.311 rest-assured:-bom 5.2.0

1.311.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2020 the original author or authors.

*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/mapping/JsonbMapper.groovy
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/ResponseParserRegistrar.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2019 the original author or authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* <http://www.apache.org/licenses/LICENSE-2.0>
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/specification/PreemptiveAuthSpec.java
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/specification/ResponseSpecification.java
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/assertion/HeaderMatcher.groovy
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/MultiValueEntity.java
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/specification/FilterableRequestSpecification.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/specification/QueryableRequestSpecification.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/http/BoundaryExtractor.groovy

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/matcher/ResponseAwareMatcherComposer.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/ResponseSpecificationImpl.groovy

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/PreemptiveAuthSpecImpl.groovy

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/mapper/ObjectMapper.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/authentication/NTLMAuthScheme.groovy

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/print/ResponsePrinter.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/authentication/OAuth2Scheme.groovy

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/mapping/JakartaEEMapper.groovy

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/config/MultiPartConfig.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/filter/log/ResponseLoggingFilter.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/http/StringHashMap.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/config/DecoderConfig.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/filter/CsrfFilter.groovy

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/matcher/xml/XmlDtdMatcher.groovy

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/multipart/MultiPartInternal.groovy

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/ResponseLogSpecificationImpl.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/proxy/RestAssuredProxySelectorRoutePlanner.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/MapCreator.groovy

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/http/HttpContextDecorator.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/mapping/GsonMapper.groovy

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/config/OAuthConfig.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/mapping/ObjectMapperSerializationContextImpl.groovy

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/http/CharsetExtractor.groovy

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/http/HttpResponseContentTypeFinder.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/authentication/PreemptiveBasicAuthScheme.groovy

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/builder/ResponseSpecBuilder.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/mapping/ObjectMapping.groovy

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/config/CsrfConfig.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/specification/RequestSender.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/response/ResponseBody.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/log/LogRepository.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/TrustAndKeystoreSpec.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/specification/RequestSenderOptions.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/config/HeaderConfig.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/RestAssuredResponseOptionsImpl.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/config/XmlConfig.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/specification/LogSpecification.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/RequestLogSpecificationImpl.groovy

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/response/Response.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/config/ConnectionConfig.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/config/SSLConfig.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/http/GZIPEncoding.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/matcher/RestAssuredMatchers.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/http/ContentTypeExtractor.groovy

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/response/ResponseOptions.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/authentication/AuthenticationScheme.groovy

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/multipart/RestAssuredMultiPartEntity.groovy
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/specification/ResponseLogSpecification.java
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/util/IOUtils.java
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/RestAssuredResponseImpl.java
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/http/Header.java
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/TestSpecificationImpl.groovy
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/http/DeflateEncoding.java
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/http/HttpRequestFactory.java
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/matcher/xml/LoadFromClasspathSupport.groovy
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/mapper/ObjectMapperSerializationContext.java
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/config/LogConfig.java
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/response/ValidatableResponseOptions.java
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/multipart/MultiPartSpecificationImpl.groovy
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/config/FailureConfig.java
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/NoParameterValue.groovy
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/config/ParamConfig.java
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/ContentParser.groovy
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/serialization/SerializationSupport.java
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/http/URIBuilder.java
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/authentication/FormAuthConfig.java
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/authentication/BasicAuthScheme.groovy
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/filter/Filter.java
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/filter/OrderedFilter.java
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/filter/log/LogDetail.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/http/Headers.java
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/config/HttpClientConfig.java
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/filter/session/SessionFilter.java
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/specification/RequestSpecification.java
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/builder/RequestSpecBuilder.java
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/config/MatcherConfig.java
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/mapping/JohnzonMapper.groovy
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/matcher/xml/XmlXsdMatcher.groovy
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/filter/cookie/CookieFilter.java
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/authentication/PreemptiveAuthProvider.java
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/config/JsonConfig.java
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/filter/SendRequestFilter.groovy
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/listener/ResponseValidationFailureListener.java
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/assertion/DetailedCookieAssertion.groovy
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/authentication/CertificateAuthSettings.java
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/specification/MultiPartSpecification.java
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/config/ObjectMapperConfig.java
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/filter/log/StatusCodeBasedLoggingFilter.java
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/authentication/CertAuthScheme.groovy
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/http/Method.java
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/filter/log/RequestLoggingFilter.java
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/authentication/NoAuthScheme.groovy
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/http/ContentEncoding.java
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/support/PathSupport.groovy

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/UriValidator.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/filter/FormAuthFilter.groovy

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/authentication/OAuthScheme.groovy

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/http/EncoderRegistry.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/response/ResponseBodyData.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/config/RestAssuredConfig.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/mapping/ObjectMapperDeserializationContextImpl.groovy

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/authentication/ExplicitNoAuthScheme.groovy

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/http/ResponseParseException.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/config/Config.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/http/BasicNameValuePairWithNoValueSupport.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/mapper/ObjectMapperDeserializationContext.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/mapper/ObjectMapperType.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/authentication/OAuthSignature.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/specification/FilterableResponseSpecification.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/specification/RequestLogSpecification.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/LogRequestAndResponseOnFailListener.groovy

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/util/MatcherErrorMessageBuilder.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/specification/Argument.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/http/HttpResponseDecorator.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/print/RequestPrinter.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/specification/RedirectSpecification.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/proxy/RestAssuredProxySelector.groovy

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/http/Cookie.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/ValidatableResponseOptionsImpl.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/builder/ResponseBuilder.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/response/ResponseBodyExtractionOptions.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/specification/AuthenticationSpecification.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/http/ContentEncodingRegistry.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/parsing/Parser.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/config/EncoderConfig.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/authentication/FormAuthScheme.groovy

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/builder/MultiPartSpecBuilder.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/assertion/StreamVerifier.groovy

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/NameAndValue.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/http/Status.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/RestAssured.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/specification/ProxySpecification.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/ValidatableResponseImpl.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/response/ExtractableResponse.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/config/SessionConfig.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/TrustAndKeystoreSpecImpl.groovy

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/util/SafeExceptionRethrower.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/filter/FilterContextImpl.groovy

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/support/FileReader.groovy

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/config/RedirectConfig.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/mapping/Jackson2Mapper.groovy

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/RedirectSpecificationImpl.groovy

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/filter/time/TimingFilter.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/SpecificationMerger.groovy

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/http/ContentType.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/http/ContentTypeSubTypeExtractor.groovy

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/mapping/JacksonIMapper.groovy

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/specification/SpecificationQuerier.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/AuthenticationSpecificationImpl.groovy

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/matcher/ResponseAwareMatcher.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/LogSpecificationImpl.groovy

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/http/CustomHttpMethod.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/http/HttpResponseException.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/filter/FilterContext.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/filter/log/UrlDecoder.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/http/Cookies.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/support/ParameterUpdater.groovy

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/filter/log/ErrorLoggingFilter.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/RequestSpecificationImpl.groovy

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/response/ValidatableResponse.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/response/ValidatableResponseLogSpec.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/spi/AuthFilter.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/support/Prettifier.groovy

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/internal/http/AuthConfig.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/response/Validatable.java

* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-jar/io/restassured/authentication/PreemptiveOAuth2HeaderScheme.groovy

```
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-
jar/io/restassured/internal/http/HTTPBuilder.java
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-
jar/io/restassured/matcher/DetailedCookieMatcher.java
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-
jar/io/restassured/internal/mapping/JaxbMapper.groovy
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-
jar/io/restassured/internal/RestAssuredResponseOptionsGroovyImpl.groovy
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-
jar/io/restassured/internal/support/CloseHttpClientConnectionInputStreamWrapper.java
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-
jar/io/restassured/assertion/ResponseTimeMatcher.groovy
```

No license file was found, but licenses were detected in source scan.

```
/*
```

```
* Copyright 2022 the original author or authors.
```

```
*
```

```
* Licensed under the Apache License, Version 2.0 (the "License");
```

```
* you may not use this file except in compliance with the License.
```

```
* You may obtain a copy of the License at
```

```
*
```

```
* http://www.apache.org/licenses/LICENSE-2.0
```

```
*
```

```
* Unless required by applicable law or agreed to in writing, software
```

```
* distributed under the License is distributed on an "AS IS" BASIS,
```

```
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
```

```
* See the License for the specific language governing permissions and
```

```
* limitations under the License.
```

```
*/
```

Found in path(s):

```
* /opt/cola/permits/1473459990_1668571024.6479063/0/rest-assured-5-2-0-sources-
jar/io/restassured/internal/RestAssuredHttpBuilder.java
```

1.312 javax-ws-rs-api 2.1.1

1.312.1 Available under license :

Found license 'General Public License 2.0' in '* Copyright (c) 2013, 2017 Oracle and/or its affiliates. All rights reserved. * This program and the accompanying materials are made available under the * terms of the Eclipse Public License v. 2.0, which is available at * Eclipse Public License v. 2.0 are satisfied: GNU General Public License,'

Found license 'Eclipse Public License 1.0' in '* Copyright (c) 2010, 2017 Oracle and/or its affiliates. All rights reserved. * This program and the accompanying materials are made available under the * terms of the Eclipse Public License v. 2.0, which is available at * Eclipse Public License v. 2.0 are satisfied: GNU General Public License,'

Found license 'General Public License 2.0' in '* Copyright (c) 2017 Oracle and/or its affiliates. All rights reserved. * This program and the accompanying materials are made available under the * terms of the Eclipse Public License v. 2.0, which is available at * Eclipse Public License v. 2.0 are satisfied: GNU General Public License,'

Found license 'Eclipse Public License 1.0' in '* Copyright (c) 2012, 2017 Oracle and/or its affiliates. All rights reserved. * This program and the accompanying materials are made available under the * terms of the Eclipse Public License v. 2.0, which is available at * Eclipse Public License v. 2.0 are satisfied: GNU General Public License, * This method is reserved for future use. Proprietary JAX-RS extensions may leverage the method.'

Found license 'Eclipse Public License 1.0' in '* Copyright (c) 2012, 2017 Oracle and/or its affiliates. All rights reserved. * This program and the accompanying materials are made available under the * terms of the Eclipse Public License v. 2.0, which is available at * Eclipse Public License v. 2.0 are satisfied: GNU General Public License,'

Found license 'Eclipse Public License 1.0' in '* Copyright (c) 2015, 2017 Oracle and/or its affiliates. All rights reserved. * This program and the accompanying materials are made available under the * terms of the Eclipse Public License v. 2.0, which is available at * Eclipse Public License v. 2.0 are satisfied: GNU General Public License,'

Found license 'Eclipse Public License 1.0' in '* Copyright (c) 2011, 2017 Oracle and/or its affiliates. All rights reserved. * This program and the accompanying materials are made available under the * terms of the Eclipse Public License v. 2.0, which is available at * Eclipse Public License v. 2.0 are satisfied: GNU General Public License,'

Found license 'Eclipse Public License 1.0' in '* Copyright (c) 2013, 2017 Oracle and/or its affiliates. All rights reserved. * This program and the accompanying materials are made available under the * terms of the Eclipse Public License v. 2.0, which is available at * Eclipse Public License v. 2.0 are satisfied: GNU General Public License,'

Found license 'General Public License 2.0' in '* Copyright (c) 2012, 2017 Oracle and/or its affiliates. All rights reserved. * This program and the accompanying materials are made available under the * terms of the Eclipse Public License v. 2.0, which is available at * Eclipse Public License v. 2.0 are satisfied: GNU General Public License, * This method is reserved for future use. Proprietary JAX-RS extensions may leverage the method.'

Found license 'General Public License 2.0' in '* Copyright (c) 2011, 2017 Oracle and/or its affiliates. All rights reserved. * This program and the accompanying materials are made available under the * terms of the Eclipse Public License v. 2.0, which is available at * Eclipse Public License v. 2.0 are satisfied: GNU General Public License,'

Found license 'General Public License 2.0' in '* Copyright (c) 2010, 2017 Oracle and/or its affiliates. All rights reserved. * This program and the accompanying materials are made available under the * terms of the Eclipse Public License v. 2.0, which is available at * Eclipse Public License v. 2.0 are satisfied: GNU General Public License,'

Found license 'General Public License 2.0' in '* Copyright (c) 2012, 2017 Oracle and/or its affiliates. All rights reserved. * This program and the accompanying materials are made available under the * terms of the Eclipse Public License v. 2.0, which is available at * Eclipse Public License v. 2.0 are satisfied: GNU General Public License,'

Found license 'General Public License 2.0' in '* Copyright (c) 2015, 2017 Oracle and/or its affiliates. All rights reserved. * This program and the accompanying materials are made available under the * terms of the Eclipse Public License v. 2.0, which is available at * Eclipse Public License v. 2.0 are satisfied: GNU General Public License,'

Found license 'Eclipse Public License 1.0' in '* Copyright (c) 2017 Oracle and/or its affiliates. All rights reserved. * This program and the accompanying materials are made available under the * terms of the Eclipse Public License v. 2.0, which is available at * Eclipse Public License v. 2.0 are satisfied: GNU General Public License,'

1.313 error_prone_annotations 2.10.0

1.313.1 Available under license :

No license file was found, but licenses were detected in source scan.

<!--

Copyright 2015 The Error Prone Authors.

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

-->

Found in path(s):

* /opt/cola/permits/1337069915_1654119249.153431/0/error-prone-annotations-2-10-0-jar/META-INF/maven/com.google.errorprone/error_prone_annotations/pom.xml

1.314 aws-sdk-for-java---bom 1.12.301

1.314.1 Available under license :

Apache License

Version 2.0, January 2004

<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical

transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable

by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use,

reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.315 metrics-health-checks 4.1.11

1.315.1 Available under license :

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0

Build-Jdk-Spec: 1.8

Bundle-Description: An addition to Metrics which provides the ability to run application-specific health checks, allowing you to check your application's health in production.

Bundle-License: <https://www.apache.org/licenses/LICENSE-2.0.html>

Bundle-ManifestVersion: 2

Bundle-Name: Metrics Health Checks

Bundle-SymbolicName: io.dropwizard.metrics.healthchecks

Bundle-Version: 4.1.11

Created-By: Apache Maven Bundle Plugin

Export-Package: com.codahale.metrics.health;uses:="com.codahale.metrics";version="4.1.11",com.codahale.metrics.health.annotation;version="4.1.11",com.codahale.metrics.health.jvm;uses:="com.codahale.metrics.he

alth,com.codahale.metrics.jvm";version="4.1.11"
Implementation-Title: Metrics Health Checks
Implementation-Version: 4.1.11
Import-Package: com.codahale.metrics;resolution:=optional;version="[4.1,5)",com.codahale.metrics.health,com.codahale.metrics.health.annotation,com.codahale.metrics.jvm;resolution:=optional;version="[4.1,5)",org.slf4j;version="[1.6.0,2.0.0)"
Private-Package: com.codahale.metrics.health,com.codahale.metrics.health.annotation,com.codahale.metrics.health.jvm
Require-Capability: osgi.ee;filter="(&(osgi.ee=JavaSE)(version=1.8))"
Tool: Bnd-5.1.1.202006162103

Found in path(s):

* /opt/cola/permits/1341130506_1654768535.3386562/0/metrics-healthchecks-4-1-11-5-jar/META-INF/MANIFEST.MF

1.316 micronaut-picocli-configuration 4.3.0

1.316.1 Available under license :

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made,

use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability

incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.317 okhttp-logging-interceptor 3.12.12

1.317.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2015 Square, Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):

* /opt/cola/permits/1294625109_1649800590.01/0/logging-interceptor-3-12-12-sources-
jar/okhttp3/logging/HttpLoggingInterceptor.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2018 Square, Inc.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1294625109_1649800590.01/0/logging-interceptor-3-12-12-sources-
jar/okhttp3/logging/LoggingEventListener.java

1.318 micronaut-kafka 4.4.0

1.318.1 Available under license :

Apache License

Version 2.0, January 2004

<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all

other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and

subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed

as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the

Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.319 hdrhistogram 2.1.12

1.319.1 Available under license :

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0

Bnd-LastModified: 1575980548657

Build-Jdk: 1.8.0_232

Built-By: gil

Bundle-Description: HdrHistogram supports the recording and analyzing sampled data value counts across a configurable integer value range with configurable value precision within the range. Value precision is expressed as the number of significant digits in the value recording, and provides control over value quantization behavior across the value range and the subsequent value resolution at any given level.

Bundle-License: <http://creativecommons.org/publicdomain/zero/1.0/>, <https://opensource.org/licenses/BSD-2-Clause>

Bundle-ManifestVersion: 2

Bundle-Name: HdrHistogram

Bundle-SymbolicName: org.hdrhistogram.HdrHistogram

Bundle-Version: 2.1.12

Created-By: Apache Maven Bundle Plugin

Export-Package: org.HdrHistogram;version="2.1.12",org.HdrHistogram.packagedarray;version="2.1.12"

Implementation-Title: HdrHistogram

Implementation-Vendor-Id: org.hdrhistogram

Implementation-Version: 2.1.12

Require-Capability: osgi.ee;filter="(&(osgi.ee=JavaSE)(version=1.7))"

Specification-Title: HdrHistogram

Specification-Version: 2.1.12

Tool: Bnd-2.3.0.201405100607

Found in path(s):

* /opt/cola/permits/1185944801_1628011594.04/0/hdrhistogram-2-1-12-3-jar/META-INF/MANIFEST.MF

No license file was found, but licenses were detected in source scan.

* public domain, as explained at <http://creativecommons.org/publicdomain/zero/1.0/>
<name>Public Domain, per Creative Commons CC0</name>

Found in path(s):

* /opt/cola/permits/1185944801_1628011594.04/0/hdrhistogram-2-1-12-3-jar/META-INF/maven/org.hdrhistogram/HdrHistogram/pom.xml

1.320 protocol-buffer-java-util-package 3.20.1

1.320.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
// Copyright 2008 Google Inc. All rights reserved.  
// Redistribution and use in source and binary forms, with or without  
// modification, are permitted provided that the following conditions are  
// * Redistributions of source code must retain the above copyright  
// notice, this list of conditions and the following disclaimer.  
// * Redistributions in binary form must reproduce the above  
// copyright notice, this list of conditions and the following disclaimer  
// in the documentation and/or other materials provided with the  
// * Neither the name of Google Inc. nor the names of its  
// this software without specific prior written permission.
```

Found in path(s):

```
* /opt/cola/permits/1427929573_1664346243.9376018/0/protobuf-java-util-3-20-1-sources-  
jar/com/google/protobuf/util/FieldMaskUtil.java  
* /opt/cola/permits/1427929573_1664346243.9376018/0/protobuf-java-util-3-20-1-sources-  
jar/com/google/protobuf/util/Structs.java  
* /opt/cola/permits/1427929573_1664346243.9376018/0/protobuf-java-util-3-20-1-sources-  
jar/com/google/protobuf/util/Timestamps.java  
* /opt/cola/permits/1427929573_1664346243.9376018/0/protobuf-java-util-3-20-1-sources-  
jar/com/google/protobuf/util/Values.java  
* /opt/cola/permits/1427929573_1664346243.9376018/0/protobuf-java-util-3-20-1-sources-  
jar/com/google/protobuf/util/Durations.java  
* /opt/cola/permits/1427929573_1664346243.9376018/0/protobuf-java-util-3-20-1-sources-  
jar/com/google/protobuf/util/FieldMaskTree.java  
* /opt/cola/permits/1427929573_1664346243.9376018/0/protobuf-java-util-3-20-1-sources-  
jar/com/google/protobuf/util/JsonFormat.java
```

1.321 jakarta-activation 1.1.1

1.321.1 Available under license :

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

1. Definitions.

1.1. Contributor. means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version. means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software. means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable. means the Covered Software in any form other than Source Code.

1.5. Initial Developer. means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work. means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License. means this document.

1.8. Licensable. means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications. means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software. means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims. means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code. means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You. (or .Your.) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, .You. includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, .control. means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may

create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN .AS IS. BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as .Participant.) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY.S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH

LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a .commercial item., as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of .commercial computer software. (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and .commercial computer software documentation. as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction.s conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys. fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307
USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications

and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only

way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE

PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.

Copyright (C)

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author

Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

"CLASSPATH" EXCEPTION TO THE GPL VERSION 2

Certain source files distributed by Sun Microsystems, Inc. are subject to the following clarification and special exception to the GPL Version 2, but only where Sun has expressly included in the particular source file's header the words

"Sun designates this particular file as subject to the "Classpath" exception as provided by Sun in the License file that accompanied this code."

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License Version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module.? An independent module is a module which is not derived from or based on this library.? If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so.? If you do not wish to do so, delete this exception statement from your version.

1.322 io-projectreactor 3.4.23

1.322.1 Available under license :

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by

the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained

within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be

liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<https://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Copyright (c) 2006, Ivan Sagalaev

All rights reserved.

Redistribution and use in source and binary forms, with or without

modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of highlight.js nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS AND CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.323 scala-java8-compat_2.12 1.0.2

1.323.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * Scala (https://www.scala-lang.org)
 *
 * Copyright EPFL and Lightbend, Inc.
 *
 * Licensed under Apache License 2.0
 * (http://www.apache.org/licenses/LICENSE-2.0).
 *
 * See the NOTICE file distributed with this work for
 * additional information regarding copyright ownership.
 */
```

Found in path(s):

```
* /opt/cola/permits/1425205998_1663952624.3076522/0/scala-java8-compat-2-11-1-0-2-sources-jar/scala/compat/java8/converterImpl/StepsLikeIndexed.scala
* /opt/cola/permits/1425205998_1663952624.3076522/0/scala-java8-compat-2-11-1-0-2-sources-jar/scala/compat/java8/converterImpl/StepsHashTable.scala
* /opt/cola/permits/1425205998_1663952624.3076522/0/scala-java8-compat-2-11-1-0-2-sources-jar/scala/compat/java8/converterImpl/Accumulates.scala
* /opt/cola/permits/1425205998_1663952624.3076522/0/scala-java8-compat-2-11-1-0-2-sources-
```

jar/scala/compat/java8/converterImpl/StepsArray.scala
* /opt/cola/permits/1425205998_1663952624.3076522/0/scala-java8-compat-2-11-1-0-2-sources-
jar/scala/compat/java8/DurationConverters.scala
* /opt/cola/permits/1425205998_1663952624.3076522/0/scala-java8-compat-2-11-1-0-2-sources-
jar/scala/compat/java8/converterImpl/StepsRange.scala
* /opt/cola/permits/1425205998_1663952624.3076522/0/scala-java8-compat-2-11-1-0-2-sources-
jar/scala/compat/java8/StreamConverters.scala
* /opt/cola/permits/1425205998_1663952624.3076522/0/scala-java8-compat-2-11-1-0-2-sources-
jar/scala/compat/java8/wrappers/IteratorPrimitiveIntWrapper.java
* /opt/cola/permits/1425205998_1663952624.3076522/0/scala-java8-compat-2-11-1-0-2-sources-
jar/scala/compat/java8/collectionImpl/Stepper.scala
* /opt/cola/permits/1425205998_1663952624.3076522/0/scala-java8-compat-2-11-1-0-2-sources-
jar/scala/concurrent/java8/FutureConvertersImpl.scala
* /opt/cola/permits/1425205998_1663952624.3076522/0/scala-java8-compat-2-11-1-0-2-sources-
jar/scala/compat/java8/WrapperTraits.scala
* /opt/cola/permits/1425205998_1663952624.3076522/0/scala-java8-compat-2-11-1-0-2-sources-
jar/scala/compat/java8/collectionImpl/LongAccumulator.scala
* /opt/cola/permits/1425205998_1663952624.3076522/0/scala-java8-compat-2-11-1-0-2-sources-
jar/scala/compat/java8/converterImpl/StepsBitSet.scala
* /opt/cola/permits/1425205998_1663952624.3076522/0/scala-java8-compat-2-11-1-0-2-sources-
jar/scala/compat/java8/converterImpl/StepsString.scala
* /opt/cola/permits/1425205998_1663952624.3076522/0/scala-java8-compat-2-11-1-0-2-sources-
jar/scala/compat/java8/converterImpl/StepsFlatHashTable.scala
* /opt/cola/permits/1425205998_1663952624.3076522/0/scala-java8-compat-2-11-1-0-2-sources-
jar/scala/compat/java8/converterImpl/StepsVector.scala
* /opt/cola/permits/1425205998_1663952624.3076522/0/scala-java8-compat-2-11-1-0-2-sources-
jar/scala/compat/java8/converterImpl/StepsLikeGapped.scala
* /opt/cola/permits/1425205998_1663952624.3076522/0/scala-java8-compat-2-11-1-0-2-sources-
jar/scala/compat/java8/converterImpl/StepsLikeImmHashMap.scala
* /opt/cola/permits/1425205998_1663952624.3076522/0/scala-java8-compat-2-11-1-0-2-sources-
jar/scala/compat/java8/converterImpl/StepsImmHashMap.scala
* /opt/cola/permits/1425205998_1663952624.3076522/0/scala-java8-compat-2-11-1-0-2-sources-
jar/scala/compat/java8/converterImpl/StepsLinearSeq.scala
* /opt/cola/permits/1425205998_1663952624.3076522/0/scala-java8-compat-2-11-1-0-2-sources-
jar/scala/compat/java8/converterImpl/StepsIterator.scala
* /opt/cola/permits/1425205998_1663952624.3076522/0/scala-java8-compat-2-11-1-0-2-sources-
jar/scala/compat/java8/runtime/CollectionInternals.java
* /opt/cola/permits/1425205998_1663952624.3076522/0/scala-java8-compat-2-11-1-0-2-sources-
jar/scala/concurrent/java8/FuturesConvertersImplCompat.scala
* /opt/cola/permits/1425205998_1663952624.3076522/0/scala-java8-compat-2-11-1-0-2-sources-
jar/scala/compat/java8/converterImpl/AccumulatorConverters.scala
* /opt/cola/permits/1425205998_1663952624.3076522/0/scala-java8-compat-2-11-1-0-2-sources-
jar/scala/compat/java8/wrappers/IteratorPrimitiveLongWrapper.java
* /opt/cola/permits/1425205998_1663952624.3076522/0/scala-java8-compat-2-11-1-0-2-sources-
jar/scala/compat/java8/collectionImpl/IntAccumulator.scala
* /opt/cola/permits/1425205998_1663952624.3076522/0/scala-java8-compat-2-11-1-0-2-sources-
jar/scala/compat/java8/converterImpl/StepConverters.scala
* /opt/cola/permits/1425205998_1663952624.3076522/0/scala-java8-compat-2-11-1-0-2-sources-

```

jar/scala/compat/java8/collectionImpl/DoubleAccumulator.scala
* /opt/cola/permits/1425205998_1663952624.3076522/0/scala-java8-compat-2-11-1-0-2-sources-
jar/scala/compat/java8/converterImpl/StepsIndexedSeq.scala
* /opt/cola/permits/1425205998_1663952624.3076522/0/scala-java8-compat-2-11-1-0-2-sources-
jar/scala/compat/java8/runtime/LambdaDeserializer.scala
* /opt/cola/permits/1425205998_1663952624.3076522/0/scala-java8-compat-2-11-1-0-2-sources-
jar/scala/compat/java8/converterImpl/StepsLikeTrieIterator.scala
* /opt/cola/permits/1425205998_1663952624.3076522/0/scala-java8-compat-2-11-1-0-2-sources-
jar/scala/compat/java8/converterImpl/MakesSteppers.scala
* /opt/cola/permits/1425205998_1663952624.3076522/0/scala-java8-compat-2-11-1-0-2-sources-
jar/scala/compat/java8/converterImpl/StepsIterable.scala
* /opt/cola/permits/1425205998_1663952624.3076522/0/scala-java8-compat-2-11-1-0-2-sources-
jar/scala/compat/java8/ScalaStreamSupport.java
* /opt/cola/permits/1425205998_1663952624.3076522/0/scala-java8-compat-2-11-1-0-2-sources-
jar/scala/compat/java8/converterImpl/StepsLikeSliced.scala
* /opt/cola/permits/1425205998_1663952624.3076522/0/scala-java8-compat-2-11-1-0-2-sources-
jar/scala/compat/java8/collectionImpl/AccumulatorLike.scala
* /opt/cola/permits/1425205998_1663952624.3076522/0/scala-java8-compat-2-11-1-0-2-sources-
jar/scala/compat/java8/wrappers/IteratorPrimitiveDoubleWrapper.java
* /opt/cola/permits/1425205998_1663952624.3076522/0/scala-java8-compat-2-11-1-0-2-sources-
jar/scala/compat/java8/collectionImpl/Accumulator.scala
* /opt/cola/permits/1425205998_1663952624.3076522/0/scala-java8-compat-2-11-1-0-2-sources-
jar/scala/compat/java8/OptionConverters.scala
* /opt/cola/permits/1425205998_1663952624.3076522/0/scala-java8-compat-2-11-1-0-2-sources-
jar/scala/compat/java8/SplitterConverters.scala
* /opt/cola/permits/1425205998_1663952624.3076522/0/scala-java8-compat-2-11-1-0-2-sources-
jar/scala/compat/java8/PrimitiveIteratorConversions.scala
* /opt/cola/permits/1425205998_1663952624.3076522/0/scala-java8-compat-2-11-1-0-2-sources-
jar/scala/compat/java8/converterImpl/StepsWithTail.scala
* /opt/cola/permits/1425205998_1663952624.3076522/0/scala-java8-compat-2-11-1-0-2-sources-
jar/scala/compat/java8/FutureConverters.scala
* /opt/cola/permits/1425205998_1663952624.3076522/0/scala-java8-compat-2-11-1-0-2-sources-
jar/scala/compat/java8/converterImpl/StepsLikeIterator.scala
* /opt/cola/permits/1425205998_1663952624.3076522/0/scala-java8-compat-2-11-1-0-2-sources-
jar/scala/compat/java8/converterImpl/StepsImmHashSet.scala
* /opt/cola/permits/1425205998_1663952624.3076522/0/scala-java8-compat-2-11-1-0-2-sources-
jar/scala/compat/java8/converterImpl/StepsMap.scala

```

1.324 javassist 3.18.2-GA

1.324.1 Available under license :

No license file was found, but licenses were detected in source scan.

```

/*
 * Javassist, a Java-bytecode translator toolkit.
 * Copyright (C) 1999- Shigeru Chiba. All Rights Reserved.
 *

```

* The contents of this file are subject to the Mozilla Public License Version
* 1.1 (the "License"); you may not use this file except in compliance with
* the License. Alternatively, the contents of this file may be used under
* the terms of the GNU Lesser General Public License Version 2.1 or later,
* or the Apache License Version 2.0.
*
* Software distributed under the License is distributed on an "AS IS" basis,
* WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License
* for the specific language governing rights and limitations under the
* License.
*/

Found in path(s):

* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/scopedpool/ScopedClassPoolRepositoryImpl.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/bytecode/BadBytecode.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/convert/TransformFieldAccess.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/compiler/ast/Member.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/tools/web/Webserver.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/tools/reflect/Metaobject.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/compiler/ast/Symbol.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/util/HotSwapper.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/bytecode/InnerClassesAttribute.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/bytecode/AnnotationsAttribute.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/bytecode/annotation/AnnotationsWriter.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/CannotCompileException.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/util/proxy/ProxyObject.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/bytecode/analysis/SubroutineScanner.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/bytecode/InstructionPrinter.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/ByteArrayClassPath.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/bytecode/ParameterAnnotationsAttribute.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-

jar/javassist/compiler/ast/StringL.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/util/proxy/Proxy.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/CtMethod.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/tools/rmi/RemoteRef.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/ClassMap.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/expr/MethodCall.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/bytecode/analysis/ControlFlow.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/bytecode/CodeIterator.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/util/proxy/MethodHandler.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/bytecode/stackmap/Tracer.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/Modifier.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/scopedpool/ScopedClassPoolFactory.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/ClassPath.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/bytecode/Descriptor.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/compiler/ast/CondExpr.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/bytecode/analysis/IntQueue.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/compiler/ast/ASTree.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/bytecode/AccessFlag.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/CtClassType.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/compiler/ast/Keyword.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/CtPrimitiveType.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/tools/reflect/Sample.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/tools/reflect/Compiler.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/util/proxy/RuntimeSupport.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-

jar/javassist/CtConstructor.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/bytecode/LongVector.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/bytecode/stackmap/MapMaker.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/bytecode/stackmap/TypeData.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/compiler/ast/ASTList.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/util/proxy/ProxyObjectOutputStream.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/CtNewConstructor.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/CtClass.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/CtNewWrappedMethod.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/CtArray.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/bytecode/ExceptionTable.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/bytecode/ByteStream.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/compiler/Parser.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/NotFoundException.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/ClassClassPath.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/compiler/Javac.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/tools/Dump.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/CtNewClass.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/bytecode/stackmap/BasicBlock.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/CtBehavior.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/convert/TransformNew.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/compiler/MemberResolver.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/bytecode/StackMap.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/convert/TransformWriteField.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-

jar/javassist/compiler/JvstCodeGen.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/Loader.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/Mnemonic.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/annotation/NoSuchClassError.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/analysis/FramePrinter.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/ClassFile.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/CtField.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/expr/Instanceof.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/EnclosingMethodAttribute.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/SerialVersionUID.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/LineNumberAttribute.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/expr/Handler.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/tools/web/Viewer.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/compiler/ast/FieldDecl.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/compiler/ast/MethodDecl.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/compiler/SymbolTable.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/compiler/ast/ArrayInit.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/tools/rmi/ObjectImporter.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/analysis/Frame.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/Bytecode.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/compiler/ast/Stmnt.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/tools/web/BadHttpRequest.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/tools/rmi/Proxy.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/tools/rmi/RemoteException.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-

jar/javassist/compiler/JvstTypeChecker.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/bytecode/SignatureAttribute.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/convert/TransformNewClass.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/compiler/ast/BinExpr.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/LoaderClassPath.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/bytecode/ClassFilePrinter.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/tools/reflect/CannotInvokeException.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/compiler/CompileError.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/tools/rmi/ObjectNotFoundException.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/compiler/TokenId.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/CtNewNestedClass.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/bytecode/analysis/Type.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/convert/TransformBefore.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/compiler/KeywordTable.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/CodeConverter.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/scopedpool/ScopedClassPool.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/bytecode/DeprecatedAttribute.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/convert/TransformCall.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/tools/framedump.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/tools/reflect/CannotCreateException.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/compiler/ast/Declarator.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/bytecode/MethodInfo.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/expr/FieldAccess.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/compiler/ast/IntConst.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-

jar/javassist/CtNewWrappedConstructor.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/bytecode/stackmap/TypeTag.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/bytecode/ByteArray.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/runtime/DotClass.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/compiler/ast/Variable.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/bytecode/stackmap/TypedBlock.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/compiler/AccessorMaker.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/util/proxy/MethodFilter.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/scopedpool/ScopedClassPoolFactoryImpl.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/expr/NewExpr.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/compiler/ast/InstanceOfExpr.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/util/proxy/ProxyObjectInputStream.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/compiler/ast/Expr.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/Translator.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/tools/reflect/CannotReflectException.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/compiler/ast/CallExpr.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/compiler/ast/DoubleConst.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/bytecode/LocalVariableTypeAttribute.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/tools/reflect/Loader.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/expr/Cast.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/util/proxy/SecurityActions.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/tools/reflect/Metalevel.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/CtMember.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/convert/TransformAccessArrayField.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-

jar/javassist/runtime/Inner.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/SourceFileAttribute.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/analysis/MultiArrayType.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/scopedpool/ScopedClassPoolRepository.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/CodeAnalyzer.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/ClassFileWriter.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/tools/rmi/StubGenerator.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/CodeAttribute.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/CtNewMethod.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/compiler/CodeGen.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/AnnotationDefaultAttribute.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/ConstantAttribute.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/analysis/MultiType.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/SyntheticAttribute.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/util/proxy/ProxyFactory.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/compiler/NoFieldException.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/compiler/ast/Pair.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/compiler/ast/AssignExpr.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/ExceptionsAttribute.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/tools/rmi/Sample.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/ClassPoolTail.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/runtime/Desc.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/expr/NewArray.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/expr/ExprEditor.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-

jar/javassist/bytecode/AttributeInfo.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/expr/ConstructorCall.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/compiler/ast/Visitor.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/bytecode/analysis/Subroutine.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/tools/rmi/AppletServer.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/bytecode/StackMapTable.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/compiler/MemberCodeGen.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/tools/reflect/ClassMetaobject.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/convert/TransformAfter.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/URLClassPath.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/compiler/SyntaxError.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/scopedpool/SoftValueHashMap.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/compiler/Lex.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/ClassPool.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/compiler/ast/NewExpr.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/expr/Expr.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/compiler/TypeChecker.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/bytecode/Opcodes.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/tools/reflect/Reflection.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/convert/TransformReadField.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/bytecode/LocalVariableAttribute.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/util/proxy/FactoryHelper.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/compiler/ProceedHandler.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/bytecode/DuplicateMemberException.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-

jar/javassist/util/proxy/SerializedProxy.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/bytecode/analysis/Util.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/compiler/ast/CastExpr.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/bytecode/annotation/AnnotationImpl.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/convert/Transformer.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/bytecode/analysis/Executor.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/bytecode/ConstPool.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/runtime/Cflow.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/bytecode/analysis/Analyzer.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/bytecode/FieldInfo.java

No license file was found, but licenses were detected in source scan.

```
/*  
* Javassist, a Java-bytecode translator toolkit.  
* Copyright (C) 2004 Bill Burke. All Rights Reserved.  
*  
* The contents of this file are subject to the Mozilla Public License Version  
* 1.1 (the "License"); you may not use this file except in compliance with  
* the License. Alternatively, the contents of this file may be used under  
* the terms of the GNU Lesser General Public License Version 2.1 or later,  
* or the Apache License Version 2.0.  
*  
* Software distributed under the License is distributed on an "AS IS" basis,  
* WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License  
* for the specific language governing rights and limitations under the  
* License.  
*/
```

Found in path(s):

* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/bytecode/annotation/IntegerMemberValue.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/bytecode/annotation/Annotation.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/bytecode/annotation/ArrayMemberValue.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/bytecode/annotation/CharMemberValue.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-
jar/javassist/bytecode/annotation/EnumMemberValue.java

* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/annotation/StringMemberValue.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/annotation/FloatMemberValue.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/annotation/LongMemberValue.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/annotation/ShortMemberValue.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/annotation/DoubleMemberValue.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/annotation/MemberValueVisitor.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/annotation/ByteMemberValue.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/annotation/ClassMemberValue.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/annotation/MemberValue.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/annotation/AnnotationMemberValue.java
* /opt/ws_local/PERMITS_SQL/1014221952_1591370586.83/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/annotation/BooleanMemberValue.java

1.325 micronaut-redis 5.3.0

1.325.1 Available under license :

Apache License

Version 2.0, January 2004

<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of,

publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution

notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing

the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.326 snappy-java 1.1.7

1.326.1 Available under license :

This product includes software developed by Google
Snappy: <http://code.google.com/p/snappy/> (New BSD License)

This product includes software developed by Apache
PureJavaCrc32C from apache-hadoop-common <http://hadoop.apache.org/>
(Apache 2.0 license)

This library contained statically linked libstdc++. This inclusion is allowed by "GCC Runtime Library Exception"
<http://gcc.gnu.org/onlinedocs/libstdc++/manual/license.html>

== Contributors ==

- * Tatu Saloranta
 - * Providing benchmark suite
- * Alec Wysoker
 - * Performance and memory usage improvement

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate

as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify

the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include

the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.327 micronaut-test 3.6.2

1.327.1 Available under license :

Bean Validation API

License: Apache License, Version 2.0

See the license.txt file in the root directory or <<http://www.apache.org/licenses/LICENSE-2.0>>.

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

1. Definitions.

1.1. Contributor. means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version. means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software. means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable. means the Covered Software in any form other than Source Code.

1.5. Initial Developer. means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work. means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License. means this document.

1.8. Licensable. means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications. means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software. means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims. means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code. means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You. (or .Your.) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, .You. includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, .control. means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the

combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN .AS IS. BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as .Participant.) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a .commercial item., as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of .commercial computer software. (as that term is defined at 48 C.F.R. ? 252.227-7014(a)(1)) and .commercial computer software documentation. as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction.s conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys. fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program).

Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest

validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.

Copyright (C)

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author

Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

"CLASSPATH" EXCEPTION TO THE GPL VERSION 2

Certain source files distributed by Sun Microsystems, Inc. are subject to the following clarification and special exception to the GPL Version 2, but only where Sun has expressly included in the particular source file's header the words

"Sun designates this particular file as subject to the "Classpath" exception as provided by Sun in the License file that accompanied this code."

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License Version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module.? An independent module is a module which is not derived from or based on this library.? If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so.? If you do not wish to do so, delete this exception statement from your version.

Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library.

It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has been in development since 2007.

It is currently developed by a community of developers, as well as supported commercially by FasterXML.com.

Licensing

Jackson core and extension components may be licensed under different licenses.

To find the details that apply to this artifact see the accompanying LICENSE file.

For more information, including possible other licensing options, contact FasterXML.com (<http://fasterxml.com>).

Credits

A list of contributors may be found from CREDITS file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.

Apache License

Version 2.0, January 2004

<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner

or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions

of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the

editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the

same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.328 metrics-integration-with-jmx 4.2.10

1.328.1 Available under license :

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0

Automatic-Module-Name: com.codahale.metrics.jmx

Build-Jdk-Spec: 11

Bundle-Description: A set of classes which allow you to report metrics via JMX.

Bundle-License: <https://www.apache.org/licenses/LICENSE-2.0.html>

Bundle-ManifestVersion: 2

Bundle-Name: Metrics Integration with JMX

Bundle-SymbolicName: io.dropwizard.metrics.jmx

Bundle-Version: 4.2.10

Created-By: Apache Maven Bundle Plugin 5.1.6

Export-Package: com.codahale.metrics.jmx;uses:="com.codahale.metrics,javax.management";version="4.2.10"

Implementation-Title: Metrics Integration with JMX

Implementation-Version: 4.2.10

Import-Package: org.slf4j;version="[1.6.0,2.0.0)",com.codahale.metrics;version="[4.2,5)",javax.management

Require-Capability: osgi.ee;filter:="(&(osgi.ee=JavaSE)(version=1.8))"

Tool: Bnd-6.2.0.202202251641

Found in path(s):

* /opt/cola/permits/1473459846_1668563467.3623354/0/metrics-jmx-4-2-10-jar/META-INF/MANIFEST.MF

1.329 paranamer-core 2.8

1.329.1 Available under license :

<OWNER> = Regents of the University of California

<ORGANIZATION> = University of California, Berkeley

<YEAR> = 1998

In the original BSD license, both occurrences of the phrase "COPYRIGHT HOLDERS AND CONTRIBUTORS" in the disclaimer read "REGENTS AND CONTRIBUTORS".

Here is the license template:

Copyright (c) <YEAR>, <OWNER>

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the <ORGANIZATION> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.330 apache-log4j-jul-adapter 2.17.1

1.330.1 Available under license :

Apache Log4j JUL Adapter
Copyright 1999-1969 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work

(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses

granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]"

replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.331 google-guice 4.2.2

1.331.1 Available under license :

Google Guice - Extensions - MultiBindings
Copyright 2006-2018 Google, Inc.

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common

control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or

documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill,

work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

©2023 Cisco Systems, Inc. All rights reserved.